MPC5607B MCU Family Device Errata

Introduction

This device errata applies to the following products:

- MPC5607B

e3738PS :

RGM_FES[F_FMPLL] bit is set in case of LVD2.7 event.

Description

When the PLL is used and a low voltage event is detected on LVD2.7, the RGM_FES[F_FMPLL] bit may be set.

Workarounds

Following the LVD assertion and reset recovery phase, the application should ensure that the PLL_FAIL state bit is cleared during the device re-initialisation phase.

e3796PS :

Register protection not implemented for PCU_PCONF3, ME_PCTL8..11, ME_PCTL12..15, ME_PCTL52..55, IFMC16..23 registers

Description

Register protection is not implemented for registers PCU_PCONF3, ME_PCTL8..11, ME_PCTL12..15,ME_PCTL52..55, IFMC16..23. It is not possible to block write transactions to these registers.

Workarounds

No workaround

e4210PS :

Ports PH[11..15], PJ[4] supports only slow strength capability.

Description

SRC bit in PCR[123..127] and PCR[148] registers is not programmable. Therefore, PH[11..15] and PJ[4] ports support only slow strength capability and cannot be configured to have medium strength capability.

Workarounds

No workaround

e5194PS :

Interrupt vector of wakeup lines mapped on GPIO[103], GPIO[105], GPIO[89] and GPIO[131] is not correct.

Description

Wakeup lines mapped on GPIO[103], GPIO[105], GPIO[89] and GPIO[131] are associated to WakeUp_IRQ_3 interrupt vector instead of WakeUp_IRQ_2.

Workarounds

No workaround

e5879PS :

Serial Boot and Censorship - RCHW 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 can be 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 cannot be accessed.

Workarounds

No Workaround

e5880PS :

Serial boot in uncensored device - SRAM access allowed

Description

In uncensored devices it is possible to download code via LINFlex or FlexCAN (Serial Boot Mode) into internal SRAM even if the 64-bit private password stored in the flash and provided during the boot sequence is an illegal password.

Workarounds

No workaround

e6121PS :

FLASH: RWW-Error during stall-while-write

Description

If Stall/Abort-While-Write is enabled and an erase operation is started on one sector while fetching code from another then the following sequence is executed:
- CPU is stalled when flash is unavailable
- PEG flag set (stall case) or reset (abort case)
- Interrupt triggered if enabled

However, in all cases also the RWE flag is set, indicating a RWW error.

Workarounds

If Stall/Abort-While-Write is used then application software should ignore the setting of the RWE flag. The RWE flag should be cleared after each HV operation.
If Stall/Abort-While-Write is not used the application software should handle RWE error.

e6192PS :

TDO pin floating in STANDBY mode

Description

The TDO pad has been moved into the STANDBY domain in order to allow low-power debug handshaking in STANDBY mode. However, no pull-resistor is active on the TDO pad while in STANDBY mode. At this time the pad is configured as an input. When no debugger is connected the TDO pad is floating causing additional current consumption.

Workarounds

To avoid the extra consumption TDO must be connected. An external pull-up resistor in the range of 47-100 kOhms should be added between the TDO pin and VDD. Only in case the TDO pin is used as application pin and a pull-up cannot be used then a pull-down resistor with the same value should be used between TDO pin and GND instead.

e6193PS :

Device will not enter STOP mode if configured for MVREG off when working at temperature higher than 140C.

Description

At ambient temperature higher than 140C, device will not enter STOP mode when configured for MVREG off. Due to higher leakage in the device, the device consumption before entering STOP is greater than a predefined threshold, due to which MVREG stays on. Therefore, STOP mode transition is not complete and the device total consumption is 5mA more than expected otherwise.

Workarounds

Always program MVRON bit of ME_STOP_MC register to '1'.

e6668PS :

Flexcan modules gets reset when PD2 is switched off.

Description

Flexcan modules are incorrectly associated to PD2 instead of PD1. Therefore, whenever PD2 (i.e. 24K SRAM after first 8K) is switched off using PCU_PCONF2 register, flexcan modules will be reset.

Workarounds

STOP0, HALT0, RUN3, RUN2, RUN1, RUN0, DRUN, SAFE, TEST bits of PCONF2 register must always be kept at '1'. If 24K SRAM (i.e PD2) need to be retained in STANDBY0 mode, PCU_PCONF2.STANDBY0 should be programmed to '1'.

e8329PS :

PIT events cannot be used to trigger ADC conversion incase BCTU runs on divided system clock

Description

If BCTU operates on divided system clock (i.e MC_CGM.CGM_SC_DC2.DIV0 NOT EQUAL 0x0), events from PIT timer cannot be used to trigger ADC conversion. BCTU passes the trigger event, but ADC channel information gets corrupted.

Workarounds

Always write MC_CGM.CGM_SC_DC2.DIV0 to 0x0 to run BCTU at system frequency.

e8498PS :

Device may hang due to functional reset while using debug hand-shaking mechanism

Description

When debug session is on-going and low power hand-shaking mode enabled, a functional reset may hang-up the device, waiting for debugger acknowledge.

