NXP Semiconductors
Application Note

Document Number: AN5117
Rev. 1, 06/2016

Kinetis Migration Guide: K10 - 120 MHz to KV5x

1. Introduction

This document describes the details of migrating from Kinetis K10-120 MHz to KV5x-240 MHz microcontrollers. Migrating between the two devices may require hardware and/or software changes. This document describes the changes required when migrating from Kinetis K10-120 MHz to KV5x-240 MHz.

Contents

1. Introduction 2
1.1. Part numbering and mask set information 3
2. System level comparison 3
2.1. KV5x 240 MHz device overview 3
2.2. K10 120 MHz device overview 3
2.3. High level comparison 4
2.4. System modules comparison 7
3. Peripheral module comparison 17
3.1. Unchanged modules 18
3.2. Modified modules 20
3.3. New modules 35
4. Hardware comparison 37
4.1. Package/pinout differences 38
4.2. GPIO considerations 43
4.3. Clocking considerations 43
5. Revision history 43

1.1. Part numbering and mask set information

The table below lists the Kinetis K10-120 MHz and KV5x-240 MHz part numbers referred in this document.

Table 1. Part number migration

Part Number Origin (144 LQFP)	Part Number Destination
MK10FN1M0VLQ12	MKV58F1M0VLQ24
	MKV56F1M0VLQ24
MK10FX512VLQ12	MKV58F512VLQ24
	MKV56F512VLQ24

This document is focused on addressing the changes in functionality between these Kinetis microcontrollers. For general hardware and software design for the Kinetis V microcontrollers, see the Quick Reference Users Guide (KVQRUG.pdf).

2. System level comparison

This section provides a comparison between the KV5x 240 MHz and K10 120 MHz devices.

2.1. KV5x 240 MHz device overview

The KV5x device is the first Cortex-M7 based device by NXP that offers exceptional precision, sensing, and control for some of the most demanding applications in motor control and power conversion. This device operates at a maximum clock frequency of 240 MHz and features a variety of modules suited to motor control and power conversion, such as:

- High resolution pulse-width modulation (PWM) with down to 260 picosecond resolution
- Four 12-bit analog-to-digital converters (ADCs) sampling at a maximum of 5 Mega Samples Per Second (MSPS)
- Total of 44 PWM channels for support of multi-motor systems with PFC
- Three FlexCAN modules
- Optional Ethernet communications

2.2. K10 120 MHz device overview

The K10 120 MHz device is a Cortex-M4 based device targeted for the general market. It features:

- Fast, high precision 16-bit ADCs
- 12-bit DACs
- High speed comparators
- Powerful FlexTimers for PWM and motor control functions

2.3. High level comparison

As these systems are built on different processor cores and meant for different purposes, there will be a significant number of differences between the two devices. This does not mean that there is not a logical migration path between the two devices. Table 2 outlines the system level differences at a high level.

Table 2. System level differences

Feature	K10 – 120 MHz	KV5x – 240 MHz
Processor Core	Cortex-M4	Cortex-M7
Max CPU frequency	120 MHz	240 MHz
Mips/MHz	1.27 / 1.55 / 1.95	2.14 / 2.55 / 3.23
Cache	2 x 8KB I/D	16 KB I + 8 KB D
MPU	NXP System MPU	ARM MPU + NXP System MPU
FPU	Single Floating Point	Single Floating Point
DSP	Yes	Yes
Debug	JTAG + cJTAG + SWD	JTAG + cJTAG + SWD
Boundary Scan	Yes	Yes
Flash size	1MB or 512KB + 512 KB FlexNVM	1MB or 512 KB
RAM	128KB	256KB/128KB
SDRAM	LPDDR / DDR / DDR2	None
NAND flash	8/16 b data interface	None
External Flexbus	Yes	Yes
CRC	Yes	Yes
Oscillator	3-32 MHz	3-32 MHz
PLL	64 – 120 MHz	90 – 240 MHz
FLL	20-100 MHz	20-100 MHz

2.3.1. Memory map comparison

The memory map of the KV5x device is much different from the K10 120 MHz device. It is very important that you update your linker control file and do not try to use the K10 120 MHz device linker control file when compiling your KV5x project or vice versa. Table 3 is a side-by-side comparison of the two memory maps.

Table 3. Side-by-side comparison of memory maps

K10			KV5X	
System 32-bit Address Range	**Destination Slave**		**System 32-bit Byte Address Range**	**Destination Slave**
0x0000_0000–0x07FF_FFFF	Program flash and read-only data (Includes exception vectors in first 1024 bytes)		0x0000_0000–0x0000_FFFF	ITCM RAM - 64K bytes
0x0800_0000–0x0FFF_FFFF	DRAM Controller (Aliased Area)		0x0001_0000–0x0FFF_FFFF	Reserved
0x1000_0000–0x13FF_FFFF	FlexNVM		0x1000_0000–0x100F_FFFF	Code Flash 1M bytes (cache-able)
0x1000_0000–0x13FF_FFFF	Reserved		0x1010_0000–0x17FF_FFFF	Reserved
0x1400_0000–0x17FF_FFFF	For devices with FlexNVM: FlexRAM		0x1800_0000-0x1800_0FFF	EEERAM (4Kbytes)
0x1400_0000–0x17FF_FFFF	For devices with program flash only: Programming acceleration RAM		0x1800_1000-0x1FFF_FFFF	Reserved

Table 3. Side-by-side comparison of memory maps (contd…)

K10			KV5X	
0x1800_0000–0x1BFF_FFFF	FlexBus (Aliased Area). 0x1800_0000-0x1BFF_FFFF are mapped to the same access space of 0x9800_0000-0x9BFF_FFFF.		0x2000_0000–0x2001_FFFF	DTCM - D0TCM - 64K and D1TCM 64K
0x1C00_0000–0x1FFF_FFFF	SRAM_L: Lower SRAM (ICODE/DCODE)		0x2002_0000–0x2EFF_FFFF	Reserved
0x2000_0000–0x200F_FFFF	SRAM_U: Upper SRAM bitband region		0x2F00_0000–0x2F00_FFFF	OC-RAM 64K (cache-able)
0x2010_0000–0x21FF_FFFF	Reserved		0x2F01_0000–0x3FFF_FFFF	Reserved
0x2200_0000–0x23FF_FFFF	Aliased to TCMU SRAM bitband		0x4000_0000–0x4007_FFFF	AIPS0
0x2400_0000–0x3FFF_FFFF	Reserved		0x4008_0000–0x400F_EFFF	AIPS1
0x4000_0000–0x4007_FFFF	Bitband region for AIPS0		0x400F_F000–0x400F_FFFF	Bitband region for general purpose input/ output (GPIO)
0x4008_0000–0x400F_EFFF	Bitband region for AIPS1		0x4010_0000–0x43FF_FFFF	Reserved
0x400F_F000–0x400F_FFFF	Bitband region for GPIO		0x4400_0000–0x5FFF_FFFF	BME
0x4010_0000–0x41FF_FFFF	Reserved		0x6000_0000–0xAFFF_FFFF	Reserved
0x4200_0000–0x43FF_FFFF	Aliased to AIPS and GPIO bitband		0xB000_0000-0xBFFF_FFFF	Flexbus (256Mbyte range)
0x4400_0000–0x5FFF_FFFF	Reserved		0xC000_0000-0xDFFF_FFFF	Reserved
0x6000_0000–0x6FFF_FFFF	Flexbus (External memory - Write-back)		0xE000_0000–0xE00F_FFFF	Private Peripherals
0x9000_0000–0x9FFF_FFFF	FlexBus (External memory - Write-through)		0xE010_0000 - 0xFFFF_FFFF	Reserved
0xA000_0000–0xDFFF_FFFF	FlexBus (External peripheral - not executable)			
0xE000_0000–0xE00F_FFFF	Private Peripherals			
0xE010_0000–0xFFFF_FFFF	Reserved			

2.3.2. Clocking differences

It is important to note the differences in the clocking diagrams as these differences can significantly affect the setup of your application. Figure 1 shows the K10 120 MHz clocking diagram and Figure 2 shows the KV5x 240 MHz clocking diagram.

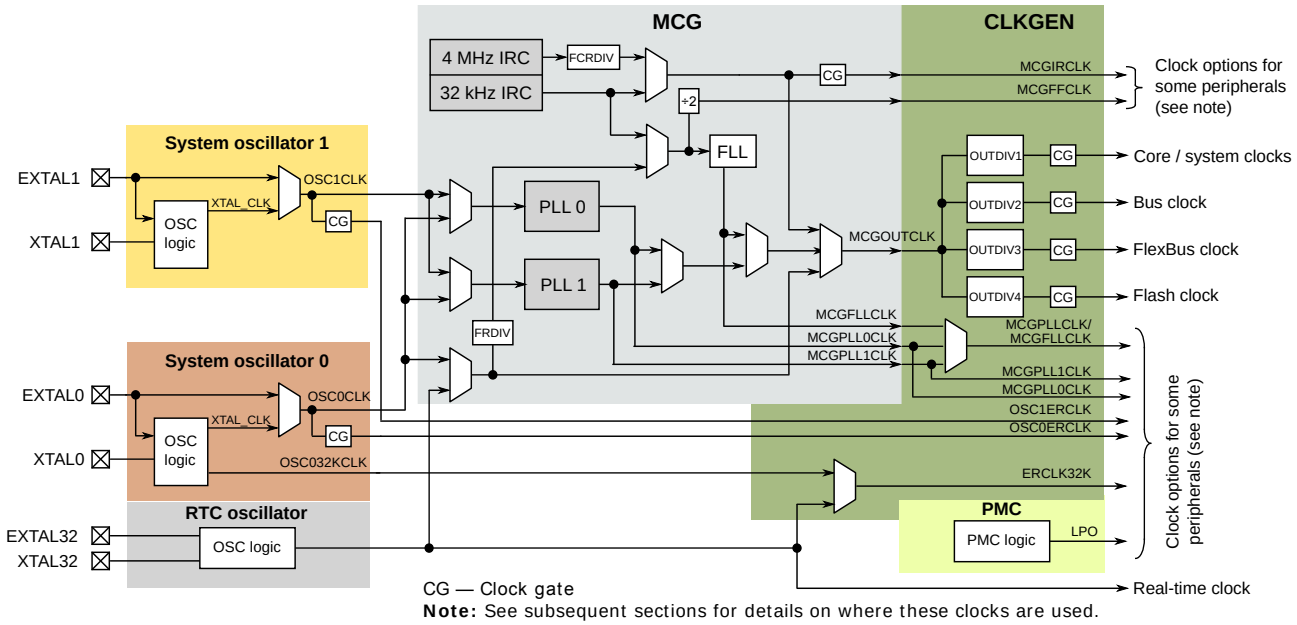
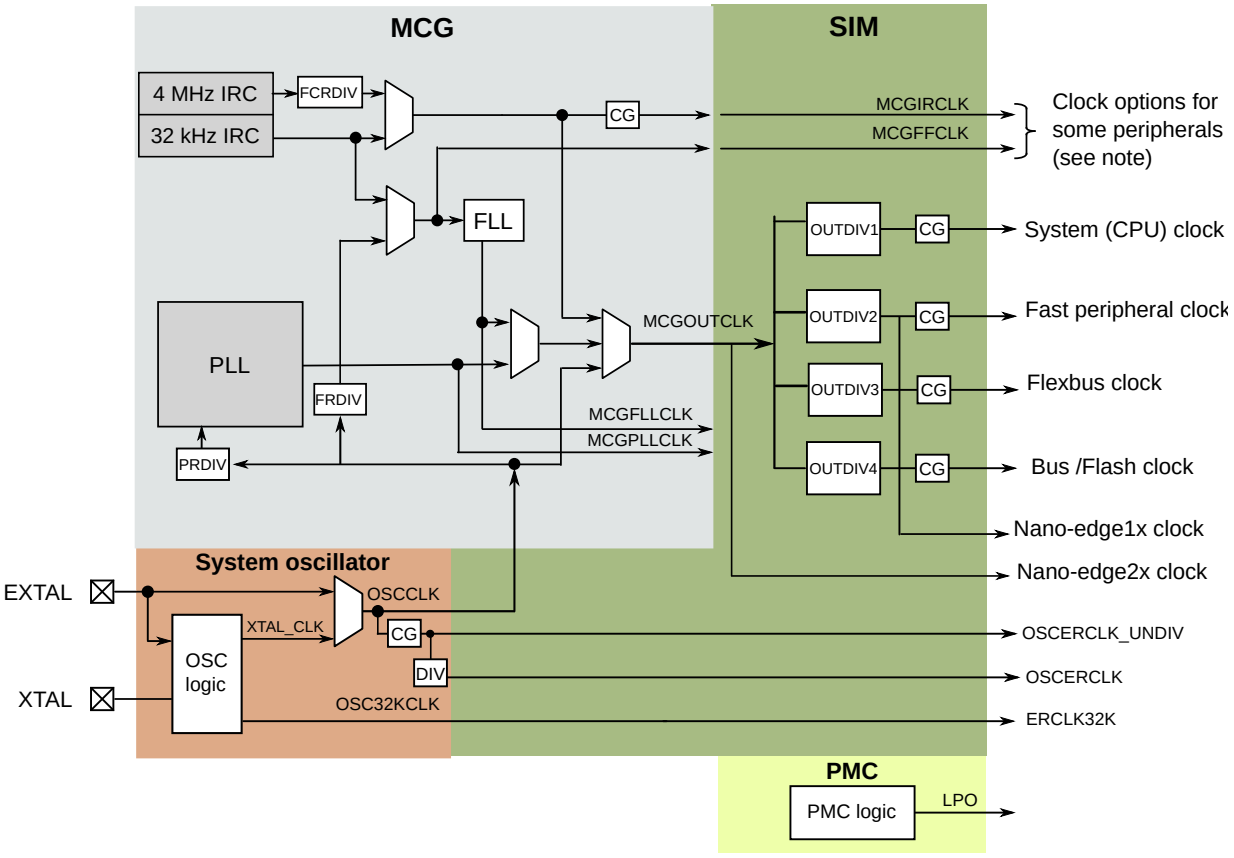


Figure 1. K10 120 MHz device clocking diagram



CG — Clock gate
Note: See subsequent sections for details on where these clocks are used.

