MPC5604P MCU Family Device Errata

Introduction

This device errata applies to the following products:

- MPC5604P

e3100PS :

Register Protection on full CMU_CSR

Description

The register protection on CMU_CSR of CMU0 works only on the full 32 bit, while it should protect only the bits 24-31.
As a consequence, when register protection is active on CMU_CSR the frequency meter cannot be used anymore.
This issue is also present on CMU1.

Workarounds

In order to perform a frequency meter operation, the register protection of the relevant CMU must be disabled first; this workaround would work only when soft lock is active.

e3263PS :

Serial Boot and Censorship: flash read access

Description

In a secured device, starting with a serial boot, it is possible to read the content of the four flash locations where the RCHW is stored.
For example if the RCHW is stored at address 0x00000000, the reads at address 0x00000000, 0x00000004, 0x00000008 and 0x0000000C will return a correct value.
Any other flash address is not readable.

Workarounds

None

e3660PS :

MCM: MRSR does not report Power On Reset event

Description

The flag POR of MRSR register stays low after Power On Reset event on the device.

Workarounds

Do not use MRSR[POR] to determine Power on reset cause. Use RGM_DES instead.

e6475PS :

Device ADC electrical stress during power-up

Description

During power-up/power-down sequence, if using separate supplies for VDD_HV_ADC and VDD_HV, an error condition may occur when VDD_HV_ADC is at a lower voltage than VDD_LV. This may result in VDD_HV_ADC domain temporarily supplying current through VDD_LV.

Workarounds

Ensure that VDD_HV and VDD_HV_ADC remain at a higher voltage than [VDD_LV + threshold voltage(0.6v)] during power-up sequence.

e9090PS :

ESD-MM limited to 150V protection

Description

ESD-MM protection is limited to 150V, rather than 200V specification.

Workarounds

None

e9097PS :

FlexRay: The mechanisms provided by SFTCCSR are not functional.

Description

The application can not read the sync frame ID and deviation tables into the system RAM using the mechanisms provided by the Sync Frame Table Configuration, Control, Status Register (SFTCCSR).

Note: This is only related to making this information available to the application and does not affect the operation of the FlexRay communication controller module. The actual content of those tables is correct within the FlexRay module and the related clock synchronization calculation is performed correctly.

Workarounds

None

e4783PS :

SWT: Watchdog is disabled during BAM execution

Description

The watchdog is disabled at the start of BAM execution. In the case of an unexpected issue during BAM execution the CPU may be stalled and it will be necessary to generate an external reset to recover.

Workarounds

None

e7073PS :

BAM: Code download via FlexCAN not functioning in a CAN network

Description

When the serial download via FlexCAN is selected setting the FAB (Force Alternate Boot) pin, and ABS (Alternate Boot Selector) pins (ABS0 = 1 and ABS1 = 0) and the micro is part of a CAN network, the serial download protocol may unexpectedly stop in case of CAN traffic.
After the code has been downloaded, the BAM tries to disable the FLexCAN module writing the MCR (Module Configuration Register) without waiting for the acknowledge bit LPM_ACK (Low Power Mode Acknowledge) to be set. The FlexCAN cannot enter the low power mode until all current transmissions or receptions have finished. Further writings into any FlexCAN register may cause the low power mode not to be entered and, as consequence, the BAM to stop.

Workarounds

Since the higher the traffic, the higher the chance for the BAM to try to disable the FlexCAN module during a CAN frame reception, make sure that no other CAN frame is sent until the code download protocol has been completed.

e6934IPG : DSPI: Changing CTARs between frames in continuous PCS mode causes error

Description

Erroneous data could be transmitted if multiple Clock and Transfer Attribute Registers (CTAR) are used while using the Continuous Peripheral Chip Select mode (DSPIx_PUSHR[CONT=1]). The conditions that can generate an error are:

1) If DSPIx_CTARn[CPHA]=1 and DSPIx_MCR[CONT_SCKE = 0] and DSPIx_CTARn[CPOL, CPHA, PCSSCK or PBR] change between frames.

