Mask Set Errata for Mask 0N61C_1M09S

This report applies to mask 0N61C_1M09S for these products:

- MPC5668X

Table 1. Errata and Information Summary

Erratum ID	Erratum Title
ERR004168	ADC: Abort switch aborts the ongoing injected channel as well as the upcoming normal channel
ERR050438	ADC: Divided clock frequency limitation
ERR004186	ADC: Do not trigger ABORT or ABORTCHAIN prior to the start of CTU triggered ADC conversions and do not trigger ABORTCHAIN prior to the start of INJECTED triggered ADC conversions.
ERR005569	ADC: The channel sequence order will be corrupted when a new normal conversion chain is started prior to completion of a pending normal conversion chain
ERR008775	CRP: Incorrect device configuration values loaded during external resets
ERR001282	CRP: Spurious Pin Wakeup in Run Mode
ERR006026	DSPI: Incorrect SPI Frame Generated in Combined Serial Interface Configuration
ERR007352	DSPI: reserved bits in slave CTAR are writable
ERR050782	e200: Time Base TBU register contains wrong value during TBL rollover
ERR003421	e200z: Cache returns value, even when disabled
ERR006966	eDMA: Possible misbehavior of a preempted channel when using continuous link mode
ERR050575	eMIOS: In Center Aligned Output PWM Buffered (OPWMCB) Mode , channel prescaler value should be less than associated timebase prescaler value
ERR009344	eSCI: Late assertion of Transmit Data Ready Interrupt Flag (TXRDY) for Local Interconnect Network (LIN) frame receive (RX) operation
ERR009001	eSCI: Incorrect behavior while in LIN Standard Bit error detection mode
ERR009361	eSCI: Timing of TXRDY interrupt flag assertion is incorrect for LIN TX Frame
ERR002340	FEC: slot time is designed for 516 bit times; deviation from the 802.3
ERR010546	Flash: Flash Memory Data Disturb due to Sleep Cycling and Power Cycling
ERR003853	FLASH: Flash array access may be incorrect after sleep mode exit or after power up
ERR002382	FLASH: Flash Array Integrity Check
ERR003659	FLASH: Resuming after a suspend during an Erase may prevent the erase from completing.

Table continues on the next page...

Table 1. Errata and Information Summary (continued)

Erratum ID	Erratum Title
ERR007322	FlexCAN: Bus Off Interrupt bit is erroneously asserted when soft reset is performed while FlexCAN is in Bus Off state
ERR008951	I2C: Attempting a start cycle while the bus is busy may generate a short clock pulse
ERR002501	LPM: High VDD Sleep Mode Current
ERR006726	NPC: MCKO clock may be gated one clock period early when MCKO frequency is programmed as SYS_CLK/8.and gating is enabled
ERR004146	SARADC: Interrupted conversions are aborted, but may not be properly restored

Table 2. Revision History

Revision	Changes
04 JUN 2012	Initial revision
17 APR 2015	Errata added: e5569, e8775, e6026, e7352, e7322, e6726, e3421, e6966 Errata removed: None Errata changed: e4186, e4146 No changes to errata with this revision
17 OCT 2016	The following errata were added. • ERR008951 • ERR009001 • ERR009344 • ERR009361 • ERR010546 The following erratum was revised. • ERR004146
31 JAN 2020	The following erratum was added. • ERR050438
30 JULY 2021	The following errata were added. • ERR050575 • ERR050782 The following errata were revised. • ERR003853 • ERR050438

ERR004168: ADC: Abort switch aborts the ongoing injected channel as well as the upcoming normal channel

Description: If an Injected chain (jch1,jch2,jch3) is injected over a Normal chain (nch1,nch2,nch3,nch4) the Abort switch does not behave as expected.