Figure 2. KV5x 240 MHz clocking diagram

As can be seen in the KV5x clocking diagram, the main clock (MCGOUTCLK) is routed to four dividers just as in K10. However, in KV5x, OUTDIV2 is routed to a new clock path, the fast peripheral clock. The Bus and Flash clock are now combined onto OUTDIV4 in KV5x. The following sections describe which peripherals are used by which clocks.

NOTE

The Fast Peripheral clock on KV5x is capable of much higher frequencies than the Bus clock on K10. This is considered in the migration from K10 to KV5x.

Other key differences include only a single PLL and a single oscillator for KV5x. If your application uses PLL1, System Oscillator 1, or the RTC oscillator, you will need to remove the code and the hardware associated with these items and migrate to the main oscillator (System Oscillator 0) and PLL (PLL0).

2.3.3. Peripheral interconnect

The KV5x device adds an Inter-Peripheral Crossbar Switch (XBAR) that allows flexibility in connecting inputs (external GPIO or internal module outputs) to outputs (other external GPIO or internal module inputs). This module is discussed in the subsequent sections where peripheral interconnect options change.

2.4. System modules comparison

Table 4 lists the system level module differences in the programming model on the KV5x device. These modules are treated slightly differently because there is only one of each of these per Kinetis device (with the exception of the oscillator) and/or are typically device specific. Each of these modules that require changes is discussed in detail in the following subsections.

Table 4. System modules comparison

Module	Programming Model Comments
MCG	Some differences
OSC	Additions
SMC	Some differences
PMC	Additions
RCM	Some differences
AXBS	Same
FMC	Some differences
FTFE	Some differences

2.4.1. Multipurpose clock generator

The biggest difference between the K10 120 MHz device MCG and the KV5x 240 MHz device MCG is that the KV5x MCG does not include a second PLL and it only includes one oscillator. Therefore, if your application is using PLL1, then it is necessary to modify your code to use the main PLL (PLL0). Likewise, if you are using an oscillator other than OSC0, you need to use OSC0. This also requires a

hardware change. For details on hardware changes, see to the pin mux comparison in this migration guide. See Figure 3 and Figure 4 for a block diagram comparison.

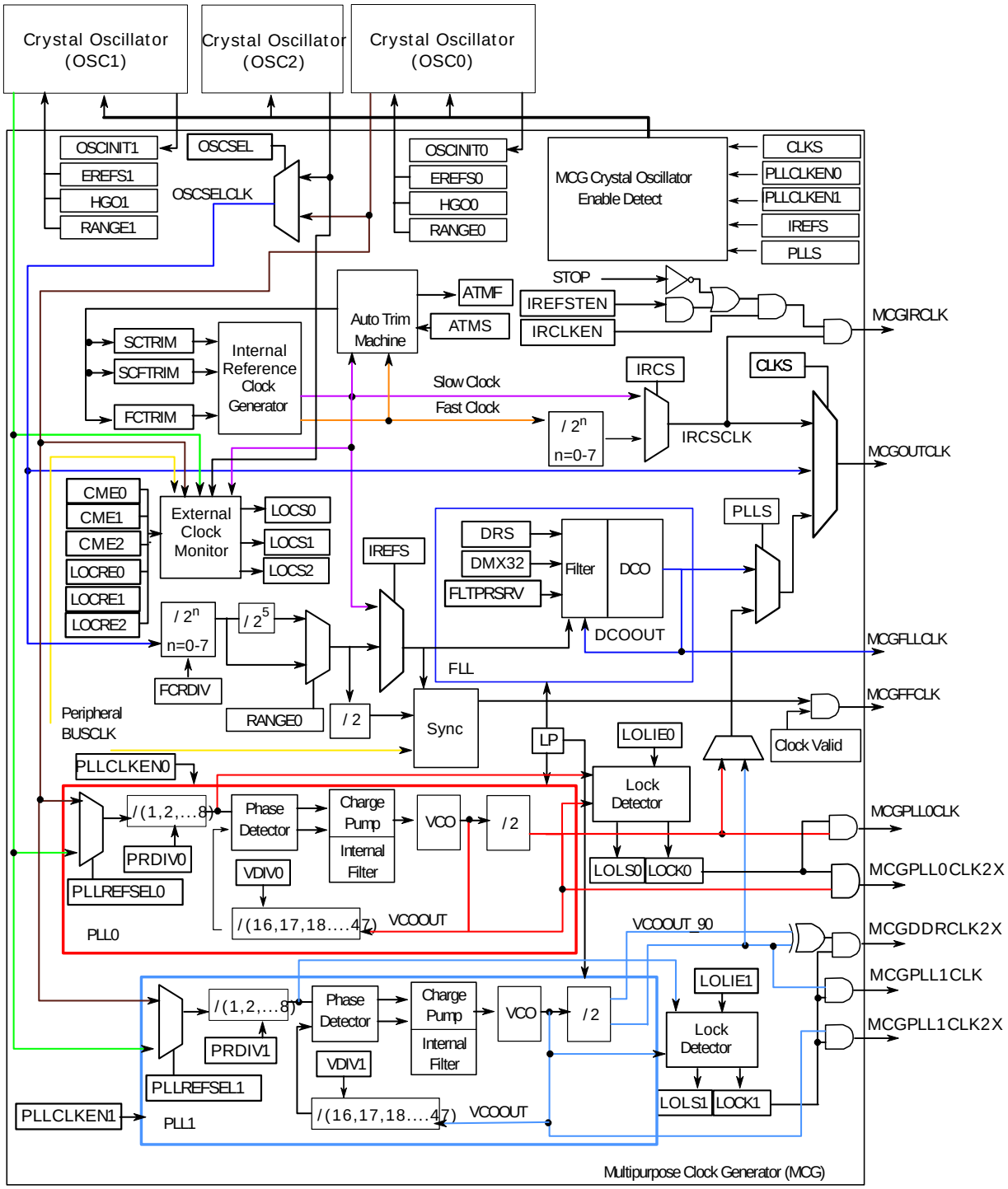


Figure 3. K10 MCG block diagram

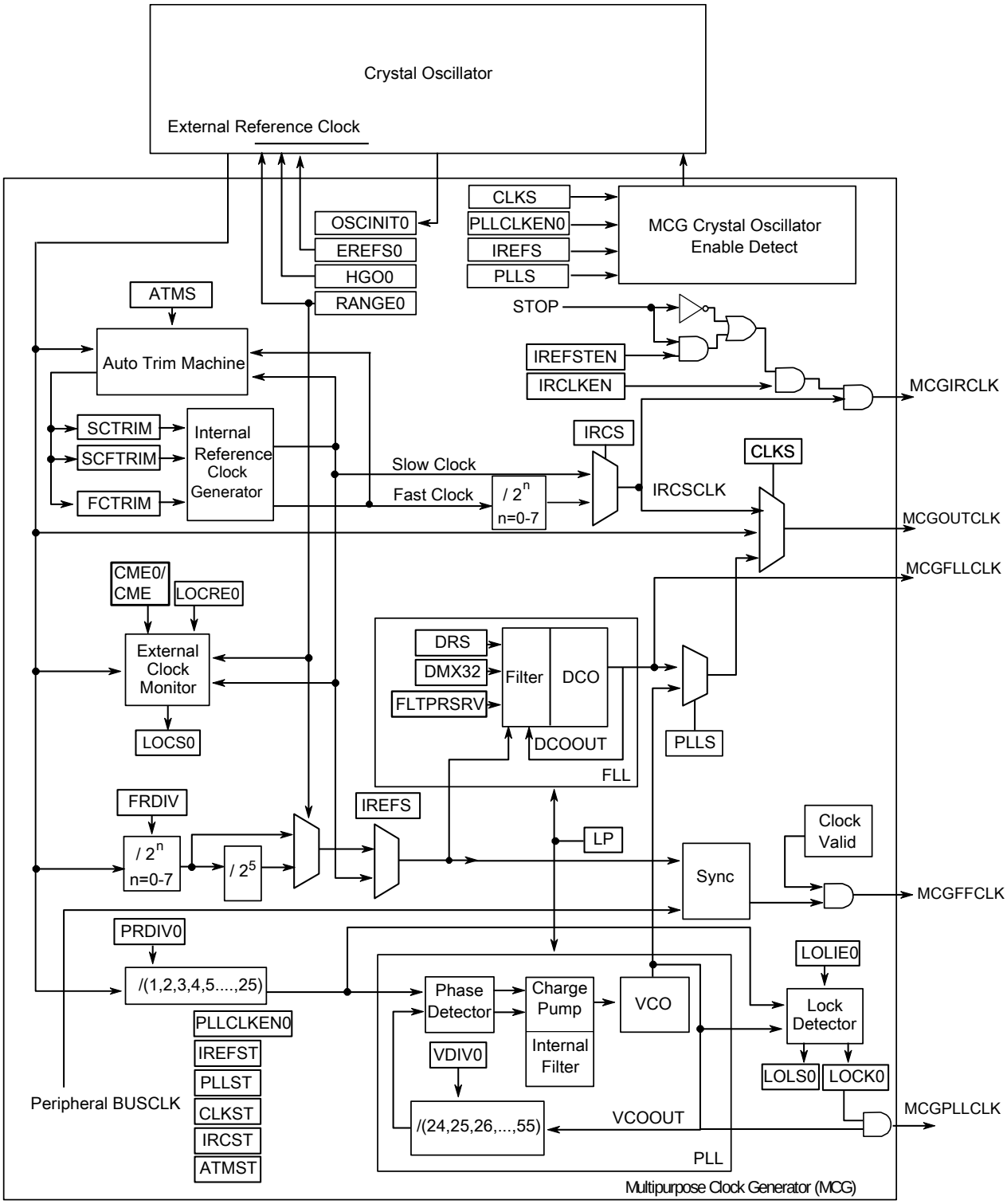


Figure 4. KV5x MCG block diagram

2.4.1.1. MCG memory map comparison

There is no change to the main memory map. The registers of the MCG for KV5x are a subset of the MCG registers for K10 and they reside in the same locations for both devices. However, there may be some differences in the functionality of some bits within the registers. The following section compares those register differences.

Bit	7	6	5	4	3	2	1	0
Read / Write	LOCRE0	FCFTRIM	RANGE		HGO	EREFS	LP	IRCS
Reset	1	0	0	0	0	0	0	0

Figure 5. MCG Control 2 register (MCG_C2) – KV5x

Added bits:

- **FCFTRIM:** Fast Internal Reference Clock Fine Trim – This addition enables greater resolution in the Fast Internal Reference Clock Fine Trim.

Bit	7	6	5	4	3	2	1	0
Read	0	PLLCLKEN	PLLSTEN	0		PRDIV		
Write								
Reset	0	0	0	0	0	0	0	0

Figure 6. MCG Control 5 register (MCG_C5) – KV5x

Removed bits:

- **PLLREFSEL0:** PLL0 External Reference Select –This is because KV5x has only one reference selection and the PLLREFSEL0 bit has been removed.

Bit	7	6	5	4	3	2	1	0
Read	0	LOLRE	0	0				0
Write								
Reset	1	0	0	0	0	0	0	0

Figure 7. MCG Control 8 register (MCG_C8)

Added bits:

- **LOLRE:** PLL Loss of Lock Reset Enable. This bit enables the loss of lock detection circuitry to request a reset of the chip upon loss of PLL lock.

Removed bits:

- **LOCRE1:** Loss of Clock Reset Enable – This bit was used strictly for the second oscillator, which is not present on KV5x.
- **CME1:** Clock Monitor Enable 1 – This bit was strictly for the second oscillator, which is not present on KV5x.
- **LOCS1:** RTC Loss of Clock Status – This bit was used strictly for the RTC clock, which is not present on KV5x.

2.4.2. Oscillator

There are two major differences between the oscillators on these two devices:

- The K10 120 MHz device contains two high-frequency capable oscillators. If you are using the second oscillator, you should migrate to the first.
- The KV5x device adds a mechanism to divide OSCERCLK to allow more flexibility in peripheral frequencies. This results in a slight block diagram modification and memory map register definition modification.

Figure 8 is a block diagram of the KV5x Oscillator block with the highlighted addition.