2) If DSPIx_CTARn[CPHA]=0 or DSPIx_MCR[CONT_SCKE = 1] and any bit field of DSPIx_CTARn changes between frames except DSPIx_CTARn[PBR].

Workarounds

When generating DSPI bit frames in continuous PCS mode, adhere to the aforementioned conditions when changing DSPIx_CTARn bit fields between frames.

e10483IPG : DSPI: PCS Continuous Selection Format limitation

Description

When the DSPI module has more than one entry in the TX FIFO and only one entry is written and that entry has the CONT bit set, and continuous SCK clock selected the PCS levels may change between transfer complete and write of the next data to the DSPI_PUSHR register.

For example:
If the CONT bit is set with the first PUSHR write, the PCS de-asserts after the transfer because the configuration data for the next frame has already been fetched from the next (empty) fifo entry. This behavior continues till the buffer is filled once and all CONT bits are one.

Workarounds

To ensure PCS stability during data transmission in Continuous Selection Format and Continuous SCK clock enabled make sure that the data with reset CONT bit is written to DSPI_PUSHR register before previous data sub-frame (with CONT bit set) transfer is over.

e12733IPG : FlexRay: Message Buffer can not be disabled and not locked after CHI command FREEZE

Description

If a complete message was transmitted from a transmit message buffer or received into a message buffer and the controller host interface (CHI) command FREEZE is issued by the application before the end of the current slot, then this message buffer can not be disabled and locked until the module has entered the protocol state normal active.

Consequently, this message buffer can not be disabled and locked by the application in the protocol config state, which prevents the application from clearing the commit bit CMT and the module from clearing the status bits. The configuration bits in the Message Buffer Configuration, Control, Status Registers (MBCCSRn) and the message buffer configuration registers MBCCFRn, MBFIDRn, and MBIDXRn are not affected.

At most one message buffer per channel is affected.

Workarounds

There are two types of workaround.

1) The application should not send the CHI command FREEZE and use the CHI command HALT instead.

2) Before sending the CHI command FREEZE the application should repeatedly try to disable all message buffers until all message buffers are disabled. This maximum duration of this task is three static or three dynamic slots.

e27514IPG : FlexRay: Incomplete Transmission of Message Frame in Key Slot

Description

The FlexRay module will transmit an incomplete message in the key slot under the following circumstances:

1.) The transmit message buffer n assigned to the key slot is located in the message buffer segment 2, i.e. FR_MBSSUTR[MB_LAST_SEG1] < n, and

2.) the data size of the message buffer segment 1 is smaller than the static payload length, i.e. FR_MBDSR[MBSEG1DS] < PCR19[payload_length_static].

In this case, the FlexRay module will transmit only FR_MBDSR[MBSEG1DS] payload words from message buffer n. The remaining words are padded with 0's.

Workarounds

The transmit message buffer assigned to key slot must be located in message buffer segment 1.

e14593IPG : FlexCAN: Global Masks misalignment

Description

Convention: MSB=0.

During CAN messages reception by FlexCAN, the RXGMASK (Rx Global Mask) is used as acceptance mask for most of the Rx Message Buffers (MB). When the FIFO Enable bit in the FlexCAN Module Configuration Register (CANx_MCR[FEN], bit 2) is set, the RXGMASK also applies to most of the elements of the ID filter table. However there is a misalignment between the position of the ID field in the Rx MB and in RXIDA, RXIDB and RXIDC fields of the ID Tables. In fact RXIDA filter in the ID Tables is shifted one bit to the left from Rx MBs ID position as shown below:

Rx MB ID = bits 3-31 of ID word corresponding to message ID bits 0-28
RXIDA = bits 2-30 of ID Table corresponding to message ID bits 0-28

Note that the mask bits one-to-one correspondence occurs with the filters bits, not with the incoming message ID bits.
This leads the RXGMASK to affect Rx MB and Rx FIFO filtering in different ways.