Expected behavior:

Correct Case (without SW Abort on jch3): Nch1- Nch2(aborted) -Jch1 – Jch2 – Jch3 – Nch2(restored) - Nch3 – Nch4 Correct Case(with SW Abort on jch3): Nch1 – Nch2(aborted) - Jch1 – Jch2 – Jch3(aborted) - Nch2(restored) - Nch3 – Nch4

Observed unexpected behavior:

Fault1 (without SW abort on jch3): Nch1 – Nch2(aborted) - Jch1 – Jch2 - Jch3 – Nch3 – Nch4 (Nch2 not restored) Fault2 (with SW abort on jch3): Nch1- Nch2 (aborted) - Jch1 – Jch2 – Jch3(aborted) - Nch4(Nch2 not restored &Nch3 conversion skipped)

Workaround: It is possible to detect the unexpected behavior by using the CEOCFRx register. The CEOCFRx fields will not be set for a not restored or skipped channel, which indicates this issue has occurred. The CEOCFRx fields need to be checked before the next Normal chain execution (in scan mode). The CEOCFRx fields should be read by every ECH interrupt at the end of every chain execution.

ERR050438: ADC: Divided clock frequency limitation

Description: The Data Sheet specifies the Maximum ADC Clock (AD_clk) Frequency (FMAX) as 60 MHz. The ADC MCR[ADCLKSEL] analog clock frequency selector controls whether AD_clk is the same or half the frequency as the incoming ADC reference (ipg_clk) clock. If MCR[ADCLKSEL] = 0b0 to select the divided clock, then the Maximum ADC Clock (AD_clk) frequency is limited to 35 MHz.

Exceeding 35 MHz when MCR[ADCLKSEL] = 0b0 does not guarantee full ADC functionality across the full range of VDD core supply voltage, VDDA analog supply voltage, temperature, and part to part variation. Worse case conditions are lower VDD, lower VDDA, and higher temperature. Potential issues may include inability to complete a single or chain of conversions, unexpected conversions of channel 0, or repeated conversions of a selected channel resulting in the setting of the corresponding overwrite xDATAREGn[OVERW] bit (if MCR[OWREN] was previously set). Conversion results may also be affected and may drift towards a VRH/2 value.

Workaround: Full frequency 60 MHz ADC operation is possible when MCR[ADCLKSEL] = 0b1, which does not select the divided clock. This configuration requires that both SIU_SYSCLK[SYSCLKDIV] = 0b00 and SIU_SYSCLK[LPCLKDIV2] = 0b00 to provide an undivided 50% duty cycle clock to the ADC which also implies that the main System clock is limited to a maximum of 60 MHz.

If MCR[ADCLKSEL] = 0b0 to select the divided clock, then limit the Maximum ADC Clock (AD_clk) frequency to 35 MHz. This configuration supports a maximum System clock frequency of 70 MHz when SIU_SYSCLK[LPCLKDIV2] = 0b00. When SIU_SYSCLK[LPCLKDIV2] is set greater than 0b00, then the maximum System clock frequency of 116MHz is supported with a lower ADC Clock frequency.

Correct operation and conversion results may be confirmed by checking after each conversion for the expected results for each of the appropriate status flags such as ISR[ECH], ISR[EOC], CEOCFRn[EOCx], xDATAREGn[VALID], and xDATAREGn[OVERW].

ERR004186: ADC: Do not trigger ABORT or ABORTCHAIN prior to the start of CTU triggered ADC conversions and do not trigger ABORTCHAIN prior to the start of INJECTED triggered ADC conversions.

Description: When ADC_MCR[ABORT] or ADC_MCR[ABORTCHAIN] is set prior to the ADC receiving a CTU trigger, the next CTU triggered ADC conversion will not be performed and further CTU triggered ADC conversions will be blocked.

When ADC_MCR[ABORTCHAIN] is set prior to the ADC receiving an INJECTED trigger, the next INJECTED ADC conversion will not be performed. Following the ABORTCHAIN command the MCU behaviour does not meet the specification as ADC_ISR[JECH] is not set and ADC_MCR[ABORTCHAIN] is not cleared.

Workaround: Do not program ADC_MCR[ABORT] or ADC_MCR[ABORTCHAIN] before the start of ADC conversions.

The case when CTU triggered ADC conversions are blocked should be avoided however it is possible to reactivate CTU conversions by clearing and setting ADC_MCR[CTUEN].