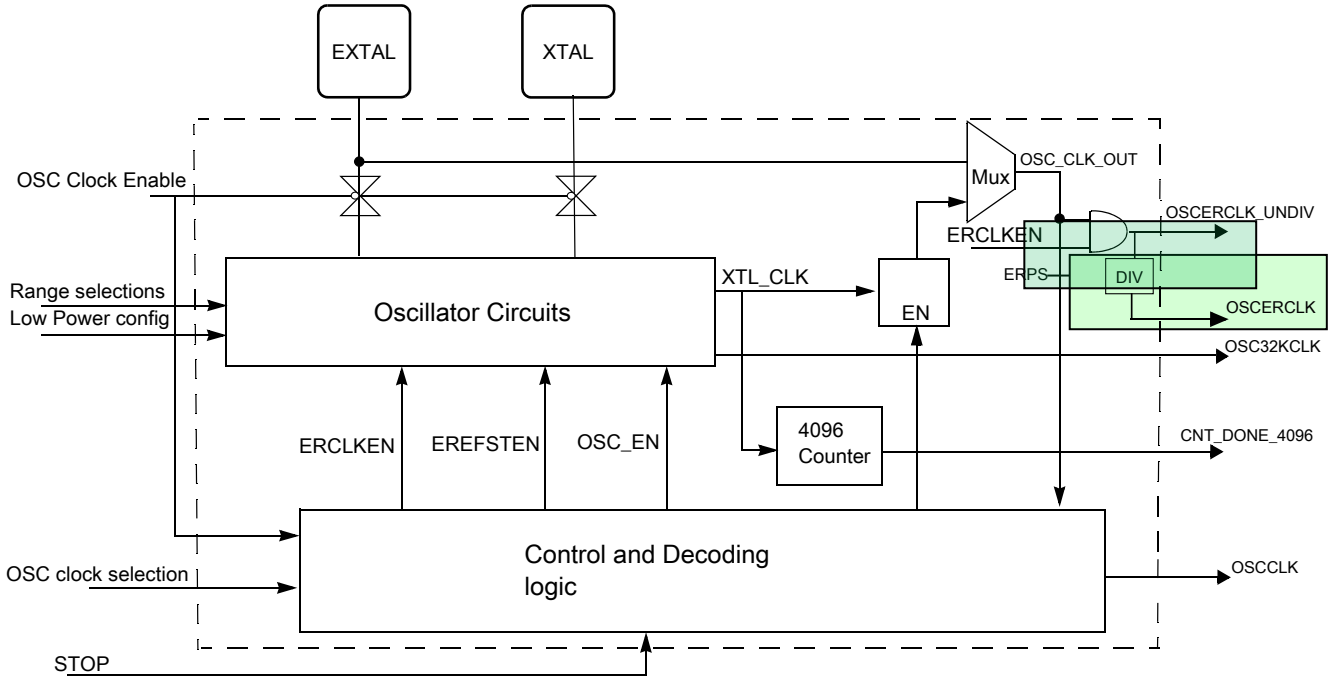


Figure 8. Oscillator block diagram – KV5x

The following is a comparison of the memory map of the K10 and KV5x oscillators

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value
4006_5000	OSC Control Register (OSC0_CR)	8	R/W	00h
400E_5000	OSC Control Register (OSC1_CR)	8	R/W	00h

Figure 9. OSC Memory Map/Register Definition – K10

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value
4006_5000	OSC Control Register (OSC0_CR)	8	R/W	00h
4006_5002	OSC_DIV (OSC0_OSC_DIV)	8	R/W	00h

Figure 10. OSC memory map/register definition – KV5x

Since there are no changes between the K10 120 MHz device OSC Control register and the KV5x OSC Control register, your code should work as is on the KV5x device and no memory map comparison is needed. Only the new OSC_DIV register are briefly discussed below.

2.4.2.1. OSC_DIV register

Bit	7	6	5	4	3	2	1	0
Read	ERPS		0	0	0	0	0	0
Write								
Reset	0	0	0	0	0	0	0	0

Figure 11. OSC_DIV register

Added bits:

ERPS: ERCLK prescaler – This bit field allows the oscillator to be divided by 1, 2, 4, or 8 enabling flexibility in the clock frequency that is fed to the peripheral modules.

2.4.3. System mode controller (SMC)

The K10 120 MHz device implements an older version of the SMC than is present on the KV5x device. The following list summarizes the changes between the K10 and KV5x:

- HSRUN mode – The KV5x SMC contains a High-Speed RUN (HSRUN) option that the K10 does not offer. This feature allows the KV5x to operation at speeds up to 240 MHz.
- VLLS0 mode – The KV5x SMC adds a VLLS0 mode of operation for increased power savings mode.
- LLS mode – The LLS mode has been removed from KV5x. If your application was using this mode of operation, it is recommended to use VLPS mode.
- SMC_VLLSCTRL register – The SMC_VLLSCTRL register of K10 has been renamed to SMC_STOPCTRL in KV5x.

The following section 2.4.3.1 is a detailed register comparison.

2.4.3.1. SMC memory map comparison

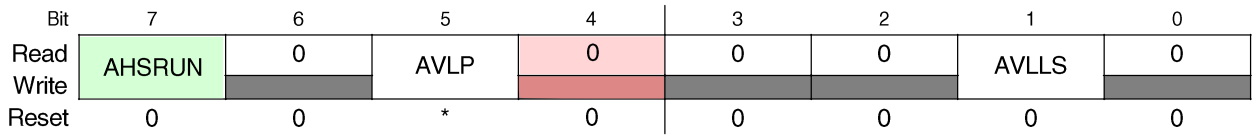


Figure 12. SMC Power Mode Protection register (SMC_PMPROT)

Added bits:

- AHSRUN: Allow High Speed RUN mode – This bit has been added to allow the High Speed Run mode.

Removed bits:

- ALLS: Allow Low-Leakage Stop mode – This bit has been removed because KV5x does not support the Low-Leakage Stop mode.

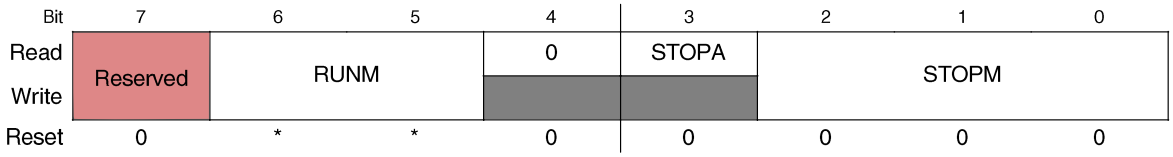


Figure 13. SMC Power Mode Control register (SMC_PMCTRL)

Removed bits:

- LPWUI: Low Power Wake Up on Interrupt – The LPWUI bit has been removed from the KV5x SMC_PMCTRL register because KV5x does not support this functionality.

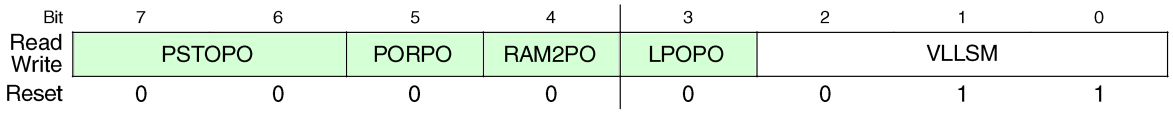


Figure 14. SMC Stop Control register (SMC_STOPCTRL; formerly SMC_VLLSCTRL) – KV5x

Added bits:

- PSTOPO: Partial Stop Option – Controls the Partial STOP mode options. Partial stop is an added clocking option of the STOP mode and allows the customization of which clocks to disable when entering STOP.
- PORPO: POR Power Option – Controls whether the POR detect circuit is enabled or disabled in VLLS0.
- RAM2PO: RAM2 Power Option – Determines whether the RAM2 section is powered in VLLS2.
- LPOPO: LPO Power Option – Enables or disables the 1 kHz LPO clock in VLLSx modes.

2.4.4. Power mode controller (PMC)

The KV5x device implements an improved PMC that adds an option to keep the band-gap voltage reference enabled in VLPx modes. The KV5x device includes a high-voltage detect circuit as well as a

low-voltage detect circuit. The high-voltage detect circuit is configured to reset the device or simply generate an interrupt. The PMC registers are of the same width and are in the same locations on both the devices. So you can port the K10 code can directly to the KV5x code. The following is a detailed look at the enhancements of the KV5x PMC.

2.4.4.1. PMC memory map comparison

Bit	7	6	5	4	3	2	1	0
Read	0	0	Reserved	BGEN	ACKISO	REGONS	Reserved	BGBE
Write					w1c			
Reset	0	0	0	0	0	1	0	0

Figure 15. PMC Regulator Status and Control register (PMC_REGSC) – KV5x

Added bits:

- BGEN: Bandgap Enable in VLPx Operation – KV5x adds the BGEN bit which allows the bandgap voltage reference to be enabled in VLPx and VLLSx modes.

Bit	7	6	5	4	3	2	1	0
Read	HVDF	0	HVDIE	HVDRE	0			HVDV
Write		HVDACK						
Reset	0	0	0	0	0	0	0	1

Figure 16. PMC High Voltage Detect Status and Control 1 register (PMC_HVDSC1) – KV5x

Added bits:

- HVDF: High-Voltage Detect Flag – Indicates if a high-voltage detect event has occurred.
- HVDACK High-Voltage Detect Acknowledge – This bit is used to acknowledge high voltage detect errors and clears the HVDF bit.
- HVDIE: High-Voltage Detect Interrupt Enable – Enables the High-Voltage Detect interrupt.
- HVDRE: High-Voltage Detect Reset Enable – Enables the ability to generate a reset when a high-voltage event occurs.
- HVDV: High-Voltage Detect Voltage Select – Selects the trip point voltage for high-voltage events.

2.4.5. Reset control module (RCM)

The KV5x module adds two registers (plus some bits) to the RCM and removes registers/bits related to the EZPort module. If your application was using the EZPort you need to remove these references. Figure 17 below shows the high-level memory map for the KV5x devices and highlights the differences as compared against the 120MHz K10.

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value
4007_F000	System Reset Status Register 0 (RCM_SRS0)	8	R	82h
4007_F001	System Reset Status Register 1 (RCM_SRS1)	8	R	00h
4007_F004	Reset Pin Filter Control register (RCM_RPFC)	8	R/W	00h
4007_F005	Reset Pin Filter Width register (RCM_RPFW)	8	R/W	00h
4007_F008	Sticky System Reset Status Register 0 (RCM_SSRS0)	8	R/W	82h
4007_F009	Sticky System Reset Status Register 1 (RCM_SSRS1)	8	R/W	00h

Figure 17. KV5x RCM Memory Map

2.4.5.1. RCM memory map comparison

Bit	7	6	5	4	3	2	1	0
Read	POR	PIN	WDOG	0	LOL	LOC	LVD	WAKEUP
Write								
Reset	1	0	0	0	0	0	1	0

Figure 18. RCM System Reset Status register 0 (RCM_SRS0) – KV5x

Added bits:

- LOL: Loss-of-Lock – Indicates that reset was caused by a loss of lock in the PLL.

Bit	7	6	5	4	3	2	1	0
Read	0	0	SACKERR	0	MDM_AP	SW	LOCKUP	JTAG
Write								
Reset	0	0	0	0	0	0	0	0

Figure 19. RCM System Reset Status register 1 (RCM_SRS1) – KV5x

Removed bits:

- EZPT: The EZPT bit has been removed from the RCM_SRS1 register as KV5x does not include the EZ Port module.

2.4.5.2. Register additions

Bit	7	6	5	4	3	2	1	0
Read	SPOR	SPIN	SWDOG	0	SLOL	SLOC	SLVD	SWAKEUP
Write	w1c	w1c	w1c		w1c	w1c	w1c	w1c
Reset	1	0	0	0	0	0	1	0

Figure 20. RCM Sticky System Reset Status register 0 (RCM_SSRS0) – KV5x

Added bits:

- SPOR: Sticky Power-On Reset – Indicates a reset has been caused by the power-on detection logic; must be cleared by software as it is unaffected by resets except for POR, LVD, or VLLSx resets.
- SPIN: Sticky External Reset Pin – Indicates a reset has been caused by an active-low level on the external RESET pin; must be cleared by software as it is unaffected by resets except for POR, LVD, or VLLSx resets.
- SWDOG: Sticky Watchdog – Indicates a reset has been caused by the watchdog timer timing out; must be cleared by software as it is unaffected by resets except for POR, LVD, or VLLSx resets.
- SLOL: Sticky Loss-of-Lock Reset – Indicates a reset has been caused by a loss of lock in the MCG PLL; must be cleared by software as it is unaffected by resets except for POR, LVD, or VLLSx resets.
- SLOC: Sticky Loss-of-Clock Reset – Indicates a reset has been caused by a loss of the external clock; must be cleared by software as it is unaffected by resets except for POR, LVD, or VLLSx resets.
- SLVD: Sticky Low-Voltage Detect Reset – Indicates a reset has been caused by the device supply voltage dropping below the LVD trip point (and PMC_LVDSC1[LVDRE] is set); must be cleared by software as it is unaffected by resets except for POR, LVD, or VLLSx resets.
- SWAKEUP: Sticky Low Leakage Wakeup Reset – Indicates a reset has been caused by an enabled LLWU module wakeup source while the device was in a low leakage mode; must be cleared by software as it is unaffected by resets except for POR, LVD, or VLLSx resets.

Bit	7	6	5	4	3	2	1	0
Read	0	0	SSACKERR	0	SMDM_AP	SSW	SLOCKUP	SJTAG
Write			w1c		w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

Figure 21. RCM Sticky System Reset Status register 1 (RCM_SSRS1) – KV5x

Added bits:

- SSACKERR: Sticky Stop Mode Acknowledge Error Reset – Indicates a reset has been caused by a failure of one or more peripherals to acknowledge a Stop mode entry signal; must be cleared by software as it is unaffected by resets except for POR, LVD, or VLLSx resets.
- SMDM_AP: Sticky MDM-AP System Reset Request – Indicates a reset has been caused by the host debugger setting the System Reset Request bit in the MDM-AP Control Register; must be cleared by software as it is unaffected by resets except for POR, LVD, or VLLSx resets.
- SSW: Sticky Software Reset – Indicates a reset has been caused by setting the SYSRESETREQ bit in the Application Interrupt and Reset Control Register in the core; must be cleared by software as it is unaffected by resets except for POR, LVD, or VLLSx resets.
- SLOCKUP: Sticky Core Lockup – Indicates a reset has been caused by a software LOCKUP event; must be cleared by software as it is unaffected by resets except for POR, LVD, or VLLSx resets.
- SJTAG: Sticky JTAG Generated Reset – Indicates a reset has been caused by selection of certain

IR codes (EXTEST, HIGHZ, and CLAMP) by the JTAG module; must be cleared by software as it is unaffected by resets except for POR, LVD, or VLLSx resets.

3. Peripheral module comparison

The peripheral modules are classified.

The **unchanged modules** section outlines the details of the SOC implementation of the modules. The modules in this section are marked by *Unchanged* in the Programming Model Comments column of the Peripheral Differences table below (Table 5). Even though these modules were unchanged designs they may have been integrated differently and/or different clock sources may now be sourcing these modules.

The **modified modules** section outlines the modules that have been updated to use newer/different versions or simply have some minor differences. The overall functionality provided is similar. However, changes are required in software and possibly hardware changes are required in order to utilize updated features. These modules are marked by *Minor differences or Additions* in the Programming Model Comments column of the Peripheral Differences table below (Table 5).