For example, if the user intends to mask out the bit 24 of the ID filter of Message Buffers then the RXGMASK will be configured as 0xffff_ffef. As result, bit 24 of the ID field of the incoming message will be ignored during filtering process for Message Buffers. This very same configuration of RXGMASK would lead bit 24 of RXIDA to be "don`t care" and thus bit 25 of the ID field of the incoming message would be ignored during filtering process for Rx FIFO.

Similarly, both RXIDB and RXIDC filters have multiple misalignments with regards to position of ID field in Rx MBs, which can lead to erroneous masking during filtering process for either Rx FIFO or MBs.

RX14MASK (Rx 14 Mask) and RX15MASK (Rx 15 Mask) have the same structure as the RXGMASK. This includes the misalignment problem between the position of the ID field in the Rx MBs and in RXIDA, RXIDB and RXIDC fields of the ID Tables.

Workarounds

Therefore it is recommended that one of the following actions be taken to avoid problems:

*) Do not enable the RxFIFO. If CANx_MCR[FEN]=0 then the Rx FIFO is disabled and thus the masks RXGMASK, RX14MASK and RX15MASK do not affect it.

*) Enable Rx Individual Mask Registers. If the Backwards Compatibility Configuration bit in the FlexCAN Module Configuration Register (CANx_MCR[BCC], bit 15) is set then the Rx Individual Mask Registers (RXIMR0-63) are enabled and thus the masks RXGMASK, RX14MASK and RX15MASK are not used.

*) Do not use masks RXGMASK, RX14MASK and RX15MASK (i.e. let them in reset value which is 0xffff_ffff) when CANx_MCR[FEN]=1 and CANx_MCR[BCC]=0. In this case, filtering processes for both Rx MBs and Rx FIFO are not affected by those masks.

*) Do not configure any MB as Rx (i.e. let all MBs as either Tx or inactive) when CANx_MCR[FEN]=1 and CANx_MCR[BCC]=0. In this case, the masks RXGMASK, RX14MASK and RX15MASK can be used to affect ID Tables without affecting filtering process for Rx MBs.

e498PS :

MC_RGM: Reset source flag not set correctly after previous clear

Description

Bits in the RGM_FES and RGM_DES registers may not show the correct status if a reset event occurs after its corresponding flag has previously been cleared. This error occurs only if no other MC_RGM register access has taken place after clearing the status flag and before the reset event.

Workarounds

Perform a MC_RGM register access (e.g. a 'dummy' read) directly after each write access of the RGM_FES and RGM_DES registers.

e2835PS :

MC_RGM: Clearing a flag at RGM_DES or RGM_FES register may be prevented by a reset

Description

Clearing a flag at RGM_DES and RGM_FES registers requires two clock cycles because of a synchronization mechanism. As a consequence if a reset occurs while clearing is on-going the reset may interrupt the clearing mechanism leaving the flag set.

Note that this failed clearing has no impact on further flag clearing requests.

Workarounds

No workaround for all reset sources except SW reset.

Note that in case the application requests a SW reset immediately after clearing a flag in RGM_xES the same issue may occur. To avoid this effect the application must ensure that flag clearing has completed by reading the RGM_xES register before the SW reset is requested.

e3492PS :

MC_RGM: Long Reset Sequence Occurs on 'Short Functional' Reset Event

Description

If a 'functional' reset source is configured to initiate a short reset sequence via setting of the appropriate RGM_FESS bit and also configured to assert the external reset pad via setting of the appropriate RGM_FBRE bit, its assertion will not initiate a short reset starting with PHASE3 but rather a long reset sequence starting with PHASE1.