ERR005569: ADC: The channel sequence order will be corrupted when a new normal conversion chain is started prior to completion of a pending normal conversion chain

Description: If One shot mode is configured in the Main Configuration Register (MCR[MODE] = 0) the chained channels are automatically enabled in the Normal Conversion Mask Register 0 (NCMR0). If the programmer initiates a new chain normal conversion, by setting MCR[NSTART] = 0x1, before the previous chain conversion finishes, the new chained normal conversion will not follow the requested sequence of converted channels.

For example, if a chained normal conversion sequence includes three channels in following sequence: channel0, channel1 and channel2, the conversion sequence is started by MCR[NSTART] = 0x1. The software re-starts the next conversion sequence when MCR[NSTART] is set to 0x1 just before the current conversion sequence finishes.

The conversion sequence should be: channel0, channel1, channel2, channel0, channel1, channel2.

However, the conversion sequence observed will be: channel0, channel1, channel2, channel1, channel1, channel2. Channel0 is replaced by channel1 in the second chain conversion and channel1 is converted twice.

Workaround: Ensure a new conversion sequence is not started when a current conversion is ongoing. This can be ensured by issuing the new conversion setting MCR[NSTART] only when MSR[NSTART] = 0.

Note: MSR[NSTART] indicates the present status of conversion. MSR[NSTART] = 1 means that a conversion is ongoing and MSR[NSTART] = 0 means that the previous conversion is finished.

ERR008775: CRP: Incorrect device configuration values loaded during external resets

Description: Device configuration information is retrieved from FLASH by the Clocks, Reset, and Power (CRP) module during all reset sequences. This information is written to set the default trim values for the Internal Resistor/Capacitor (IRC) oscillators. These are written to the Clock Source Register (CRP_CLKSRC) for the 16 MHz IRC [TRIM16IRC] and the 128 kHz IRC ([TRIM128IRC] oscillators. The device configuration information is also used to fine tune the accuracy of the internal Low Voltage Detect circuits (LVDs) and internal voltage regulators, along with the setting of SRAM timing. During an external reset, triggered by the assertion of the RESET_B pin, a clock synchronization issue between RESET_B and the internal system clock may cause incorrect device configuration information to be retrieved from FLASH. Only external resets triggered by the assertion of RESET_B are affected by this issue. Resets triggered by Power On Resets (POR), LVD, or internal sources (Software Watchdog Timer [SWT], Loss of Clock [LOC], Loss of Lock [LOL], debug [JTAG], Checkstop [CS], software [SW]) are not affected by this issue and will retrieve correct device configuration information. When incorrect device configuration information is retrieved, the CRP_CLKSRC [TRIM16IRC] and CRP_CLKSRC[TRIM128IRC] registers will be loaded with the values of 0x0E and 0x0B, respectively.

Workaround: If possible, do not generate external resets with the assertion of the RESET_B pin. All other types of reset, for example, POR, LVDs, SW, are not affected by this erratum.

If external resets are used, software should execute one of the following options prior to accessing any SRAM:

1) Use software to check the Reset Status Register (SIU_RSR) to determine whether the reset source was an external reset. If the source was an external reset, execute a software reset, let the Software Watchdog Timer (SWT) time out and generate a reset, or force any other type of reset. Execution of any non-external reset will correct the device configuration information that is retrieved from FLASH.

2) Use software to check the values of TRIM16IRC and/or TRIM128IRC in the CRP_CLKSRC register. If TRIM16IRC=0x0E or TRIM128IRC=0x0B, then another reset should be performed.

If external resets are used without software verification of the reset type or trim values, then protection from this issue will have to rely on system safeguards, such as the SRAM Error Correction Code (ECC), SWT resets for any code execution issues, internal/external LVD resets for any voltage issues. These safeguards will reset the device and will correct the device configuration information that is retrieved from FLASH.