The **new modules** section outlines the new modules that have been added and how they can benefit your design. They are marked with +in the Programming Model Comments column of the Peripheral Differences table below (Table 5).

The **removed modules** are of note. Unpredictable results will occur if a module that is present on the K10 is written to on the KV5X. They are marked with -in the Programming Model Comments column of the Peripheral Differences table below (Table 5). If your application is using a removed module, you should remove the code for this peripheral.

Table 5 presents a comparison of the peripheral modules found on the K10 120 MHz device and the KV5x 240 MHz device.

Table 5. Comparison of peripheral modules

Peripheral	Number of Instances K10 – 120 MHz	Number of Instances KV5x – 240 MHz	Programming Model Comments
ADC	4x	1x	Changed
HS ADC	0x	4x	+
HSCMP	4x	4x	Unchanged
DAC	2x	1x	Unchanged
VREF	1x	0x	-
PGA	4x	0x	-
I2C	2x	2x	Changed
SPI	3x	3x	Unchanged
UART	6x	6x	Changed
CMT	1x	0x	-
FlexCAN	2x	3x	Changed
SAI	2x	0x	-
Ethernet	0x	1x	+
PIT	1x	1x	Changed
PDB	1x	2x	Unchanged
LPTMR	1x	1x	Unchanged

FlexTimer	4x	4x	Changed
eFlexPWM	0x	2x	+
RTC	1x	0x	-
WDOG	1x	1x	Unchanged
EWM	1x	1x	Changed
Tamper	1x	0x	-
CRC	1x	1x	Changed
TRNG	0	1x	+
TSI	1x	0x	-
LCDC	1x	0x	-
GPIO	5V Tolerant	3V only	Changed

3.1. Unchanged modules

The unchanged modules are modules that require no changes, as far as the peripheral is concerned, in your software when migrating from K10 to KV5x. The package pin assignments and clock options may have changed. The clock options for these modules will be discussed in the following sections. Please refer to the *Package/pinout difference* section of this document for information regarding package/pinout differences.

3.1.1. HSCMP

While the comparator peripheral blocks are the same, there are slight integration differences. The comparators rely mainly on the reference provided to them and the timing module that is used for the Sample/Window timing (if your application utilizes this feature). First, the voltage reference information are examined.

3.1.1.1. CMP external references

Each instantiation of the CMP on KV5x has the 6-bit DAC sub-block connected to IN7 of the comparator. The 12-bit DAC reference is connected to the IN6 reference of each CMP instantiation. The KV5x CMP removes the VREF_OUT option for a reference. If your application was relying on this feature, you need to modify your hardware to provide a reference or rely on one of the DAC modules to provide a reference voltage for your comparator.

3.1.1.2. External window/sample input

Just as on the K10, the KV5x comparator uses the PDB pulse-out signals to control each CMP Sample/Window timing.

3.1.2. DAC

The KV5x device implements only one DAC module, DAC0. If your application was making use of DAC1, you will need to migrate to DAC0. Both DAC types (K10 and KV5x) implement a FIFO on the DAC module. There are slight differences in the integration of the DAC reference voltages.

3.1.2.1. 12-bit DAC reference

On KV5x, the DAC reference can be either VREFH or VDDA (whereas K10 offered either VREF_OUT or VDDA). In both devices, DACREF_2 input is VDDA and the same bit, DACx_C0[DACRFS] is used to select the different voltage reference. If your application was using VREF_OUT, you need to either make hardware modifications to select VREFH, or select VDDA as your DAC voltage reference.

3.1.2.2. 12-bit DAC triggers

The K10 device offered only one PDB, and thus the 12-bit DAC has only one trigger. KV5x, however, offers two PDBs and, in addition, the XBARA module can also trigger the DAC. If your application was utilizing the DAC trigger, your application code should continue to work without modification. Your application may benefit from using the XBARA module or the second PDB. For more information on that, see the KV5x reference manual (document: KV5XP144M240RM).

3.1.3. SPI

The SPI module on KV5x is essentially unchanged from the version used on the 120MHz K10. However, it is important to note that there is a clocking change that will affect your code. This is described in section 3.1.3.1.

3.1.3.1. Clocking

The K10 SPI module's source clock is the Bus clock, which is restricted to a maximum of 75 MHz. On the KV5x device, however, the SPI modules are sourced by the Fast Peripheral clock. This clock is restricted to a maximum of 120 MHz and is not required to be slower than the Core/System clock as in the K10. The calculations for the SPI baud rate remain the same, but you may need to account for a higher frequency clock if your application configures MCGOUTCLK to the maximum achievable frequency.

3.1.4. WDOG

Although the WDOG programming models are the same between the two devices, the clocking between the two are not. Both devices still support the 1 kHz LPO clock as a source and both still support the Bus clock as a source. However in the case of the KV5x device, the Bus clock is sourced from the same clock divider as the Flash clock. Therefore, the Bus clock on KV5x will be limited to 27.5 MHz or less. If your application uses the Bus clock as a source for the WDOG, you may need to account for the slower clock frequency.

3.1.5. EWM

There are no differences between the EWMs on the K10 and KV5x devices. Your application code should migrate directly between the two devices without modification.

3.1.6. PDB

The PDB of KV5x device has compatible registers and programming model as the K10 device so your PDB driver code should not need any modifications. The trigger options and output options are slightly different. The KV5x also adds a PDB instance to provide extra triggering options and flexibility. It was mainly added to allow for triggering of the second HS ADC through the FTM modules if so desired. For the PDB trigger connection information of KV5x, see the PDB chapter in the KV5x reference manual (document: KV5XP144M240RM).

3.2. Modified modules

These modules are characterized in two different groups: peripherals with differences (additions and removals) and peripherals with additions only. The sections covering peripherals with differences will outline the differences. The sections covering the additions only will outline how these changes can benefit your application design.

3.2.1. Changed modules

The module listed in this section have at least one change to the peripheral IP (this does not include high level clocking changes).

3.2.1.1. ADC

The legacy 16-bit ADC module implements the following changes.

- Removes ADC1, ADC2, ADC3.
- Removes the PGA module.
- Changes in the clock selections.
- Changes in the trigger selections.
- Updated register bit attributes.

3.2.1.2. Removal of ADC instances and the PGA

If your application requires multiple ADC instances, you need to migrate to the HSADC as only one legacy 16-bit ADC instance is available on KV5x devices. If your application is using only one ADC instance and this ADC is not ADC0, you need to first migrate your application code to instance ADC0.

No PGA instances or equivalents are available on KV5x are available. Therefore, if your application was utilizing the PGA modules, you will need to remove this application code. If the application requires a PGA, an external PGA will need to be implemented.

3.2.1.3. Clock selection changes

The legacy 16-bit ADC is a SAR converter that utilizes a clock source to enable the conversions. The K10 device allows this main clock to be sourced by the bus clock, bus clock divided by two, an alternate

clock (OSC0ERCLK), and an Asynchronous clock (internal to the ADC peripheral). The KV5x removes the bus clock divided by two options and replaces it with a second alternate clock option.

Another consideration that must be taken into account is the fact that the bus clock on KV5x is now combined with the flash clock and as such, it is also limited to 27.5 MHz.

3.2.1.4. Trigger selections

The K10 device allowed for two main options to trigger the ADC (selected by ADC0ALTTRGEN bit):

- PDB
- Alternate trigger defined by the ADC0TRGSEL bit field (SIM_SOPT7[AD0TRGSEL])

The KV5x device allows for four main options.

- XBARA output 39.
- PDB0 channel 1 trigger.
- PDB1 channel 0 trigger.
- Alternate trigger defined by the ADC0TRGSEL bit field (SIM_ADCOPT[ADC0ALTTRGEN]).

KV5x and K10 have the same alternate triggers with the exception of the RTC alarm (option 0xC), RTC seconds (option 0xD), and the high-speed comparator 3 asynchronous interrupt (option 0xF). Instead, KV5x allows for the selection of the following options in those places:

- XBARA output 38 (0xC)
- NC (0xD)
- NC (0xF)

3.2.1.5. Register differences

Removed Registers:

- ADC PGA register (ADCx_PGA)

3.2.1.6. I^2C

The KV5x I^2C peripheral adds the following functionality.

- Start/Stop bit detection interrupt.
- STOP mode entry delay when an I2C transaction is in progress.

Additionally, the KV5x also has clocking changes that should be considered when transitioning from K10 to KV5x.

3.2.1.7. Clocking changes

The I^2C module on KV5x is clocked by the bus/flash clock on KV5x. This differs from K10 because this clock is limited to 27.5 MHz. Therefore, the clock calculations in your application will need to be adjusted accordingly. There is no change to the formula to calculate the baud rate and hold times. The divider values have also not been affected by the updates to the I^2C peripheral.

3.2.1.8. Register changes

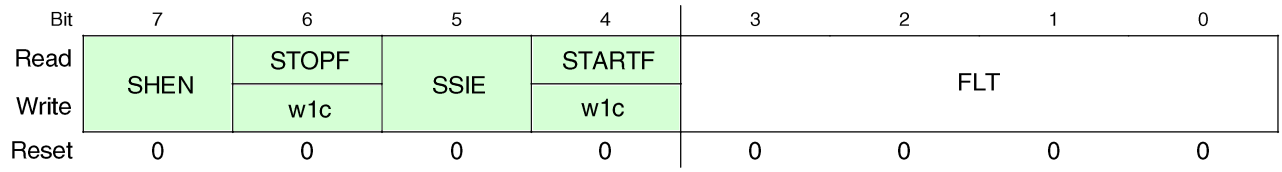


Figure 22. I²C Programmable Input Glitch Filter register (I2Cx_FLT) – KV5x

Added bits:

- **SHEN**: Stop Hold Enable–This bit enables/disables the delay of entry to stop mode when any data transmission or reception is occurring.
- **STOPF**: I²C Bus Stop Detect Flag–This bit is set when the I²C bus's stop status is detected and must be cleared by software.
- **SSIE**: I²C Bus Stop or Start Interrupt Enable–This bit enables the interrupt for I²C bus stop or start detection.
- **STARTF**: I²C Bus Start Detect Flag–This bit is set when the I²C bus's start status is detected and must be cleared by software.

3.2.1.9. UART

Basic functionality of the KV5x UART has not changed. There are clocking and CEA709.1-B changes of which you should be aware. First, the clocking changes will be discussed.

3.2.1.10. Clocking changes

The UARTs on KV5x are clocked by the Fast Peripheral clock. This clock is sourced from OUTDIV2 in the SIM module. The restrictions on this clock are less restrictive than on the K10 only in the fact that it can operate at a higher frequency than the System/Core clock. It must still be an integer multiple or divide of the System /Core clock, so your application code may work without any modifications. Although, you may need to adjust your fast peripheral clock or UART configuration code if you are using a faster clock speed on your KV5x application than on your K10 application. It may also be advantageous to (and it is important) know that the UART clock can operate at higher frequencies on KV5x.

3.2.1.11. CEA709.1-B functionality

The CEA709.1-B functionality is enhanced with a greater resolution for the transmit and receive indeterminate timers. The transmit and receive Beta1 timers are added. The following figures show the differences between the memory maps for the K10 and KV5x.

Table 6. K10 memory map

Absolute address (hex)	Register name	Width (in bits)
4006_B024	UART CEA709.1-B Beta1 Timer (UART1_BIT)	8
4006_B030	UART CEA709.1-B Receive Indeterminate Time High (UART1_RIDTH)	8

4006_B031	UART CEA709.1-B Transmit Indeterminate Time High (UART1_TIDTH)	8

4006_B024	UART CEA709.1-B Interrupt Enable Register 0 (UART1_IE0)	8
4006_B030	UART CEA709.1-B Receive Indeterminate Time High (UART1_RIDTH)	8
4006_B031	UART CEA709.1-B Receive Indeterminate Time Low (UART1_RIDTL)	8
4006_B032	UART CEA709.1-B Transmit Indeterminate Time High (UART1_TIDTH)	8
4006_B033	UART CEA709.1-B Transmit Indeterminate Time Low (UART1_TIDTL)	8
4006_B034	UART CEA709.1-B Receive Beta1 Timer High (UART1_RB1TH)	8
4006_B035	UART CEA709.1-B Receive Beta1 Timer Low (UART1_RB1TL)	8
4006_B036	UART CEA709.1-B Transmit Beta1 Timer High (UART1_TB1TH)	8
4006_B037	UART CEA709.1-B Transmit Beta1 Timer Low (UART1_TB1TL)	8
4006_B038	UART CEA709.1-B Programmable register (UART1_PROG_REG)	8
4006_B039	UART CEA709.1-B State register (UART1_STATE_REG)	8

Figure 23. KV5x memory map

Detailed differences between the registers are shown below.

Bit	7	6	5	4	3	2	1	0
Read / Write	LBKDIE	RXEDGIE	SBNS	SBR				
Reset	0	0	0	0	0	0	0	0

Figure 24. UART Baud Rate register (UARTx_BDH) – KV5x

Added bits:

- **SBNS**: Stop Bit Number Select – Selects the number of stop bits used. Chose between 1 and 2.

Bit	7	6	5	4	3	2	1	0
Read	TDMAS	0	RDMAS	0	LBKDDMAS	0		
Write								
Reset	0	0	0	0	0	0	0	0

Figure 25. UART Control register 5 (UARTx_C5) – KV5x

Added bits:

- **LBKDDMAS**: LIN Break Detect DMA Select Bit – This bit configures the LIN break detect

flag to generate an interrupt or DMA request.

Bit	7	6	5	4	3	2	1	0
Read Write	WTE	CWTE	BWTE	INITDE	ADTE	GTVE	TXTE	RXTE
Reset	0	0	0	0	0	0	0	0

Figure 26. UART 7816 Interrupt Enable register (UARTx_IE7816) – KV5x

Added bits:

- **ADTE**: ATR Duration Timer Interrupt Enable – This bit configures the IS7816[ADT] flag to generate an interrupt.