Workarounds

Do not configure 'functional' resets that are configured to initiate a short reset sequence to also assert the external reset pad. In other words, if RGM_FESS[i] is '1', RGM_FBRE[i] should be '0'.

e4107PS :

MC_CGM and MC_PCU: A data storage exception is not generated on an access to MC_CGM or MC_PCU when the respective peripheral is disabled at MC_ME.

Description

If a peripheral with registers mapped to MC_CGM or MC_PCU address spaces is disabled via the MC_ME any read or write accesses to this peripheral will be ignored without producing a data storage exception.

Workarounds

For any mode other than a low-power mode do not disable any peripheral that is mapped to MC_CGM or

MC_PCU.

e4163PS :

MC_ME: ME_PSx registers may show '1' in a reserved bit field

Description

Some bits in the ME_PSx registers that are defined to always return the value '0' may instead return the value '1'.

Workarounds

It is recommended as a general rule that the User should always ignore the read value of reserved bit fields.

e5095PS :

MC_RGM: External Reset not asserted if Short Reset enabled

Description

For the case when the External Reset is enabled for a specific reset source at RGM_FBRE and a short reset is requested for the same reset source at RGM_FESS the External Reset is not asserted.

Workarounds

None

e5301PS :

MC_RGM: SAFE mode exit may be possible even though condition causing the SAFE mode request has not been cleared

Description

A SAFE mode exit should not be possible as long as any condition that caused a SAFE mode entry is still active. However, if the corresponding status flag in the RGM_FES register has been cleared, the SAFE mode exit may incorrectly occur even though the actual condition is still active.

Workarounds

Software must clear the SAFE mode request condition at the source before clearing the corresponding RGM_FES flag. This will ensure that the condition is no longer active when the RGM_FES flag is cleared and thus the SAFE mode exit can occur under the correct conditions.

e6041PS :

MC_ME: Peripherals in unknown state after SAFE mode exit

Description

Peripherals that reside in auxiliary clock domains may be in an unknown state after exiting the SAFE mode to enter the DRUN mode.

Workarounds

Execute a software reset while in the SAFE mode if one or more peripherals present in auxiliary clock domains is required for further operation.

e9367PS :

eTimer: Configuring an eTimer counter channel in GATED-COUNT mode or in SIGNED-COUNT mode may cause a possible incorrect counting of 1 tick.

Description

This bug affects eTimer in the following counting modes:
- GATED-COUNT mode, the CNTMODE field is '011' (count rising edges of primary source while secondary input is high);
-SIGNED-COUNT mode, the CNTMODE field is '101' (count primary source rising edges, secondary source specifies direction (up/down)).

Delays in the edge detection circuitry lead to a bug where the rising edge on the primary source is compared to the secondary source value one clock later in time. This means that if there is a rising edge on the primary source followed immediately by the secondary source going high, the eTimer logic could see this as a rising primary edge while the secondary is high even though the secondary input was low at the time of the rising primary edge.

This bug can occur when the transition on the secondary edge occurs within 1 IPBus clock cycle of the transition on the primary input.

The counter will also increment if the primary source is already high when the secondary source goes high.

Workarounds

The source selected as the secondary input to the eTimer channel needs to have an additional clock cycle of delay added to it. This can be done by using the input filters. For the primary source set FILT_PER==1 and FILT_CNT==0. For the secondary source set FILT_PER==1 and FILT_CNT==1. This will introduce a 5 clock cycle latency on the primary source and a 6 cycle latency on the secondary source which will properly align the two signals for count modes '011' and '101'.

Ensure that the primary source is low before the secondary source goes high to avoid a false count caused by the second form of this bug.

e9414PS :

Possible false DMA requests from eTimer.

Description

The sources of the eTimer's DMA requests are controlled by the DREQ[x] registers. If any of these registers have the same value, then multiple DMA requests can be generated in response to a single flag/condition.