Workarounds

NPC_PCR[LP_DBG_EN] bit must be cleared to ensure correct reset sequence

e8499PS :

NMI pin configuration limitation in standby mode

Description

NMI pin cannot be configured to generate Non Maskable Interrupt event to the core (WKPU_NCR[NDSS] = "00") if the following standby mode is to be used:
- NMI pin enabled for wake-up event,
- standby exit sequence boot from RAM,
- code flash module power-down on standby exit sequence.

With following configuration following scenario may happen:

1. System is in standby
2. NMI event is triggered on PA[2]
3. System wakeup z0 core power domain.
4. z0 core reset is released and NMI event is sampled by core on first clock-edge.
5. z0 core attempt to fetch code at 0x10 address (IVPR is not yet initialized by application) and receive an exception since flash is not available
6. z0 core enter machine check and execution is stalled.

Workarounds

If NMI is configured as wake-up source, WKPU_NCR[NDSS] must be configures as "11". This will ensure no NMI event is trigger on the core but ensure system wakeup is triggered.
After standby exit, core will boot and configure its IVOR/IVPR, it may then re-configure WKPU_NCR:DSS to the appropriate configuration for enabling NMI/CI/MCP.

e8500PS :

PA[1] pull-up enabled when NMI activated

Description

When NMI is enabled (either WKPU_NCR[NREE] or WKPU_NCR[NFEE] set to '1'), PA[1] pull-up is automatically activated independently from SIUL_PCR1[WPS] and SIUL_PCR1[WPE].
This has no effect during STANDBY mode. PA[1] pull-up is then correctly configured through WKPU_WIPUER[IPUE[2]].

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

No Workaround

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 ABS2 = 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 is the traffic the higher is 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 Continious 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 insure PCS stability during data transmission in Continious Selection Format and Continious 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.

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 which 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'.

e3953PS :

MC_ME: In RUNx mode after STOP0 exit system RAM can be accessed before it is ready

Description

For the case when:
ME_STOP0_MC[FXOSC] is enabled
ME_STOP0_MC[FIRC] is disabled
ME_RUNx_MC[FIRC] is enabled
ME_RUNx_MC[SYSCLK] = FXOSC or FXOSC_DIV
At the transition of STOP0 to RUNx, the RUNx mode can be entered before the system RAM is ready. If the application software accesses the RAM during this time the RAM value can not be guaranteed.

Workarounds

There are two workarounds possible:

1. Do not disable the IRC if the system clock source is not disabled. The XOSC draws a lot more current than the IRC, so there should be no noticeable increase in the STOP0 mode power consumption.

2. Have the software check that the mode transition has completed via the ME_GS register before accessing the system RAM.

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 which 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 which 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.

e4565PS :

MC_RGM: MCU will enter BAM static state if reset during STANDBY0 mode

Description

If the device is in STANDBY0 mode, and it is configured to re-boot from RAM (RGM_STDBY[BOOT_FROM_BKP_RAM] = 1) while keeping the code flash in power-down or low-power mode (ME_DRUN_MC[CFLAON] != "11"), an external reset will cause the device to enter the BAM static state where it remains inoperable until another reset occurs.

Workarounds

Generate the external reset by asserting the RESETB pin twice. The two pulses must be separated by at least 200 us.

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

No workaround.

e5955PS :

MC_RGM: External reset not asserted if STANDBY0 is exited due to external reset event

Description

If the device is in STANDBY0 mode and an external reset occurs, the MC_RGM may not assert the external reset for the duration of the reset sequence even when RGM_FBRE[BE_EXR = 0]. This incorrect behavior occurs only if the system releases the external reset before the end of reset sequence PHASE1.

Workarounds

Ensure that the system asserts the external reset for at least the maximum duration of reset sequence PHASE1.

e6440PS :

SWT: SWT interrupt does not cause STOP0 mode exit

Description

While in STOP0 mode, if the SWT is configured to generate an interrupt and the system clock is disabled (ME_STOP0_MC[SYSCLK] = 0xF), a SWT interrupt event will not trigger an exit from STOP0 mode.

Other internal or external wakeup events (RTC, API, WKUP pins) are not affected and will trigger a STOP0 exit independent of the ME_STOP0_MC[SYSCLK] configuration.

Workarounds

If a SWT interrupt is to be used to wake the device during STOP0 mode, software may not disable the system clock (ME_STOP0_MC[SYSCLK] != 0xF).

e7776PS :

MC_ME: Main VREG not disabled during STOP0 mode

Description

If STOP0 mode is configured with ME_STOP0_MC.MVRON = '0' and ME_STOP0_MC.RCON = '0', the main VREG will nevertheless remain turned on during STOP0 mode, thus resulting in higher power consumption than expected.