ERR001282: CRP: Spurious Pin Wakeup in Run Mode

Description: During RUN mode, a CRP_PSCR[PWKSRCF] flag may be unintentionally set instead of the intention of only setting the wakeup flags during Sleep mode. The pin wakeup flags work fine during Sleep mode. There is not an issue when the pin wakeup flags are disabled. The flag may be set during flag configuration of CRP_WKSE[WKCLKSEL], CRP_WKPINSEL[WKPSELn], and CRP_WKSE[WKPDETn]. The flag may also be set if there is a change in the wakeup pin at the same time as the internal system clock or the internal pin wakeup clock. Note that if the pin toggles then the flag may be set regardless of whether the pin is selected as posedge or negedge.

Workaround: Workaround for unintentional flag setting during configuration, would be to first complete the pin wakeup configuration (i.e. select pin wakeup clock, select pin muxing, and configure pin enable/edge detect via the CRP registers CRP_WKSE[WKCLKSEL],

CRP_WKPINSEL[WKPSELn], and CRP_WKSE[WKPDETn]). Then poll the CRP_PSCR[PWKSCRF] pin wakeup flag and if set write one to clear and then recheck. May take one pin wakeup clock cycle to clear.

Workaround for unintentional flag setting during pin change is to have the SLEEPF and STOPF flags in CRP_PSCR checked in the CRP interrupt service routine and if neither of them is set then clear all flags in CRP_PSCR[PWKSRCF].

ERR006026: DSPI: Incorrect SPI Frame Generated in Combined Serial Interface Configuration

Description: In the Combined Serial Interface (CSI) configuration of the Deserial Serial Peripheral Interface (DSPI) where data frames are periodically being sent (Deserial Serial Interface, DSI), a Serial Peripheral Interface (SPI) frame may be transmitted with incorrect framing.

The incorrect frame may occur in this configuration if the user application writes SPI data to the DSPI Push TX FIFO Register (DSPI_PUSHR) during the last two peripheral clock cycles of the Delay-after-Transfer (DT) phase. In this case, the SPI frame is corrupted.

Workaround: Workaround 1: Perform SPI FIFO writes after halting the DSPI.

To prevent writing to the FIFO during the last two clock cycles of DT, perform the following steps every time a SPI frame is required to be transmitted:

Step 1: Halt the DSPI by setting the HALT control bit in the Module Configuration Register (DSPI_MCR[HALT]).

Step 2: Poll the Status Register's Transmit and Receive Status bit (DSPI_SR[TXRXS]) to ensure the DSPI has entered the HALT state and completed any in-progress transmission. Alternatively, if continuous polling is undesirable in the application, wait for a fixed time interval such as 35 baud clocks to ensure completion of any in-progress transmission and then check once for DSPI_SR[TXRXS].

Step 3: Perform the write to DSPI_PUSHR for the SPI frame.

Step 4: Clear bit DSPI_MCR[HALT] to bring the DSPI out of the HALT state and return to normal operation.

Workaround 2: Do not use the CSI configuration. Use the DSPI in either DSI-only mode or SPI-only mode.

Workaround 3: Use the DSPI's Transfer Complete Flag (TCF) interrupt to reduce worst-case wait time of Workaround 1.

Step 1: When a SPI frame is required to be sent, halt the DSPI as in Step 1 of Workaround 1 above.

Step 2: Enable the TCF interrupt by setting the DSPI DMA/Interrupt Request Select and Enable Register's Transmission Complete Request Enable bit (DSPI_RSER[TCF_RE])

Step 3: In the TCF interrupt service routine, clear the interrupt status (DSPI_SR[TCF]) and the interrupt request enable (DSPI_RSER[TCF_RE]). Confirm that DSPI is halted by checking DSPI_SR[TXRXS] and then write data to DSPI_PUSHR for the SPI frame. Finally, clear bit DSPI_MCR[HALT] to bring the DSPI out of the HALT state and return to normal operation.

ERR007352: DSPI: reserved bits in slave CTAR are writable

Description: When the Deserial/Serial Peripheral Interface (DSPI) module is operating in slave mode (the Master [MSTR] bit of the DSPI Module Configuration Register [DSPIx_MCR] is cleared), bits 10 to 31 (31 = least significant bit) of the Clock and Transfer Attributes Registers (DSPIx_CTARx) should be read only (and always read 0). However, these bits are writable, but setting any of these bits to a 1 does not change the operation of the module.