Workarounds

Ensure that each of the DREQ[x] registers have a unique value prior to enabling DMA requests.

e7810IPG : NPC: Automatic clock gating does not work for MCKO_DIV = 8

Description

If the Nexus clock divider (NPC_PCR[MCKO_DIV]) in the Nexus Port Controller Port Configuration Register is set to 8 and the Nexus clock gating control (NPC_PCR[MCKO_GT]) is enabled, the nexus clock (MCKO) will be disabled prior to the completion of transmission of the Nexus message data.

Workarounds

Do not enable the automatic clock gating mode when the Nexus clock divider is set to 8. If Nexus clock gating is required, use a divide value of 1, 2 or 4 (set NPC_PCR[MCKO_GT]=0b1 and NPC_PCR[MCKO_DIV]=0b000, 0b001, or 0b011). However, MCKO must be kept below the maximum Nexus clock rate as defined in the device data sheet. If divide by 8 clock divider is required, then do not enable clock gating (set NPC_PCR[MCKO_GT]=0b0 and NPC_PCR[MCKO_DIV]=0b111).

e9376PS :

FlexPWM: Configuring the count value to set PWMA/PWMB first low and then high in the same cycle the output signals are low.

Description

The VAL2 and VAL4 registers define the turn-on edge and the VAL3 and VAL5 registers define the turn off edge of the PWMA/PWMB signals respectively. VAL3 cannot be less than VAL2 and VAL5 cannot be less than VAL4. Doing so will cause the PWM signal to turn off at the correct time (VAL3 or VAL5), but it will not turn on at the time defined by VAL2 or VAL4.
This can be an issue during the generation of phase delayed pulses where the PWM signal goes high late in PWM cycle N and remains high across the cycle boundary before going low early in cycle N+1 and goes high again in PWM cycle N+1. This errata will allow that to happen.
VAL3 register must be "greater than or equal to" VAL2 register and VAL5 must be "greater than or equal to" VAL4.

Workarounds

None

e6276IPG : JTAGC: EVTI and RDY require TCK to toggle

Description

The Nexus/JTAG Read/Write Access Control/Status Register (RWCS) write (to begin a read access) or the write to the Read/Write Access Data Register (RWD)(to begin a write access) does not actually begin its action until 1 JTAG clock (TCK) after leaving the JTAG Update-DR state. This prevents the access from being performed and therefore will not signal its completion via the READY (RDY) output unless the JTAG controller receives an additional TCK. In addition, EVTI is not latched into the device unless there are clock transitions on TCK.

Workarounds

The tool/debugger must provide at least one TCK clock for the EVTI signal to be recognized by the MCU. When using the RDY signal to indicate the end of a Nexus read/write access, ensure that TCK continues to run for at least 1 TCK after leaving the Update-DR state. This can be just a TCK with TMS low while in the Run-Test/Idle state or by continuing with the next Nexus/JTAG command. Expect the affect of EVTI and RDY to be delayed by edges of TCK. Note: RDY is not available in all packages of all devices.

e6309PS :

Debugging functionality could be lost when unsecuring a secured device.

Description

Providing the backdoor password via JTAG or via serial boot would unsecure the device, but on some devices may leave the Nexus interface and potentially the CPUs in an undetermined state.

Normal operation without a debugger is unaffected, debugging unsecured devices is also unaffected.

Workarounds

- A second connection attempt may be successful.
- Boot in serial mode (using the Flash password), then execute code which unsecures the device. (The JTAG interface needs to be inactive while the unsecure happens.)
- Implement a separate backdoor in application software. Once the software detects the custom backdoor sequence it can unlock the device via Flash write.
- Leave device unsecured for debugging.

e1844PS :

ADC: Minimum sampling time for ADC at 32MHz

Description

When ADC is running at 32MHz the minimum sampling time of 135ns specified is not met.

Workarounds

At 32MHZ the minimum sampling time must be at least 180ns.

e1861PS :

ADC: Offset Cancellation Not Required

Description

The offset cancellation mechanism does not improve the ADC performance as the intrinsic ADC precision is better than the offset cancellation resolution.