Bit	7	6	5	4	3	2	1	0
Read	WT	CWT	BWT	INITD	ADT	GTV	TXT	RXT
Write	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

Figure 27. UART 7816 Interrupt Status register (UARTx_IS7816) – KV5x

Added bits:

- **ADT**: ATR Duration Time Interrupt – This bit indicates that the ATR duration time has exceeded the programmed value. Software must clear this bit by writing a 1 to it.

Bit	7	6	5	4	3	2	1	0
Read Write	PEIE	WBEIE	ISDIE	PRXIE	PTXIE	PCTEIE	PSIE	TXDIE
Reset	0	0	0	0	0	0	0	0

Figure 28. UART CEA709.1-B Interrupt Enable register (UARTx_IE) – KV5x

Added bits:

- **PEIE**: Preamble Error Interrupt Enable – Enables / disables the preamble error interrupt capability.

In addition to adding a bit to the UARTx_IE register, a second register is added to further expand the interrupt capabilities of the CEA709.1-B standard. This register is at the offset 0x24 and takes the place of the Beta1 Timer register (this register has been moved/expanded).

Bit	7	6	5	4	3	2	1	0
Read	0					RPLOFIE	CTXDIE	CPTXIE
Write								
Reset	0	0	0	0	0	0	0	0

Figure 29. UART CEA709.1-B Interrupt Enable register 0 (UARTx_IE0) – KV5x

Added bits:

- **RPLOFIE**: Receive Packet Length Overflow Interrupt Enable – Enables/disables the receive packet length overflow functionality.
- **CTXDIE**: Collision during transmission of byte sync or later packet Interrupt Enable – Enables/disables the interrupt function for collisions during transmission of byte sync or later

packets.

- **CPTXIE**: Collision during preamble transmission Interrupt Enable – Enables / disables the interrupt on collision during preamble transmission.

Bit	7	6	5	4	3	2	1	0
Read	0		LNF	RPLOF	CDET		TXDF	FE
Write			w1c	w1c	w1c		w1c	w1c
Reset	0	0	0	0	0	0	0	0

Figure 30. UART CEA709.1-B Status register (UARTx_S4) – KV5x

Added bits:

- **LNF: LON** Noise Flag – Indicates that noise is present during the sampling of received packets.

Changed bits:

- **RPLOF**: Received Packet Length Overflow Flag – This bit replaces the Initial Synchronization Fail Flag. It now indicates that the received packet length exceeds 255 bytes.
- **TXDF**: Transmission Delay Flag – This bit replaces the improper line code violation bit. It is asserted if a packet that is queued for transmission has been delayed because a receive packet starting coming in before the packet could be transmitted.

3.2.1.12. FlexCAN

The KV5x FlexCAN adds the following features:

- Flexible message buffers (MBs), totaling 64 message buffers of 8 bytes data length each, configurable as Rx or Tx.
- ISO 11898-1 standard compliance.
- DMA Support
- Low Power mode enhancements.
- Extended Bit Time Control

NOTE

The KV5x device adds a third FlexCAN module. If you are using NXP supplied header files, the changes should be relatively simple. If not (or if using assembly code), you will need to make note of the changes. The CAN instances are located at the addresses listed in Table 7.

Table 7. CAN instance addresses

CAN Instance	Address
CAN0	0x4002_4000
CAN1	0x4002_5000
CAN2	0x400A_4000

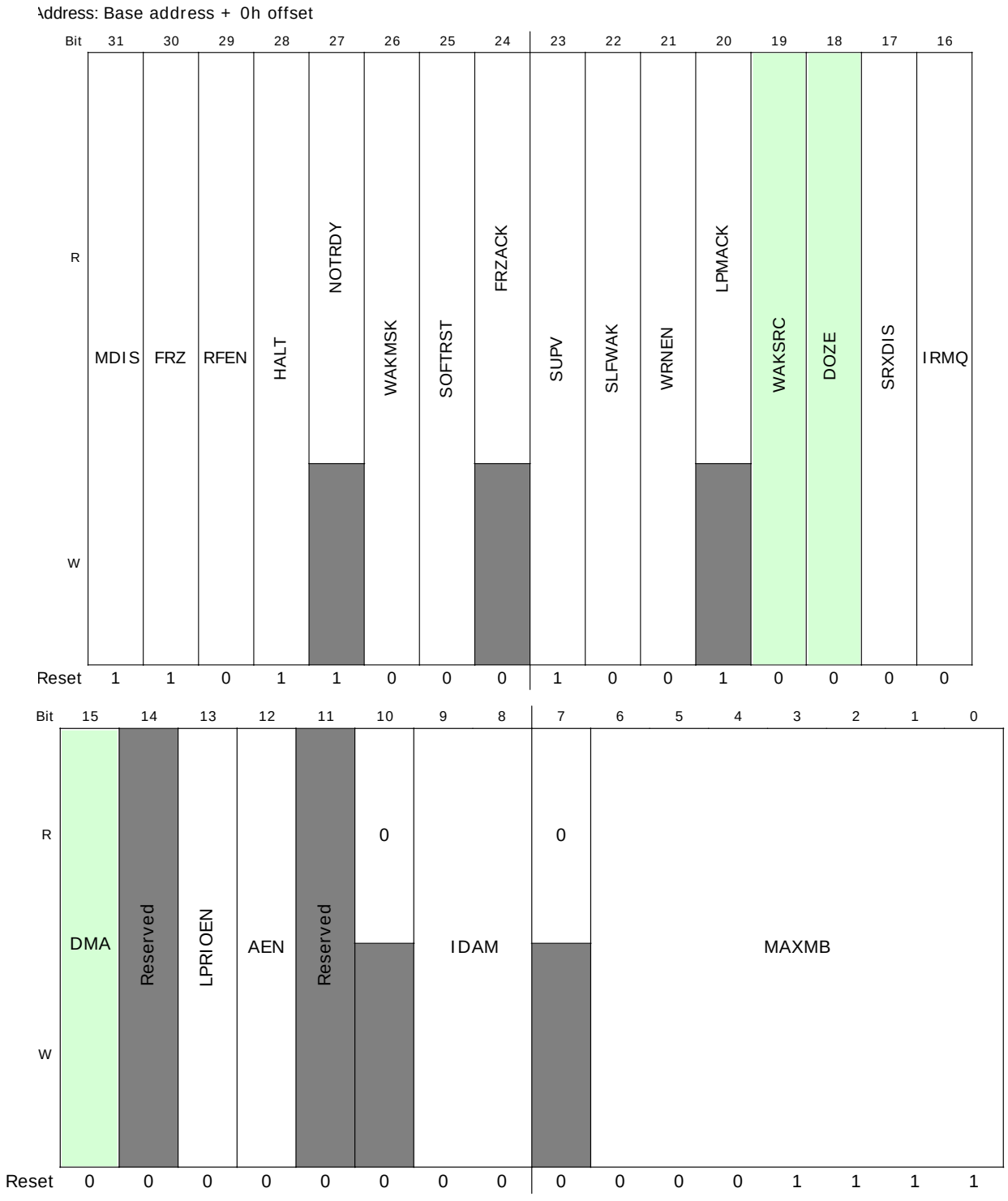


Figure 31. FlexCAN module configuration register (CANx_MCR) – KV5x

Added bits:

- **WAKSRC:** Wake Up Source – Enables / disables the integrated low-pass filter on the Rx CAN Input to protect from spurious wake up events.
- **DOZE:** Doze Mode Enable – Enables / disables the Doze mode functionality of the FlexCAN

module (note: Doze mode will only be entered when this bit is set and the software requests a low power mode entry of the SoC).

- **DMA:** DMA Enable – Enables / disables the DMA feature for the Rx FIFO.

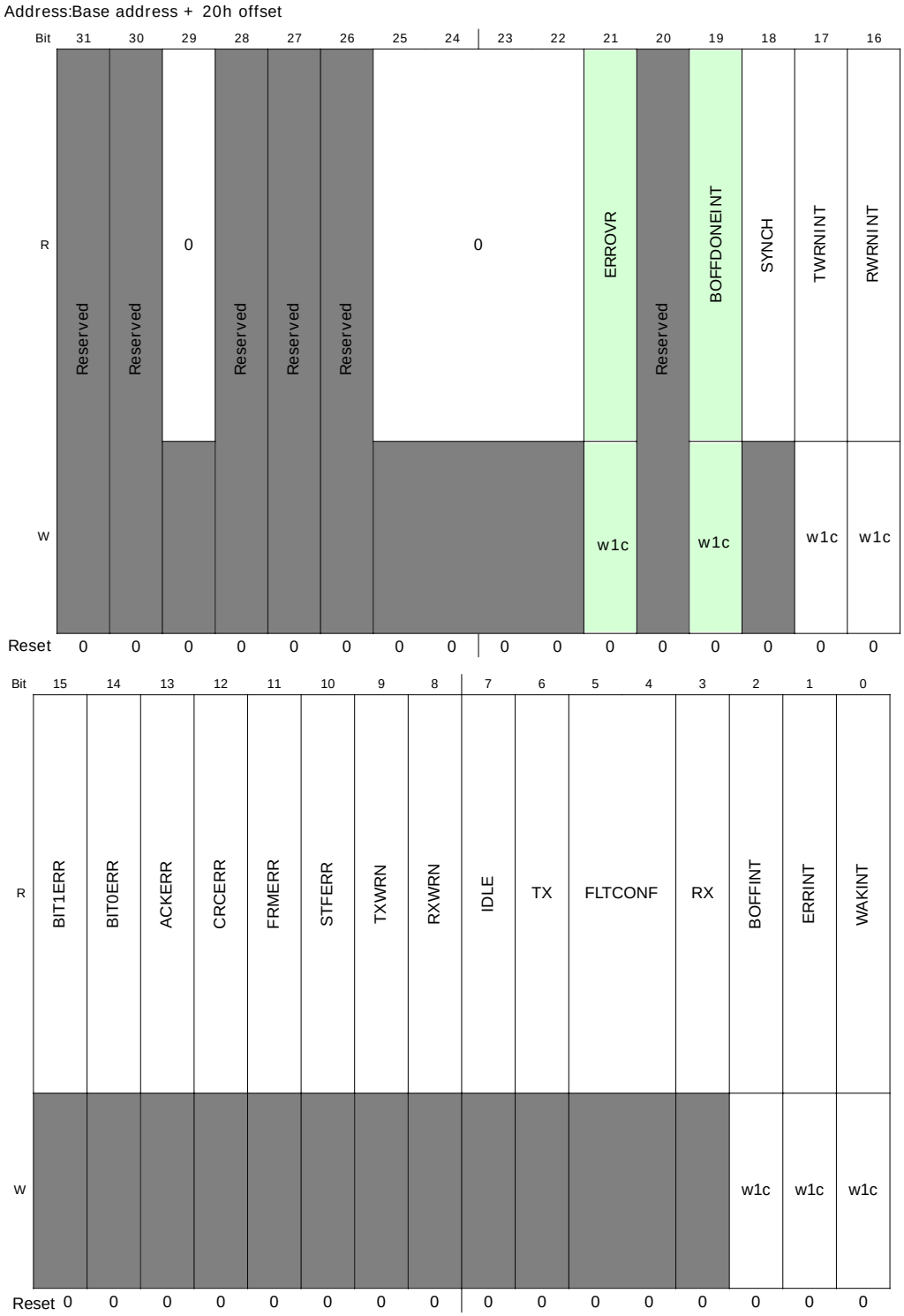


Figure 32. FlexCAN error and status 1 register (CANx_ESR1) – KV5x

Added Bits:

- **ERROVR:** Error Overrun bit – This bit indicates that an error condition occurred when any error flag is already set.
- **BOFFDONEINT:** Bus Off Done Interrupt – This bit is set when the Tx Error Counter (TXERRCNT) has finished counting 128 occurrences of 11 consecutive recessive bits on the CAN bus and is ready to leave Bus Off.

Address:Base address + 30h offset

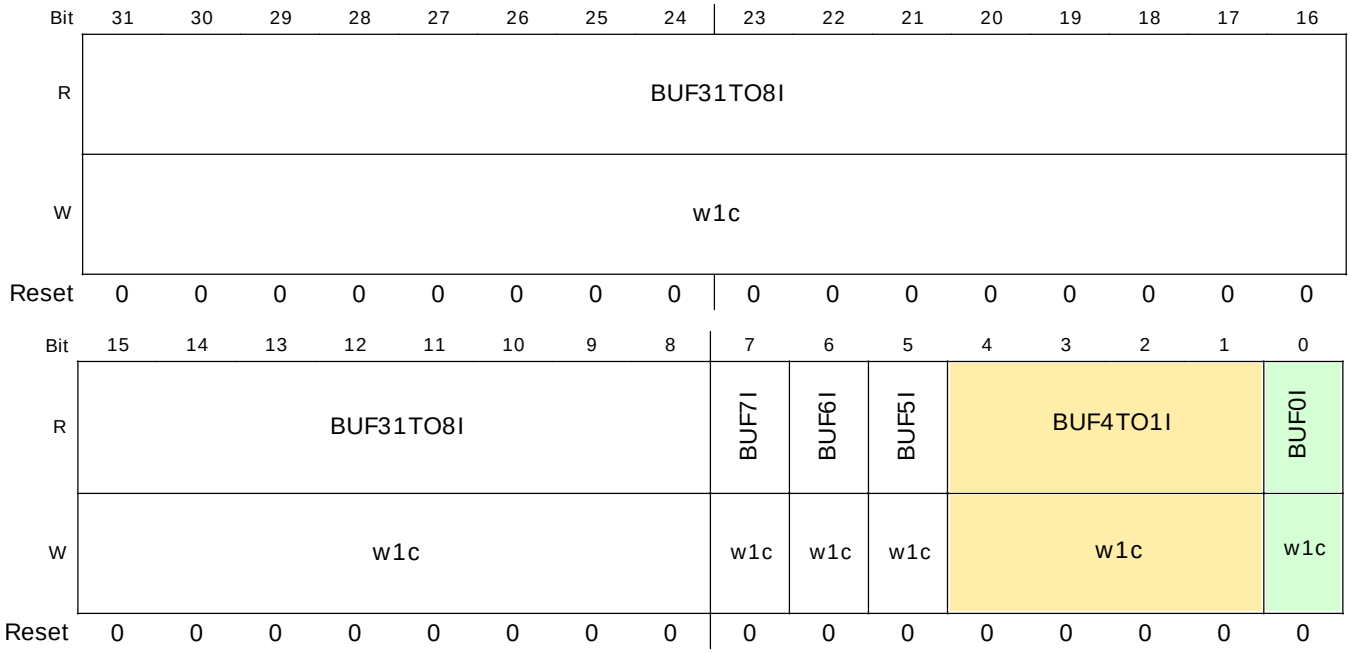


Figure 33. FlexCAN Interrupt Flags 1 register (CANx_IFLAG1) – KV5x

Changed bits:

- **BUF4TO1I:** This bit was changed from BUF4TO0I. It now controls the clearing of the interrupts for MBs 4 to 1 instead of 4 to 0.