Workaround: There are two possible workarounds.

Workaround 1: Always write zeros to the reserved bits of the DSPIx_CTARn_SLAVE (when operating in slave mode).

Workaround 2: Mask the reserved bits of DSPIx_CTARn_SLAVE when reading the register in slave mode.

ERR050782: e200: Time Base TBU register contains wrong value during TBL rollover

Description: The e200 Time Base (TB) facility is a 64-bit structure provided for maintaining the time of day and operating interval timers. The TB consists of two 32-bit registers – time base upper (TBU) and time base lower (TBL). TBU and TBL are concatenated to provide a long-period 64-bit counter. TBL increments until its value becomes 0xFFFF_FFFF. The intended behavior is that at the next increment when the TBL value becomes 0x0000_0000 that the TBU value is incremented. But the actual behavior is that after the TBL value becomes 0x0000_0000, the TBU value will not increment until the transition of the TBL value to 0x0000_0001.

Workaround: Software will need to take care about the wrong TBU value during TBL rollover. Use the following sequence for reading TBU and TBL values:

```
loop:
mfspr r12, TBL
mfspr r3, TBU
mfspr r4, TBL
cmpl r4,r12
se_blt loop
```

ERR003421: e200z: Cache returns value, even when disabled

Description: The cache line fill buffer of the e200z core will still provide a data cache hit, even if the Way Data Disable (WDD), Additional Ways Data Disable (AWDD), and the Way Access Mode (WAM, if available) bits are set.

Workaround: To avoid cache hit to the Line Fill Buffer after the WDD, AWDD, and WAM bits are disabled, a Cache Miss must be forced to clear the fill buffer. This can be done by branching to an instruction address that has not been fetched before.

ERR006966: eDMA: Possible misbehavior of a preempted channel when using continuous link mode

Description: When using Direct Memory Access (DMA) continuous link mode Control Register Continuous Link Mode (DMA_CR[CLM]) = 1) with a high priority channel linking to itself, if the high priority channel preempts a lower priority channel on the cycle before its last read/write sequence, the counters for the preempted channel (the lower priority channel) are corrupted. When the preempted channel is restored, it continues to transfer data past its “done” point (that is the byte transfer counter wraps past zero and it transfers more data than indicated by the byte transfer count (NBYTES)) instead of performing a single read/write sequence and retiring.

The preempting channel (the higher priority channel) will execute as expected.

Workaround: Disable continuous link mode (DMA_CR[CLM]=0) if a high priority channel is using minor loop channel linking to itself and preemption is enabled. The second activation of the preempting channel will experience the normal startup latency (one read/write sequence + startup) instead of the shortened latency (startup only) provided by continuous link mode.

ERR050575: eMIOS: In Center Aligned Output PWM Buffered (OPWMCB) Mode , channel prescaler value should be less than associated timebase prescaler value

Description: In Center Aligned Output PWM Buffered (OPWMCB) mode , if channel prescaler value defined by eMIOS Cn[UCPRE] and eMIOS C2_n[UCEXTPRE] fields, is greater than or equal to timebase channel prescaler values than OPWMCB mode channel output may not produce pulses as expected.

This is not applicable If both OPWMCB channel and timebase channel prescaler are 0. This is also not applicable if in OPWMCB mode lead or trail dead time insertion feature is not used.

Workaround: The channel working in OPWMCB mode should have a prescaler value which is less than the prescaler value of the channel used as associated timebase.