Workarounds

Do not use the offset cancellation feature as it will not improve ADC precision.

e3396PS :

ADC ABORT bit (single conversion)

Description

If the User starts one single ADC conversion and, after, starts a chain of conversions, when the User tries to abort one of the chained conversions, the abort command aborts the chain instead of a single conversion of the chain.

Workarounds

- A minimum of two conversions must be programmed.
- A dummy conversion must be started before or after a single conversion.

e3397PS :

ADC: Last conversion in chain not aborted

Description

If the user aborts the last ADC conversion of a chain the conversion appears to be aborted but the relevant data register is updated with the conversion data.

Workarounds

None

e3398PS :

ADC ABORT CHAIN bit

Description

If user aborts a chain of ADC conversions, the current conversion appears as aborted but the relative data register is however updated.

Workarounds

None

e3999PS :

ADC: Injected conversion not executed during scan mode.

Description

When ADC is converting a chain in scan mode -configured using NSTART bit in non-CTU mode operation; and a injected conversion arrives -triggered by software with JSTART bit or by hardware from eTimer_1 channel 5 (internal connection); the ADC gets stuck in the sampling phase (the triggered conversion is not executed and the chain is not restarted).

Workarounds

None

e4455PS :

ADC: conversion chain failing after ABORT chain

Description

During a chain conversion while the ADC is in scan mode when ADC_MCR[ABORTCHAIN] is asserted the current chain will be aborted as expected. However, in the next scan the first conversion of the chain is performed twice and the last conversion of the chain is not performed.

Workarounds

When aborting a chain conversion enable ADC_MCR[ABORTCHAIN] and disable ADC_MCR[START].

ADC_MCR[START] can be enabled when the abort is complete.

e5485PS :

ADC: ADC_DMAE[DCLR]should not be used

Description

When ADC_DMAE[DCLR] is set the DMA request should be cleared only after the data registers are read. However for this case the DMA request is automatically cleared and will not be recognised by the eDMA.

Workarounds

None

e9154PS :

ADC:Abort request during last sampling cycle corrupts the data register of next channel conversion

Description

If the abort pulse is valid in the last cycle of the SAMPLE phase, the current channel is correctly aborted but the data register (CDR[0..15]) of the next channel conversion shows an invalid value.

Workarounds

None

e8761PS :

MC_CGM: System clock may stop for case when target clock source stops during clock switching transition

Description

The clock switching is a two step process. The availability of the target clock is first verified. Then the system clock is switched to the new target clock source within two target clock cycles.

For the case when the FXOSC stops during the required two cycles, the switching process may not complete, causing the system clock to stop and prevent further clock switching. This may happen if one of the following cases occurs while the system clock source is switching to FXOSC:
- FXOSC oscillator failure
- SAFE mode request occurs, as this mode will immediately switch OFF the FXOSC (refer to ME_SAFE_MC register configuration)

Workarounds

The device is able to recover through any reset event ('functional', 'destructive', internal or external), so typically either the SWT (internal watchdog) will generate a reset or, in case it is used in the application, the external watchdog will generate an external reset. In all cases the devices will restart properly after reset.

To reduce the probability that this issue occurs in the application, disable SAFE mode transitions when the device is executing a mode transition with the FXOSC as the system clock source in the target mode.

e3052PS :

CMU XOSC Monitoring cannot be guaranteed when RCDIV>0 and FOSC is less than 1.5*Frc/2^rcdiv

Description

A False OLRI (Oscillator frequency less than RC frequency) event may be generated when FOSC is less than 1.5*(FRC / 2^RCDIV) and RCDIV>0, and not FRC/ 2^RCDIV as described in the Reference Manual).
Correct crystal clock monitoring is guaranteed when FOSC is strictly above 1.5*(FRC /2^RCDIV).