Added bits:

- **BUF0I:** This bit indicates and clears the interrupt for MB 0. It was separated into its own bit because it now controls the clearing of the FIFO when the Rx FIFO is enabled.

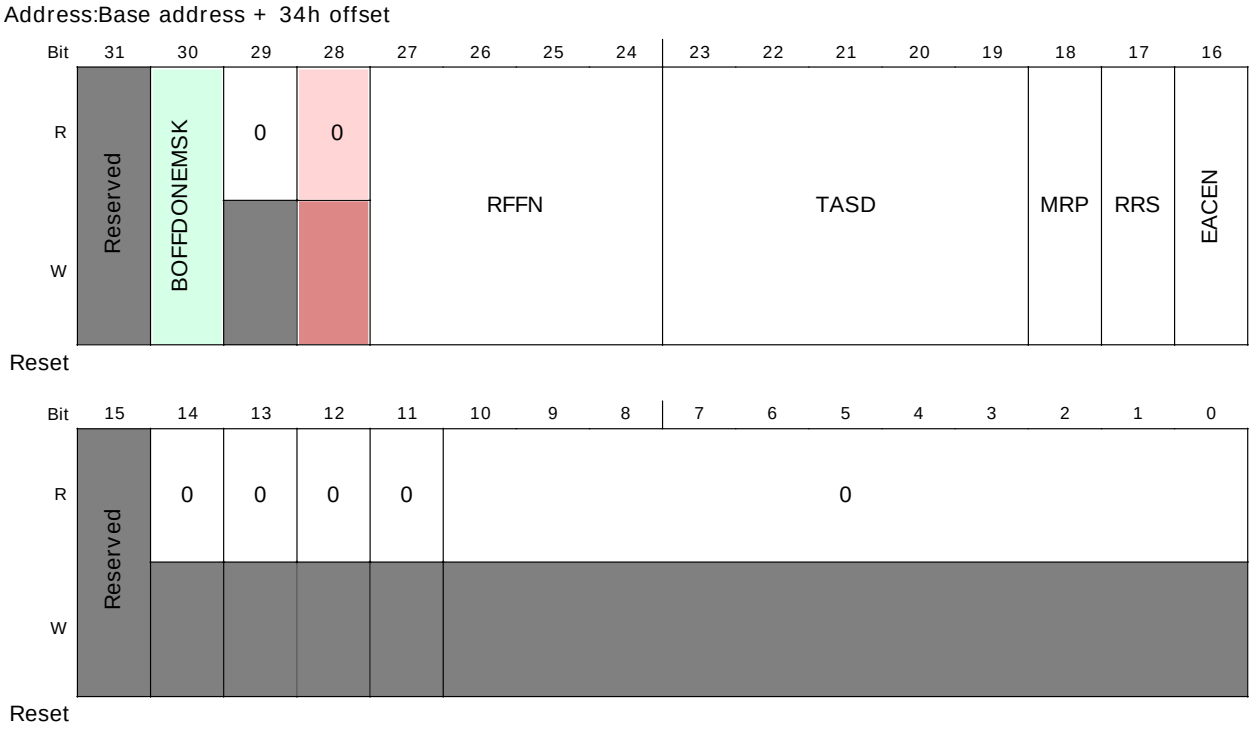


Figure 34. FlexCAN Control 2 register (CANx_CTRL2) – KV5x

Added bits:

- **BOFFDONEMSK:** Bus Off Done Interrupt Mask – This bit provides a mask for the Bus Off Done Interrupt.

Removed bits:

- **WRMFRZ:** Write-Access to Memory in Freeze Mode – Enables / disables write access to the FlexCAN memory in Freeze mode (no effect outside of Freeze mode).

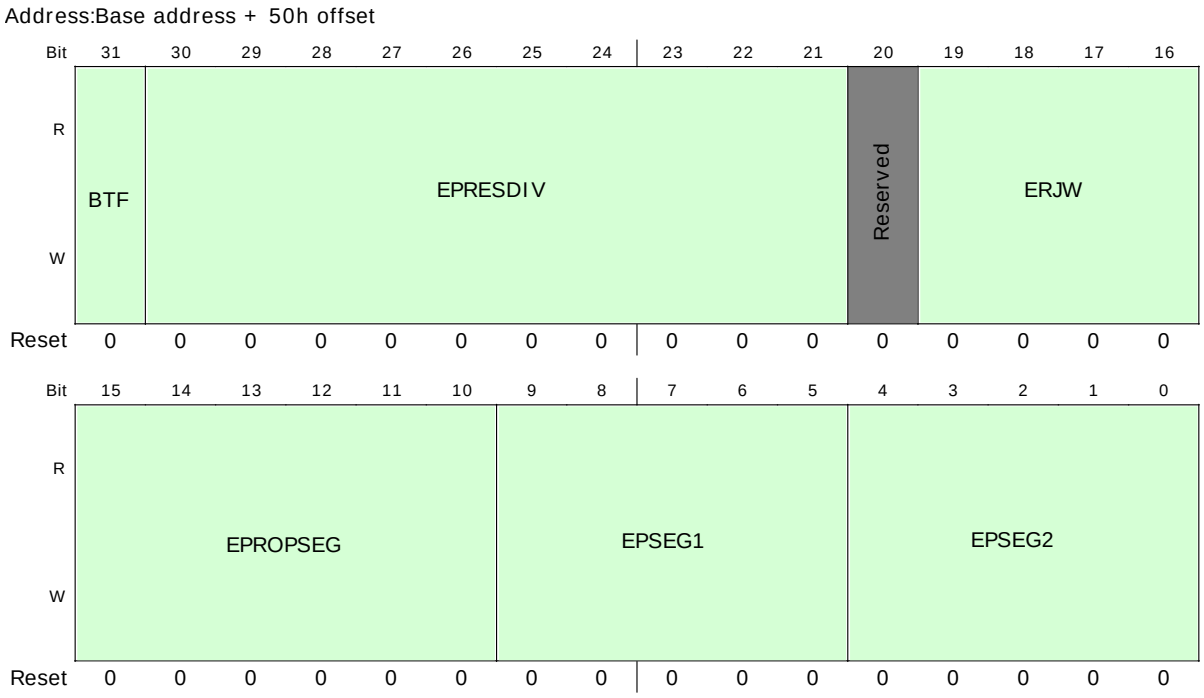


Figure 35. FlexCAN CAN Bit Timing register (CANx_CBT) – KV5x

Added Bits:

- **BTF:** Bit Timing Format Enable – This bit enables the use of the extended CAN bit timing fields EPRESDIV, EPROPSEG, EPSEG1, EPSEG2, and ERJW and can only be written in Freeze mode.
- **EPRESDIV:** Extended Prescale Division Factor – This bit defines the ratio between the PE clock frequency and the Serial Clock frequency and extends the CAN_CTRL1[PRESDIV] value range.
- **ERJW:** Extended Resync Jump Width – This bit field defines the maximum number of time quanta that a bit time can be changed by on synchronization (extends the CAN_CTRL1[RJW] value range.
- **EPROPSEG:** Extended Propagation Segment – This bit field defines the length of the Propagation Segment in the bit time (extends the CAN_CTRL1[PROPSEG] value range. It can only be written in Freeze mode.
- **EPSEG1:** Extended Phase Segment 1 – This bit field defines the length of Phase Segment 1 in the bit time (extends the CAN_CTRL1[PSEG1] value range). It can only be written in Freeze mode.
- **EPSEG2:** Extended Phase Segment 2 – This bit field defines the length of Phase Segment 2 in the bit time (extends the CAN_CTRL1[PSEG2] value range. It can only be written in Freeze mode.

Clocking Changes

Figure 36 and Figure 37 compare the clock control trees for the FlexCAN modules. First is the K10 clock tree and the second is the KV5x clock tree.

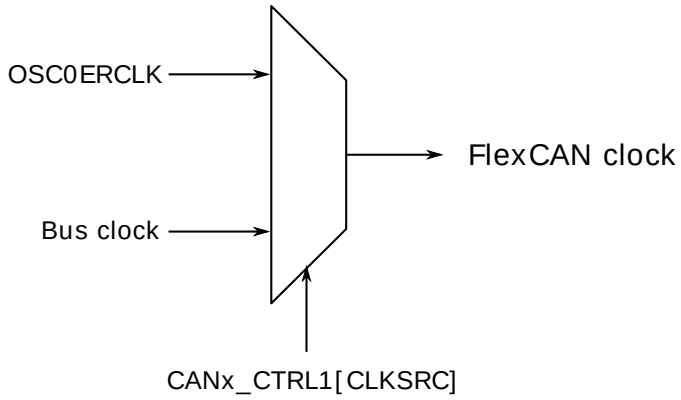


Figure 36. K10 FlexCAN clock tree

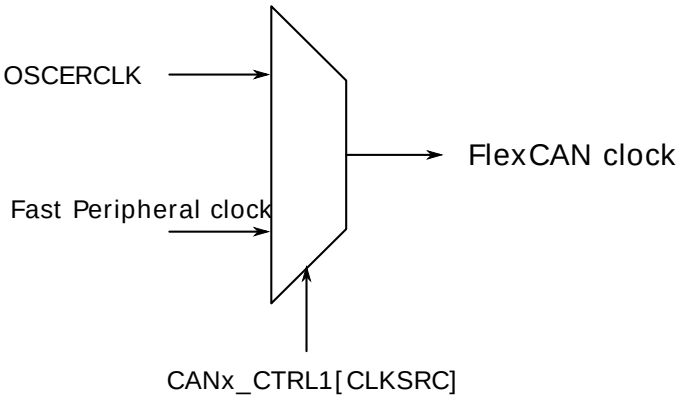


Figure 37. KV5x FlexCAN clock tree

As shown in Figure 36 and Figure 37, the only change is that the KV5x device replaces the Bus clock option with the Fast Peripheral clock option. This is important because this clock is sourced at a higher frequency than the bus clock. You will need to take this into account if you were using the bus clock in your application.

3.2.1.13. CRC

There is only one difference between the K10 CRC module and the KV5x CRC module. The CRC Data register in KV5x is named CRC_DATA while the data register in K10 is named CRC_CRC. Therefore, the only change your software will need is to rename the CRC_CRC accesses to CRC_DATA. Figure 39 illustrates the memory map comparison.

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value
4003_2000	CRC Data register (CRC_DATA)	32	R/W	FFFF_FFFFh
4003_2004	CRC Polynomial register (CRC_GPOLY)	32	R/W	0000_1021h
4003_2008	CRC Control register (CRC_CTRL)	32	R/W	0000_0000h

Figure 38. K10 CRC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value
4003_2000	CRC Data register (CRC_DATA)	32	R/W	FFFF_FFFFh
4003_2004	CRC Polynomial register (CRC_GPOLY)	32	R/W	0000_1021h
4003_2008	CRC Control register (CRC_CTRL)	32	R/W	0000_0000h

Figure 39. KV5x CRC memory map

3.2.1.14. GPIO and port control

There are only four major changes with respect to the GPIO and Port Control programming model.

- The PORT Mux field of KV5x was increased to allow for more options for each pin.
- KV5x adds Global Interrupt Control Registers.
- Bus clock has a much lower maximum speed which may affect digital filtering.
- Digital filtering is only available on PORT D pins.

Only software differences will be discussed in this section. Hardware differences and Mux selection differences will be discussed in later sections.

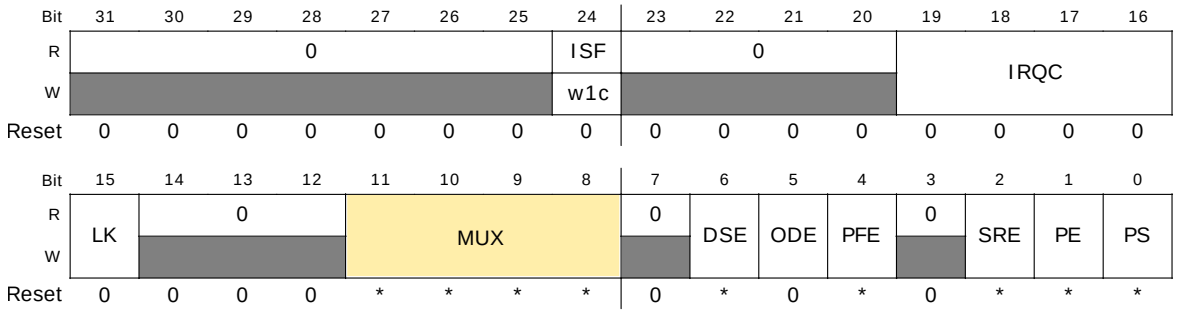


Figure 40. Pin Control register n (PORTx_PCRn) – KV5x

Changed bits:

- **MUX:** Pin Mux Control – This bit field was expanded to accommodate more options for each GPIO pin.