ERR009344: eSCI: Late assertion of Transmit Data Ready Interrupt Flag (TXRDY) for Local Interconnect Network (LIN) frame receive (RX) operation

Description: Assertion of the Transmit Data Ready Interrupt Flag (TXRDY) in the Interrupt Flag and Status Register 2 (eSCI_IFSR2) indicates that data written to the LIN Transmit Register (eSCI_LTR) has been processed by the eSCI module. For the first three data writes to the eSCI_LTR during LIN frame generation, the TXRDY flag is asserted one clock cycle after the write access. During LIN RX operation, assertion of the TXRDY flag that coincides with the fourth data write to the eSCI_LTR is delayed. The TXRDY flag is not asserted until the LIN RX frame has been completely received from the slave device. The TXRDY flag is asserted when the Frame Complete Interrupt (FRC) flag of the eSCI_IFSR2 register is asserted.

Workaround: Application software should expect a delay in the assertion of the TXRDY flag after the fourth data write to the eSCI_LTR. Instead of expecting TXRDY assertion within one clock cycle of the fourth data write to the eSCI_LTR, application software should expect assertion of the TXRDY flag after the LIN RX frame has been completely received from the slave device.

ERR009001: eSCI: Incorrect behavior while in LIN Standard Bit error detection mode

Description: After a Local Interconnect Network (LIN) wake-up signal frame is transmitted from a master device while in Standard Bit error detection mode (eSCI_CR2[FBR] = 0), a bit error is detected in any subsequent LIN Transmit (TX) or Receive (RX) frames sent from the master device. After the bit error is detected, the Bit Error Interrupt Flag (eSCI_IFSR1[BERR]) is asserted, and the LIN controller will not generate TX or RX frames.

Workaround: Workaround 1: Reset the LIN Protocol Engine of the eSCI controller by writing 1 and then a 0 to the LIN Protocol Engine Stop and Reset bit in LIN Control Register 1 (eSCI_LCR1[LRES]) after a complete wake-up frame is sent.

Workaround 2: Use the LIN module in Fast Bit error detection mode, and do not use the Standard Bit error detection mode. Fast Bit Error detection mode can be enabled by writing 1 to the Fast Bit Error Detection bit in Control Register 2 (eSCI_CR2[FBR] =1).

ERR009361: eSCI: Timing of TXRDY interrupt flag assertion is incorrect for LIN TX Frame

Description: When generating a Local Interconnect Network (LIN) Transmit (TX) Frame, the Transmit Data Ready Interrupt flag (eSCI_IFSR2[TXRDY]) should assert after the transmission of the Identifier (ID) field. In the TX frame generation, however, the eSCI_IFSR2[TXRDY] asserts after the Sync field. All subsequent TXRDY Interrupt flags in the current frame assert after each subsequent byte field has been transmitted except for the final TXRDY Interrupt flag. The last TXRDY Interrupt flag asserts after the transmission of the checksum field.

Workaround: The timing of the TXRDY Interrupt flag cannot be changed from the incorrect behavior. The incorrect TXRDY Interrupt flag behavior does not affect LIN functionality. Even though the TXRDY Interrupt flag asserts earlier than expected, the TXRDY Interrupt flag still signals that the content of the LIN Transmit Register (eSCI_LTR) was processed by the LIN Protocol Engine.

ERR002340: FEC: slot time is designed for 516 bit times; deviation from the 802.3

Description: The Fast Ethernet Controller (FEC) slot time is 516 bit times which is longer than the 512 bit times specified by the IEEE 802.3 standard.

If a collision occurs after the standard 512 bit times (but prior to 516 bit times), the FEC may generate a retry that a remote ethernet device may identify as late. In addition, the slot time is used as an input to the backoff timer, therefore the FEC retry timing could be longer than expected.

Workaround: No software workaround is needed or available.

ERR010546: Flash: Flash Memory Data Disturb due to Sleep Cycling and Power Cycling

Description: It is possible when doing multiple power and sleep cycles to the flash (greater than 1M), that the accumulation of soft program disturb condition may cause NVM memory array contents to be changed over time (data 1s change to 0s).

Workaround: Workarounds include the following:

1) limit the number of power and sleep cycles 2) if sleep cycles were done to reduce power an alternative mode is stop mode 3) if limiting the number of power and sleep cycles cannot be avoided then, periodic checksums of the contents of the array can be performed to test for disturbed contents. Refresh as frequent as necessary.