Workarounds

There are 2 workarounds available :
1. Keep RCDIV = 0
2. When RCDIV > 0, ensure FOSC is greater than FRC/2^(RCDIV -1) to avoid false OLRI (CMU external oscillator failure) event.

e5007PS :

FLASH: Programming a value over another value may not indicate an error if no bits are being programmed (1-->0)

Description

The Flash Programming/Erase Good Status Flag (PEG) in the Flash Module Configuration Register (Flash_MCR) may not indicate a failure to program if an attempt is made to program bits high (0b1) that are currently programmed low (0b0) if no bits in the 64-bit word or the 8 bit Error Correction Code (ECC) are being programmed low (to a 0b0).

For example, PEG will not indicate a failure (FLASH_MCR[PEG]=1) if an attempt is made to program the value 0x5555_FFFF over a flash location that already was programmed to 0x5555_5555. The flash contents will not be changed (stays 0x5555_5555) even though it should fail since there were no bits actually selected to be programmed (cleared).

However, PEG will correctly indicate a failure (FLASH_MCR[PEG]=0) if an attempt is made to program the value of 0x5555_00FF over a flash location that was already programmed to 0x5555_5555. The resulting flash contents will be 0x5555_0055.

NOTE: Reprogramming bits from a programmed state (0b0) to an erased state (0b1) is not supported on flash memory technology. A bit can only be set (0b1) by erasing the full flash block.

Workarounds

Do not expect an error to be flagged if there are no bits being cleared (0b0) in the word being programmed when programming a new value into a location in the flash that has already been programmed with a previous (not still erased) value.

e5008PS :

FLASH: Array Integrity Check does not check the first two double words

Description

The Array Integrity Check operation, started by setting the bit Array Integrity Enable (AIE) in the FLASH User Test 0 register (FLASH_UT0) on the selected and unlocked blocks, checks the content of all the blocks selected except for the first two double words of each block (offsets 0x0 and 0x8).
The first location checked of each block is at the offset 0x10.

Workarounds

The application should take care of checking the first two double words of each block selected for the Array Integrity Check operation.

e5034PS :

FLASH: SLL and HBL not properly initialized

Description

The Secondary Low/Mid Address Space Block Locking register (SLL) and the High Address Space Block Locking register (HBL) have a non-volatile image stored in Test Flash to allow a default state to be set by the user immediately following reset. These locations are write once only.

During boot, the register SLL and HBL are not loaded with the correct default value from the non-volatile location. This means that the default lock status of the Low, Mid, and High blocks are unknown.

Workarounds

The application should not rely on the default non-volatile value of SLL and HBL and must initialize both registers with the User defined values.

e5060PS :

FLASH: Double ECC errors of default Non Volatile image of BIU2 (PFCR2/PFAPR) are not checked

Description

The default value of the Flash BIU2 (PFCR2 in the MPC563xM/SPC563M, also named PFAPR in MPC560xB/C/P/S or SPC560B/C/P/S) register is loaded from a non-volatile memory (NVM) area. However, when this value is loaded into the BIU2 register, the presence of Double Errors in the Error Correction Code (ECC) value is not checked.

If the NVM area contains a double bit error, incorrect data will be loaded.
On the contrary single ECC errors are automatically corrected.

Workarounds

Do not rely on the contents of the BIU2 (PFCR2/PFAPR) to automatically be loaded into the register

correctly. Software should always load the User defined value manually into the register.

e5673PS :

FLASH: Margin Mode is not supported

Description

The flash contains a support for determining the program and erase margin (to being near the actual decision point for reading the bit as a high or low). However, this feature will incorrectly indicate bits in the flash that have plenty of margin as marginal and will not indicate that marginal bits are, in fact, marginal.

Workarounds

Verification of the flash programming should rely on the Array Integrity feature and the built in Error Correction to determine the state of the flash.

e7587PS :

FLASH: Erase Suspend Latency is out of spec in Flash except Code Flash 0.

Description

The Erase Suspend Latency is out of spec on all available Flash memories except Code Flash 0. A maximum latency of 34us can occur instead of the specified 30us.