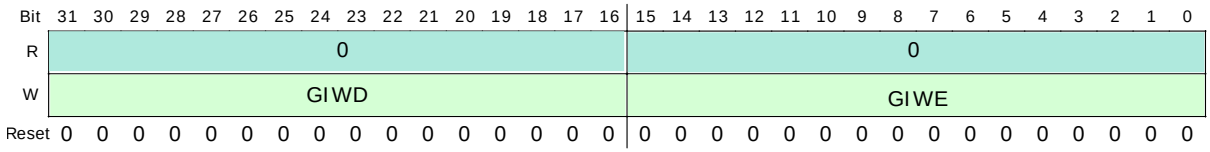


Figure 41. Global Interrupt Control Low register (PORTx_GICLR) – KV5x

Added bits:

- **GIWD:** Global Interrupt Write Data – The value written to this register will be written to all Pin Control Register bits that are selected by GIWE.
- **GIWE:** Global Interrupt Write Enable – Selects which Pin Control Registers (15 through 0) to update with the value in GIWD.

Address:Base address + 88h offset

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

Figure 42. Global Interrupt Control High register (PORTx_GICHR) – KV5x

Added bits:

- **GIWD:** Global Interrupt Write Data – The value written to this register will be written to all Pin Control Register bits that are selected by GIWE.
- **GIWE:** Global Interrupt Write Enable – Selects which Pin Control Registers (31 through 16) to update with the value in GIWD.

Software impact:

The main impact that should be taken into consideration is the expansion/changes of the MUX selections. For each pin used in your application, it is recommended that you check the table in the Hardware Impacts section of this document to find the new MUX selection for the desired functionality.

For applications which implement digital filtering, there are two items to consider:

- The bus clock, which optionally provides the clock for the digital filtering has the possibility of being much lower due to the clock frequency restriction of the KV5x bus clock
- PORT D pins are the only pins which support digital filtering

Therefore, it is recommended to double check that the pins which require digital filtering in your application are PORT D pins. You may also want to change your digital filtering settings as the bus clock is now limited to 27.5 MHz (will be the same frequency as the flash clock).

3.2.2. Additions

The peripherals listed in this section have only additions to their registers or memory maps. They can achieve the same functionality as in the 120MHz K10 but also include features that may enhance your application or ease coding.

3.2.2.1. PIT

The PIT module in KV5x adds the capability to chain timers together. This allows the timer to measure greater amounts of time. The registers that contain the additional features are shown below.

Address:4003_7000h base + 108h offset + (16d × i), where i=0d to 3d

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													CHN	TIE	TEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 43. PIT Timer Control register (PIT_TCTRLn) – KV5x

NOTE

When enabled, the timer which is chained will automatically chain to the previous timer, for example, PIT timer 2 chains to PIT timer 1.

Clocking Changes

The KV5x combines the bus clock with the flash clock. Therefore, the PIT is limited to a maximum of 27.5 MHz. You need to consider PIT limitation when migrating, as the K10 120MHz devices allow the PIT to be clocked to a maximum of 75 MHz.

3.2.2.2. FlexTimer

The FlexTimer adds counter reset upon input capture event.

Other than the above additional feature, the programming models of the two devices are the same. However, the base addresses are different. If you are using the NXP approved header files and macros, the same macros may be used and migration will be seamless. If you have written your application in assembly and/or programmed the base addresses manually, you need to know the base address changes listed in Table 8.

Table 8. Memory map comparison

FTM Instance	K10	KV5x
FTM0	0x4003_8000	0x4003_8000
FTM1	0x4003_9000	0x4003_9000
FTM2	0x400B_8000	0x4003_A000
FTM3	0x400B_9000	0x4002_6000

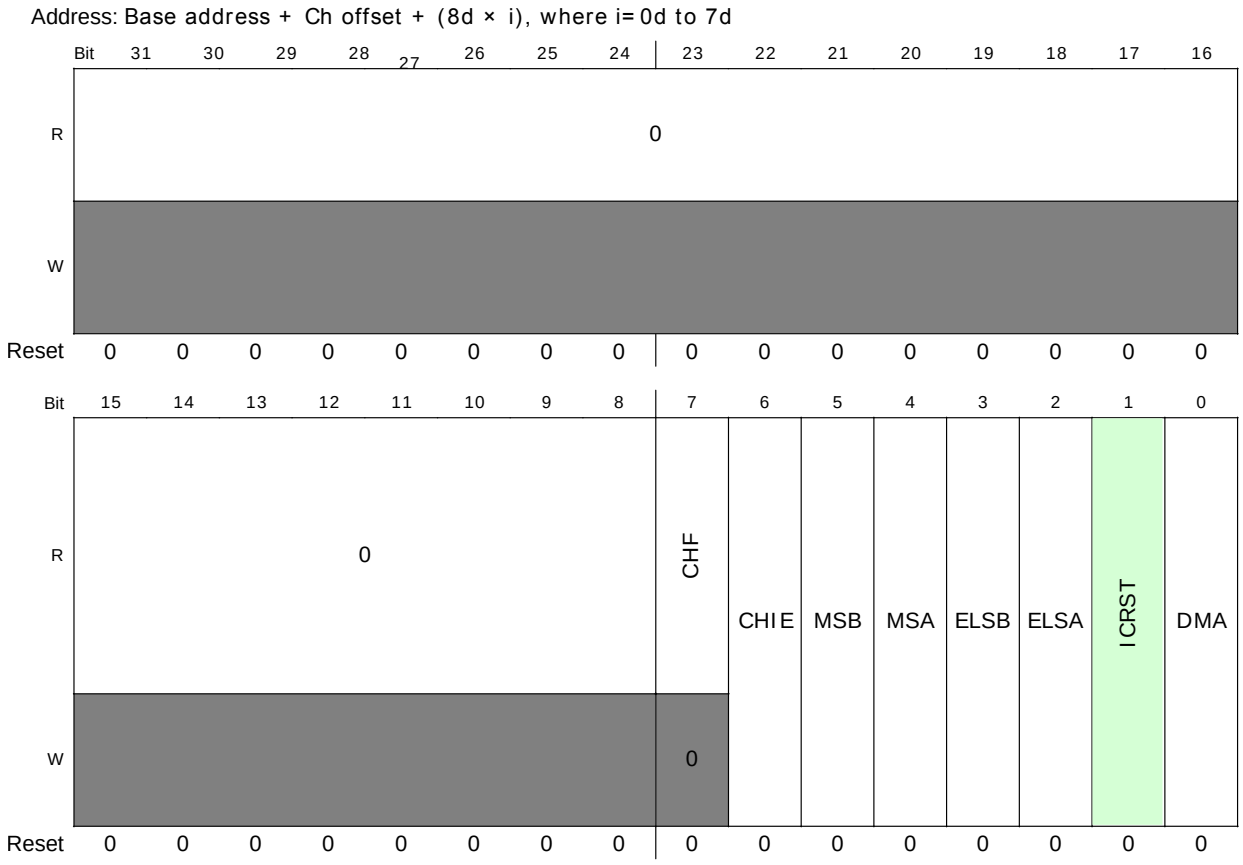


Figure 44. FTM Channel (n) Status and Control register (FTMx_CnSC) – KV5x

Added bits:

- **ICRST:** FTM Counter reset by the selected Input capture event – Initiates an FTM counter reset by the selected event of a channel (n) in the Input Capture Mode.

If your application is using input capture, this feature may reduce the calculations necessary for your application, as the input capture feature can then provide an absolute count value from the last capture event.

3.3. New modules

These are modules that are present on KV5x but not on K10. These modules are discussed because they may be necessary when migrating your application to a KV5x device (due to the lack of peripheral instances) or may be advantageous to use in your application.

3.3.1. HSADC

The HSADC is a new ADC to Kinetis devices. It is an ADC with 12-bit resolution capable of five mega samples per second. The KV5x device contains two instances of this device with each instance capable of scanning up to 16 dedicated channels sequentially. Each instance contains two independent conversion machines making simultaneous conversions possible. This makes the new ADC well suited for motor control applications. The new ADC contains the following features.

- 12/10/8/6 bits selectable resolution.
- Single-ended or differential inputs.
- Can be synchronized to other peripherals that are connected to an internal Inter-Peripheral Crossbar module through the SYNC0/1 input signal.
- Parallel synchronous scan mode (two ADCs converters scanning in parallel and simultaneously provide 16 measurements in 8 conversion times).
- Sequential scan mode (two ADCs can sequentially acquire 16 ADC measurements on one triggered event).
- Parallel independent scan mode (each ADC can independently acquire 8 ADC measurements triggered by software or hardware events).
- Pause control for conversion requests on the scan list.
- Optional interrupts at the end of scan for:
  - Out-of-range limit exceeded
  - Zero crossing
  - Calibration cycle ended
- Sample correction.
- DMA support.
- Signed or unsigned results.

3.3.1.1. Software impacts

If your application requires more than one ADC, then this ADC must be used. The programming model of the HSADC is completely different from the ADC. The best and most time efficient method of converting your application to the new HSADC is to use the Kinetis SDK HAL examples as a starting point.

3.3.1.2. Hardware impacts

The only hardware consideration taken into account is the removal of the external capacitor. The external capacitor is not recommended for the HSADC as this affects the sampling time requirements of the module. However an external capacitor could be left in place provided the RC time constant is less than the minimum sampling time.

3.3.2. Ethernet

The Ethernet module is not a new IP to Kinetis. If you plan to add Ethernet to your application, it is best to consult the Kinetis SDK HAL based Ethernet examples.

3.3.2.1. Hardware impacts

None.

3.3.2.2. Software impacts

None.

3.3.3. eFlexPWM

The eFlexPWM is not a peripheral that is available in the K series Kinetis devices and the KV5x devices contain an FTM peripheral (which is available on the K series K10). The KV5x devices also contain the same number of FTM instances and channels. Therefore, when migrating from a K10 device to a KV5x device, there will be no software or hardware impacts. However, if you plan to use eFlexPWM, it is best to use the Kinetis SDK HAL based eFlexPWM examples to get started. For more information on how to use the eFlexPWM module, see the document AN5142: Features of the FlexTimer Module.

3.3.3.1. Hardware impacts

None.

3.3.3.2. Software impacts

None.

3.3.4. TRNG

The TRNG is a hardware accelerator module that generates a 512-bit random number as needed by modules or software routines that may need a random number. It is intended for direct use by functions that generate secret keys, per-message secrets, random challenges, and other similar quantities used in cryptographic algorithms.

3.3.4.1. Hardware impacts

None.

3.3.4.2. Software impacts

None.

4. Hardware comparison

The following section outlines the differences and hardware considerations when migrating from the K10 120 MHz device to the KV5x 240 MHz device.

4.1. Package/pinout differences

The K10 120 MHz device is only offered in a 144 LQFP package and a 144 MAPBGA package. The KV5x 240 MHz device was designed to pin-to-pin compatible with the 144 LQFP K10 device. Both 144 LQFP packages use the same package. The drawing for the package can be found here.

Since the KV5x was designed to pin-to-pin compatible, the hardware changes (with respect to the pin-out) will be minimal. Table 9 highlights the differences between the pin-outs. The white cells are pins/functions that do not change from device to device. The orange cells indicate pins that have the same function options, but minor code changes may be required to continue using that pin. The red cells highlight pins that are missing functionality on that pin.

Table 9. K10 120 MHz pins

Pkg pin	K10 pin name	0	1	2	3	4	5	6	7
1	PTE0	ADC1_SE4a	PTE0	SPI1_PCS1	UART1_TX	SDHC0_D1		I2C1_SDA	RTC_CLKOUT
2	PTE1/LLWU_P0	ADC1_SE5a	PTE1/LLWU_P0	SPI1_SOUT	UART1_RX	SDHC0_D0		I2C1_SCL	SPI1_SIN
3	PTE2/LLWU_P1	ADC1_SE6a	PTE2/LLWU_P1	SPI1_SCK	UART1_CTS	SDHC0_DCLK			
4	PTE3	ADC1_SE7a	PTE3	SPI1_SIN	UART1_RTS	SDHC0_CMD			SPI1_SOUT
5	VDD	0							
6	VSS	0							
7	PTE4/LLWU_P2	0	PTE4/LLWU_P2	SPI1_PCS0	UART3_TX	SDHC0_D3			
8	PTE5	0	PTE5	SPI1_PCS2	UART3_RX	SDHC0_D2		FTM3_CH0	
9	PTE6	0	PTE6	SPI1_PCS3	UART3_CTS	I2S0_MCLK		FTM3_CH1	
10	PTE7	0	PTE7		UART3_RTS	I2S0_RXD0		FTM3_CH2	
11	PTE8	ADC2_SE16	PTE8	I2S0_RXD1	UART5_TX	I2S0_RX_FS		FTM3_CH3	
12	PTE9	ADC2_SE17	PTE9	I2S0_TXD1	UART5_RX	I2S0_RX_BCLK		FTM3_CH4	
13	PTE10	0	PTE10		UART5_CTS	I2S0_TXD0		FTM3_CH5	
14	PTE11	ADC3_SE16	PTE11		UART5_RTS	I2S0_TX_FS		FTM3_CH6	
15	PTE12	ADC3_SE17	PTE12			I2S0_TX_BCLK		FTM3_CH7	
16	VDD								
17	VSS								
18	PTE16	ADC0_SE4a	PTE16	SPI0_PCS0	UART2_TX	FTM_CLKIN0		FTM0_FLT3	
19	PTE17	ADC0_SE5a	PTE17	SPI0_SCK	UART2_RX	FTM_CLKIN1		LPTMR0_ALT3	
20	PTE18	ADC0_SE6a	PTE18	SPI0_SOUT	UART2_CTS	I2C0_SDA			
21	PTE19	ADC0_SE7a	PTE19	SPI0_SIN	UART2_RTS	I2C0_SCL		CMP3_OUT	
22	VSSUSB								
23	ADC0_DP1/OP0_DP0	ADCx_DPx							
24	ADC0_DM1/OP0_DM0	ADCx_DMx							

Table 9. K10 120 MHz pins (contd…)