Workarounds

Configure STOP0 mode so that ME_STOP0_MC.RCON = '1'.

e6394PS :

NPC: TCK frequency must be lower than system frequency in low power modes for communication

Description

The JTAG Clock (TCK) typically operates at a frequency well below the system clock frequency, as specified in the electrical data sheets. In some cases, however, such as low power mode (if the device supports low power modes), the system clock frequency may be lowered significantly from the normal operating range. If the system clock frequency is reduced below the frequency of TCK it will no longer be possible to communicate with the Nexus Port Controller Port Configuration Register (NPC_PCR). Therefore, if the tool needs to update the NPC_PCR Low Power Debug Enable (NPC[PCR[LP_DBG]) or Low Power Synchronization bits (NPC[PCR[LP1_SYN] and NPC[PCR[LP2_SYN]), the clock frequency must be lowered.

Workarounds

Insure that the frequency of TCK does not exceed the system clock frequency during normal operation and during low power operation.

e7216IPG : NPC: MCKO_DIV can be set to 0x0 (1X MCKO)

Description

The Nexus Port Controller Port Configuration Register MCKO Divider bits (NPC_PCR[MCKO_DIV]) can be set to 0b000 to select a 1X clock rate as the Nexus Auxiliary output port frequency for the MCKO and MDO pins. Note: Depending on the system frequency, this may force the MCKO and MDO pins to switch at a frequency higher than can be supported by the pins. This maximum frequency is specified in the device electrical specification of the Nexus MCKO and MDO pins.

Workarounds

Insure that the maximum operating frequency of the MDO and MCKO pins is not violated when setting the NPC_PCR[MCKO_DIV] values.

Note: tools may not support 1X mode. Check with your tool vendor.

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).

e4283PS :

ADC1 : Do not use the configuration "00" for INPCMP bits of ADC1.CTRx registers

Description

INPCMP bits of ADC1.CTRx registers cannot be programmed to "00". Due to this, ADC performance is not guaranteed above 5MHz at 3.3V and 8MHz at 5V range. The performace is same as INPCMP bits programmed to "01". If minimum conversion time is desired then other configurations can be used.

Workarounds

No workaroud.

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

No workaround.

e7199PS :

ADC: At ADC_DSDR register only increments of 2 will increase delay when ADC clock = Peripheral Clock/2

Description

For the case when ADC clock = Peripheral Clock/2 the ADC Multiplexer Decode Signal Delay Register (ADC_DSDR) has to be incremented by 2 to see an additional ADC clock cycle delay on the decode signal.

e.g
ADC_DSDR = 0 - 0 ADC clock cycle delay
ADC_DSDR = 2 - 1 ADC clock cycle delay
ADC_DSDR = 4 - 2 ADC clock cycles delay
etc.

Workarounds

No workaround

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, it is recommended to disable SAFE mode transitions when the device is executing a mode transition with the FXOSC as the system clock source in the target mode.

e5883PS :

ADC: CTUEN bit in ADC.MCR register cannot be reset if a BCTU channel is enabled

Description

While any BCTU channels is enabled (CTU.EVTCFGRx.TM =1), the CTU will continously send trigger requests to ADC. If CTUEN bit in MCR is reset while BCTU channels are enabled, the ADC DTU trigger state may become undefined and ADC module may not service trigger request from CTU anymore.

Workarounds

Ensure ADC.MCR.CTUEN is set before enabling any CTU channels (CTU.EVTCFGRx.TM =1). Ensure all CTU channels are disabled (CTU.EVTCFGRx.TM =0) before ADC.MCR.CTUEN is cleared.

e6609PS :

ADC may not acknowledge trigger request from CTU while entering peripheral clock gated mode

Description

When ADC configured in CTU mode, it sends a next command request after completion of every conversion. During mode transition, if both ADC and CTU modules are entering clock gated mode through MC_ME registers then it might happen that ADC may miss the trigger request from CTU. After peripheral stop request, it takes 1 cycle to gate ADC clock while CTU requires 3 cycles. Due to this, CTU may generate the trigger based on the pending requests and ADC does not have clock running to acknowledge it. As a result, when both ADC and CTU come out of clock gated mode, ADC expects a trigger from CTU as it has already sent next commmand request and CTU waits for next command request from ADC which can come only after completing a conversion.

Workarounds

Program TRIGGER_EN bit in CTU_EVTCFGR registers to '0' before gating the clock to ADC and CTU. After exiting clock gated mode, program the required TRIGGER_EN bit in CTU_EVTCFGR registers to '1'. All the pending requests from EMIOS/PIT will still be pending and will be processed after TRIGGER_EN bits are set back to '1'.

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 Lock 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 desired 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 desired 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 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...).

e6841PS :

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

No workaround. It is recommended to disable this functionality during LINFlex initialization by clearing LINTCSR[IOT] and LINIER[OCIE] bits, and to 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

Information in this document is provided solely to enable system and sofware implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor prodcuts are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to http://www.freescale.com/epp.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.