On the contrary the Erase Suspend Latency is correctly in spec in Code Flash 0.

Workarounds

The wait for the suspend acknowledgement must be done by polling the DONE bit of MCR and not through a wait for a fix delay.

e6384PS :

Flash: Erroneous update of the ADR register in case of multiple ECC errors

Description

An erroneous update of the Address register (ADR) occurs whenever there is a sequence of 3 or more events affecting the ADR (ECC single or double bit errors or RWW error) and both the following conditions apply:

- The priorities are ordered in such a way that only the first event should update ADR.

- The last event although it does not update ADR sets the Read While Write Event Error (RWE) or the ECC Data Correction (EDC) in the Module Configuration Register (MCR).

For this case the ADR is wrongly updated with the address related to one of the intervening events.

Example - If a sequence of two double-bit ECC errors is followed by a single-bit correction without clearing the ECC Event Error flag (EER) in the MCR, then the value found in ADR after the single-bit correction event is the one related to the second double-bit error (instead of the first one, as specified)

Workarounds

Always process Flash ECC errors as soon as they are detected.

Clear MCR[RWE] at the end of each flash operation (Program, Erase, Array Integrity Check, etc...).

e7067PS :

FCU: Timeout feature doesn't work correctly.

Description

A fault occurs, timeout for this fault is enabled and the fault is not recovered automatically before the timeout :- In this case device will go to ALARM state and waits for timeout, after timeout has elapsed it enters to FAULT state. However, if another fault occurs with timeout enabled the FCU will go directly to FAULT state without waiting for timeout to be elapsed.
If it is desired that FCU will go again to ALARM state for the second fault, a Watchdog reset needs to be asserted after first fault is detected.

Workarounds

None

e6583PS :

Diode path between VDD_LV and VDD_HV

Description

The design implements diodes between VDD_LV and VDD_HV. This leads to current flowing from VDD_LV to VDD_HV if power-up and power-down sequences are not handled correctly.
VDD_LV to VDD_HV current may stress the device and should be avoided when possible.

Workarounds

Insert a schottky diode between VDD_LV and VDD_HV.

e4781PS :

LINFlex: Unexpected LIN timeout in slave mode

Description

If the LINFlex is configured in LIN slave mode, an unexpected LIN timeout event (LINESR[OCF]) may

occur during LIN Break reception.

Workarounds

It is recommended to disable this functionality during LINFlex initialization by clearing LINTCSR[IOT] and LINIER[OCIE] bits, and ignore timeout events.

e4897PS :

LINFlex: BDRL/BDRM cannot be accessed as byte or half-word

Description

LINFlex data buffers (BDRL/BDRM) cannot be accessed as byte or halfword. Accessing BDRL/BDRM in byte/half word mode will lead to incorrect data writing/reading.

Workarounds

Access BDRL/BDRM registers as word only.

e1685PS :

FMPLL: FMPLL_CR[UNLOCK_ONCE] wrongly set

Description

If the FMPLL is locked and a functional reset occurs, FMPLL_CR[UNLOCK_ONCE] is automatically set even when the FMPLL has not lost lock.

Workarounds

Do not use the FMPLL_CR[UNLOCK_ONCE] when a functional reset occurs.

e3630PS :

FMPLL: Do not poll flag FMPLL_CR[PLL_FAIL]

Description

For the case when the FMPLL is indicating loss of lock the flag FMPLL_CR[PLL_FAIL] is unpredictable.

Workarounds

To avoid reading an incorrect value of FMPLL_CR[PLL_FAIL] only read this flag inside the FMPLL Interrupt Service Routine (ISR). The ISR indicates the flag has been set correctly and at this point the flag can be cleared. Do not poll flag FMPLL_CR[PLL_FAIL] at any other point in the application software.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+44 8 52200080 (English)
+44 80 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

MPC5604P0M36W
Rev. 2 09/21/10