Pkg pin	K10 pin name	0	1	2	3	4	5	6	7
25	ADC1_DP1/OP1_DP0	ADCx_DPx							
26	ADC1_DM1/OP1_DM0	ADCx_DMx							
27	PGA0_DP/ADC0_DP0/ADC1_DP3	ADCx_DPx							
28	PGA0_DM/ADC0_DM0/ADC1_DM3	ADCx_DMx							
29	PGA1_DP/ADC1_DP0/ADC0_DP3	ADCx_DPx							
30	PGA1_DM/ADC1_DM0/ADC0_DM3	ADCx_DMx							
31	VDDA	VDDA							
32	VREFH	VREFH							
33	VREFL	VREFL							
34	VSSA	VSSA							
35	ADC1_SE16/OP1_OUT	ADC1_SE16/ CMP2_IN2							
36	ADC0_SE16/OP0_OUT	ADC0_SE16/ CMP1_IN2							
37	VREF_OUT	VREF_OUT/ ADC1_SE18/ CMP1_IN5/ CMP0_IN5							
38	DAC0_OUT	ADC0_SE2/CMP1_IN3/DAC0_OUT							
39	DAC1_OUT	ADC_SE/CMP_IN/DAC_OUT							
40	XTAL32	XTAL32							
41	EXTAL32	EXTAL32							
42	VBAT	VBAT							
43	VDD								
44	VSS								
45	PTE24	ADC0_SE17/ EXTAL1	PTE24	CAN1_TX	UART4_TX	I2S1_TX_FS		EWM_OUT	I2S1_RXD1
46	PTE25	ADC0_SE18/ XTAL1	PTE25	CAN1_RX	UART4_RX	I2S1_TX_BCLK		EWM_IN	I2S1_TXD1
47	PTE26	ADC3_SE5b	PTE26		UART4_CTS	I2S1_TXD0		RTC_COUT	
48	PTE27	ADC3_SE4b	PTE27		UART4_RTS	I2S1_MCLK			
49	PTE28	ADC3_SE7a	PTE28						
50	PTA0	TSI0_CH1	PTA0	UART0_CTS	FTM0_CH5				JTAG_TCLK
51	PTA1	TSI0_CH2	PTA1	UART0_RX	FTM0_CH6				JTAG_TDI
52	PTA2	TSI0_CH3	PTA2	UART0_TX	FTM0_CH7				JTAG_TDO

Table 9. K10 120 MHz pins (contd…)

Pkg pin	K10 pin name	0	1	2	3	4	5	6	7
53	PTA3	TSI0_CH4	PTA3	UART0_RTS	FTM0_CH0				JTAG_TMS
54	PTA4/ LLWU_P3	TSI0_CH5	PTA4/ LLWU_P3		FTM0_CH1				NMI_B
55	PTA5	0	PTA5		FTM0_CH2		CMP2_OUT	I2S0_TX_BCLK	JTAG_TRST
56	VDD								
57	VSS								
58	PTA6	ADC3_SE6a	PTA6		FTM0_CH3	I2S1_RXD0			TRACE_COUT
59	PTA7	ADC0_SE10	PTA7		FTM0_CH4	I2S1_RX_BCLK			TRACE_D3
60	PTA8	ADC0_SE11	PTA8		FTM1_CH0	I2S1_RX_FS		FTM1_QD_PHA	TRACE_D2
61	PTA9	ADC3_SE5a	PTA9		FTM1_CH1			FTM1_QD_PHB	TRACE_D1
62	PTA10	ADC3_SE4a	PTA10		FTM2_CH0			FTM2_QD_PHA	TRACE_D0
63	PTA11	ADC3_SE15	PTA11		FTM2_CH1			FTM2_QD_PHB	
64	PTA12	CMP2_IN	PTA12	CAN0_TX	FTM1_CH0			I2S0_TXD0	FTM1_QD_PHA
65	PTA13/ LLWU_P4	CMP2_IN	PTA13/ LLWU_P4	CAN0_RX	FTM1_CH1			I2S0_TX_FS	FTM1_PD_PHB
66	PTA14	CMP3_IN	PTA14	SPI0_PCS0	UART0_TX			I2S0_RX_BCLK	I2S0_TXD1
67	PTA15	CMP3_IN	PTA15	SPI0_SCK	UART0_RX			I2S0_RXD0	
68	PTA16	CMP3_IN	PTA16	SPI0_SOUT	UART0_CTS			I2S0_RX_FS	
69	PTA17	ADC1_SE17	PTA17	SPI0_SIN	UART0_RTS			I2S0_MCLK	
70	VDD								
71	VSS								
72	PTA18	EXTAL0	PTA18		FTM0_FLT2	FTM_CLKIN0			
73	PTA19	XTAL0	PTA19		FTM1_FLT0	FTM_CLKIN1		LPTMR0_ALT1	
74	RESET_b	Reset_b							
75	PTA24	CMP3_IN4	PTA24					FB_A29	
76	PTA25	CMP3_IN5	PTA25					FB_A28	
77	PTA26	ADC2_SE15	PTA26					FB_A27	
78	PTA27	ADC2_SE14	PTA27					FB_A26	
79	PTA28	ADC2_SE13	PTA28					FB_A25	
80	PTA29	ADC2_SE12	PTA29					FB_A24	
81	PTB0/ LLWU_P5	ADCx_SE8 / TSI	PTB0/ LLWU_P5	I2C0_SCL	FTM1_CH0			FTM1_QD_PHA	
82	PTB1	ADCx_SE9 / TSI	PTB1	I2C0_SDA	FTM1_CH1			FTM1_QD_PHB	
83	PTB2	ADC0_SE12 / TSI	PTB2	I2C0_SCL	UART0_RTS			FTM0_FLT3	

Table 9. K10 120 MHz pins (contd…)

Pkg pin	K10 pin name	0	1	2	3	4	5	6	7
84	PTB3	ADC0_SE13 / TSI	PTB3	I2C0_SDA	UART0_CTS			FTM0_FLT0	
85	PTB4	ADC1_SE10	PTB4					FTM1_FLT0	
86	PTB5	ADC1_SE11	PTB5					FTM2_FLT0	
87	PTB6	ADC1_SE12	PTB6				FB_AD23		
88	PTB7	ADC1_SE13	PTB7				FB_AD22		
89	PTB8		PTB8		UART3_RTS		FB_AD21		
90	PTB9	0	PTB9	SPI1_PCS1	UART3_CTS		FB_AD20		
91	PTB10	ADC1_SE14	PTB10	SPI1_PCS0	UART3_RX	I2S1_TX_BCLK	FB_AD19	FTM0_FLT1	
92	PTB11	ADC1_SE15	PTB11	SPI1_SCK	UART3_TX	I2S1_TX_FS	FB_AD18	FTM0_FLT2	
93	VSS								
94	VDD								
95	PTB16	TSI	PTB16	SPI1_SOUT	UART0_RX	I2S1_TXD0	FB_AD17	EWM_IN	
96	PTB17	TSI	PTB17	SPI1_SIN	UART0_TX	I2S1_TXD1	FB_AD16	EWM_OUT	
97	PTB18	TSI	PTB18	CAN0_TX	FTM2_CH0	I2S0_TX_BCLK	FB_AD15	FTM2_QD_PHA	
98	PTB19	TSI	PTB19	CAN0_RX	FTM2_CH1	I2S0_RX_FS	FB_OE_B	FTM2_QD_PHB	
99	PTB20	ADC2_SE4a	PTB20	SPI2_PCS0			FB_AD31	CMP0_OUT	
100	PTB21	ADC2_SE5a	PTB21	SPI2_SCK			FB_AD30	CMP1_OUT	
101	PTB22	0	PTB22	SPI2_SOUT			FB_AD29	CMP2_OUT	
102	PTB23	0	PTB23	SPI2_SIN	SPI0_PCS5		FB_AD28	CMP3_OUT	
103	PTC0	ADC0_SE14 / TSI0_CH13	PTC0	SPI0_PCS4	PDB0_EXT		FB_AD14	I2S0_TXD1	
104	PTC1 / LLWU_P6	ADC0_SE15/ TSI0_CH14	PTC1 / LLWU_P6	SPI0_PCS3	UART1_RTS	FTM0_CH0	FB_AD13	I2S0_TXD0	
105	PTC2	ADC0_SE4b/ CMP1_IN0/ TSI0_CH15	PTC2	SPI0_PCS2	UART1_CTS	FTM0_CH1	FB_AD12	I2S0_TX_FSS	
106	PTC3 / LLWU_P7	CMP1_IN1	PTC3 / LLWU_P7	SPI0_PCS1	UART1_RX	FTM0_CH2		I2S0_TX_BCLK	
107	VSS	0							
108	VDD	0							
109	PTC4 / LLWU_P8	0	PTC4 / LLWU_P8	SPI0_PCS0	UART1_TX	FTM0_CH3	FB_AD11	CMP1_OUT	I2S1_TX_BCLK
110	PTC5 / LLWU_P9	0	PTC5 / LLWU_P9	SPI0_SCK	LPTMR0_ALT2	I2S0_RXD0	FB_AD10	CMP0_OUT	I2S1_TX_FS
111	PTC6 / LLWU_P10	CMP0_IN0	PTC6 / LLWU_P10	SPI0_SOUT	PDB0_EXT	I2S0_RX_BCLK	FB_AD9	I2S0_MCLK	

Table 9. K10 120 MHz pins (contd…)

Pkg pin	K10 pin name	0	1	2	3	4	5	6	7
112	PTC7	CMP0_IN1	PTC7	SPI0_SIN		I2S0_RX_FS	FB_AD8		
113	PTC8	ADC1_SE4b / CMP0_IN2	PTC8		FTM3_CH4	I2S0_MCLK	FB_AD7		
114	PTC9	ADC1_SE5b / CMP0_IN3	PTC9		FTM3_CH5	I2S0_RX_BCLK	FB_AD6	FTM2_FLT0	
115	PTC10	ADC1_SE6b	PTC10	I2C1_SCL	FTM3_CH6	I2S0_RX_FS	FB_AD5	I2S1_MCLK	
116	PTC11 / LLWU_P11	ADC1_SE7b	PTC11 / LLWU_P11	I2C1_SDA	FTM3_CH7	I2S0_RXD1	FB_RW		
117	PTC12	0	PTC12		UART4_RTS		FB_AD27	FTM3_FLT0	
118	PTC13	0	PTC13		UART4_CTS		FB_AD26		
119	PTC14	0	PTC14		UART4_RX		FB_AD25		
120	PTC15	0	PTC15		UART4_TX		FB_AD24		
121	VSS	0							
122	VDD	0							
123	PTC16	0	PTC16	CAN1_RX	UART3_RX		FB_CS5		
124	PTC17	0	PTC17	CAN1_TX	UART3_TX		FB_CS4		
125	PTC18	0	PTC18		UART3_RTS		FB_CS2		
126	PTC19	0	PTC19		UART3_CTS		FB_CS3	FB_TA	
127	PTD0 / LLWU_P12	0	PTD0 / LLWU_P12	SPI0_PCS0	UART2_RTS	FTM3_CH0	FB_ALE	I2S1_RXD1	
128	PTD1	ADC0_SE5b	PTD1	SPI0_SCK	UART2_CTS	FTM3_CH1	FB_CS0	I2S1_RXD0	
129	PTD2 / LLWU_P13	0	PTD2 / LLWU_P13	SPI0_SOUT	UART2_RX	FTM3_CH2	FB_AD4	I2S1_RX_FS	
130	PTD3	0	PTD3	SPI0_SIN	UART2_TX	FTM3_CH3	FB_AD3	I2S1_RX_BCLK	
131	PTD4/ LLWU_P14	0	PTD4/ LLWU_P14	SPI0_PCS1	UART0_RTS	FTM0_CH4	FB_AD2	EWM_IN	
132	PTD5	ADC0_SE6b	PTD5	SPI0_PCS2	UART0_CTS	FTM0_CH5	FB_AD1	EWM_OUT	
133	PTD6/ LLWU_P15	ADC0_SE7b	PTD6/ LLWU_P15	SPI0_PCS3	UART0_RX	FTM0_CH6	FB_AD0	FTM0_FLT0	
134	VSS	0							
135	VDD	0							
136	PTD7	0	PTD7	CMT_IRQ	UART0_TX	FTM0_CH7		FTM0_FLT1	
137	PTD8	0	PTD8	I2C0_SCL	UART5_RX			FB_A16	
138	PTD9	0	PTD9	I2C0_SDA	UART5_TX			FB_A17	
139	PTD10	0	PTD10		UART5_RTS			FB_A18	

Table 9. K10 120 MHz pins (contd…)

Pkg pin	K10 pin name	0	1	2	3	4	5	6	7
140	PTD11	0	PTD11	SPI2_PCS0	UART5_CTS	SDHC0_CLKIN		FB_A19	
141	PTD12	0	PTD12	SPI2_SCK	FTM3_FLT	SDHC0_D4		FB_A20	
142	PTD13	0	PTD13	SPI2_SOUT		SDHC0_D5		FB_A21	
143	PTD14	0	PTD14	SPI2_SIN		SDHC0_D6		FB_A22	
144	PTD15	0	PTD15	SPI2_PCS1		SDHC0_D7		FB_A23	

4.2. GPIO considerations

The most significant consideration made when migrating from K10 to KV58/KV56 is that the KV5x family pins are not 5 V tolerant. Therefore, care must be taken to limit the voltage at the pin to VDD +/- 0.3V.

While your hardware design may not be affected, it is good to know that the port pads of the KV5x are different from the port pins of K10. Therefore the slew rates will be different and may affect the operation of your application hardware. Consult the datasheet for more specifics of the slew rate.

4.3. Clocking considerations

There are no additional hardware considerations when using an external crystal to clock your system. The hardware circuitry used for K10 is compatible with the KV5x circuitry. However, if you are using a canned oscillator input or other external clock source, the maximum input clock frequency in the external clock mode is 48 MHz on KV5x and not 60 MHz limit as on K10.

5. Revision history

Table 10. Revision history

Revision number	Date	Substantive changes
0	10/2015	Initial release
1	06/2016	Changed KV5x-120 MHz to KV5x-240 MHz

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number: AN5117
Rev. 1
06/2016