ERR003853: FLASH: Flash array access may be incorrect after sleep mode exit or after power up

Description: After exiting sleep mode or power up, a flash read may return incorrect data or a program/erase operation may fail. The condition will persist from sleep mode exit or power up until the next reset assertion.

Workaround: Flash read: Software Watchdog Timer (SWT) needs to be enabled at the time the micro exits sleep mode and exits its reset sequence after power up which provides a recovery mechanism if code execution fails.

Flash program/erase: After exiting sleep mode and power up, the device must be reset before performing flash program/erase operations. However, through proper usage of software mechanisms such as EE emulation algorithms in addition to Error Correction Code (ECC), the probability of observing a failure due to ineffective programing or erasing will be greatly reduced.

ERR002382: FLASH: Flash Array Integrity Check

Description: The Flash Array Integrity Check (AIC) which may be enabled during the flash user test (UTest) mode does not return the expected UMn[MISR] values for some flash PFCRPn[RWSC] read wait state configurations. For PFCRPn[RWSC] values of 3-6, the UMn[MISR] signature computation during AIC does not include the data read from the very last address in the selected address sequence and thus the UMn[MISR] value is not as expected. For PFCRPn[RWSC] values of 7, the UMn[MISR] signature computation during AIC will not be correct as well.

Workaround: The Flash Array Integrity Check is correct for PFCRPn[RWSC] values of 0-2. For PFCRPn[RWSC] values of 3-6, the expected UMn[MISR] values will not include the data read from the very last address and thus the value expected should be for the data read up to the 2nd-last address in the selected address sequence. For a PFCRPn[RWSC] value of 7, the Array Integrity Check should not be used at all.

ERR003659: FLASH: Resuming after a suspend during an Erase may prevent the erase from completing.

Description: If an erase suspend (including the flash put into sleep or disabled mode) is done on any block in the low Address Space (LAS) or the Mid-Address Space (MAS) except the 16 KB blocks, or if a suspend is done with multiple non-adjacent blocks (including the High Address Space [HAS]), the flash state machine may not set the FLASH_MCR[DONE] bit in the flash Module Control Register. This condition only occurs if the suspend occurs during certain internal flash erase operations. The likelihood of an issue occurring is reduced by limiting the frequency of suspending the erase operation.

Workaround: If the suspend feature (including disable and sleep modes) of the flash is used, then software should ensure that if the maximum time allowed for an erase operation occurs without a valid completion flag from the flash (FLASH_MCR[DONE] = 1), the software should abort the erase operation (by first clearing the Enable High Voltage (FLASH_MCR[EHV]) bit, then clearing the Erase read/Write bit (FLASH_MCR[ERS] bit) and the erase operation should be restarted.

Note: The cycle count of the sector is increased by this abort and restart operation.

ERR007322: FlexCAN: Bus Off Interrupt bit is erroneously asserted when soft reset is performed while FlexCAN is in Bus Off state

Description: Under normal operation, when FlexCAN enters in Bus Off state, a Bus Off Interrupt is issued to the CPU if the Bus Off Mask bit (CTRL[BOFF_MSK]) in the Control Register is set. In consequence, the CPU services the interrupt and clears the ESR[BOFF_INT] flag in the Error and Status Register to turn off the Bus Off Interrupt.

In continuation, if the CPU performs a soft reset after servicing the bus off interrupt request, by either requesting a global soft reset or by asserting the MCR[SOFT_RST] bit in the Module Configuration Register, once MCR[SOFT_RST] bit transitions from 1 to 0 to acknowledge the soft reset completion, the ESR[BOFF_INT] flag (and therefore the Bus Off Interrupt) is re-asserted.

The defect under consideration is the erroneous value of Bus Off flag after soft reset under the scenario described in the previous paragraph.

The Fault Confinement State (ESR[FLT_CONF] bit field in the Error and Status Register) changes from 0b11 to 0b00 by the soft reset, but gets back to 0b11 again for a short period, resuming after certain time to the expected Error Active state (0b00). However, this late correct state does not reflect the correct ESR[BOFF_INT] flag which stays in a wrong value and in consequence may trigger a new interrupt service.

Workaround: To prevent the occurrence of the erroneous Bus Off flag (and eventual Bus Off Interrupt) the following soft reset procedure must be used:

1. Clear CTRL[BOFF_MSK] bit in the Control Register (optional step in case the Bus Off Interrupt is enabled).

2. Set MCR[SOFT_RST] bit in the Module Configuration Register.

3. Poll MCR[SOFT_RST] bit in the Module Configuration Register until this bit is cleared.

4. Wait for 4 peripheral clocks.

5. Poll ESR[FLTCONF] bit in the Error and Status Register until this field is equal to 0b00.

6. Write “1” to clear the ESR[BOFF_INT] bit in the Error and Status Register.

7. Set CTRL[BOFF_MSK] bit in the Control Register (optional step in case the Bus Off Interrupt is enabled).

ERR008951: I2C: Attempting a start cycle while the bus is busy may generate a short clock pulse

Description: When the I2C (Inter-Integrated Circuit) is operating in a multi-master network and a start cycle is attempted by the I2C device when the bus is busy, the attempting master will lose arbitration as expected but a short extra clock cycle is generated in the bus. After losing arbitration, the

master switches to slave mode but it does not detect the short clock pulse. The acknowledge signal is expected at the ninth clock by the current bus master but it is not sent as expected due to the undetected short clock pulse.

Workaround: Software must ensure that the I2C BUS is idle by checking the bus busy bit in the I2C Bus Status Register (I2C_IBSR.IBB) before switching to master mode and attempting a Start cycle.

ERR002501: LPM: High VDD Sleep Mode Current

Description: The VDD Sleep Mode currents are 350uA (typ, 25C) / 2500uA (max, 150C) compared to the targets in the preliminary Data Sheet specification of 100uA (typ, 25C) / 900uA (max, 150C).

Workaround: None.

ERR006726: NPC: MCKO clock may be gated one clock period early when MCKO frequency is programmed as SYS_CLK/8.and gating is enabled

Description: The Nexus auxiliary message clock (MCKO) may be gated one clock period early when the MCKO frequency is programmed as SYS_CLK/8 in the Nexus Port Controller Port Configuration Register (NPC_PCR[MCKO_DIV]=111) and the MCKO gating function is enabled (NPC_PCR[MCKO_GT]=1). In this case, the last MCKO received by the tool prior to the gating will correspond to the END_MESSAGE state. The tool will not receive an MCKO to indicate the transition to the IDLE state, even though the NPC will transition to the IDLE state internally. Upon re-enabling of MCKO, the first MCKO edge will drive the Message Start/End Output (MSEO=11) and move the tool's state to IDLE.

Workaround: Expect to receive the MCKO edge corresponding to the IDLE state upon re-enabling of MCKO after MCKO has been gated.

ERR004146: SARADC: Interrupted conversions are aborted, but may not be properly restored

Description: When a triggered conversion interrupts an in process conversion in the Successive Approximation Analog to Digital Converter (SARADC), it is possible that the aborted conversion does not get restored to the SARADC and is not converted during the chain. Vulnerable configurations are:

- Injected chain over a normal chain
- Cross Trigging Unit (CTU) trigger over a normal chain
- CTU trigger over an injected chain

When any of these triggers arrive while the SARADC is in the conversion stage of the sample and conversion, the sample is discarded and is not restored. This means that the channel data register (SARADC_xCDRn) will not show the channel as being valid and the register SARADC_xCIPRn field CEOCFRx will not indicate a pending conversion. The sample that was aborted is lost.

If the injection occurs when the finite state machine switches from the sample phase, it is possible that on resuming normal chain, the chain is restored from an incorrect channel. This may lead to a second conversion on one of the channels in the chain.

When the trigger arrives during the final (last) channel conversion in a normal or injected chain, the same failure mode can cause two ECH (End of chain) interrupts to be raised in the interrupt register (SARADC_ISR).

If the trigger arrives during the sampling phase of the last channel in the chain, an ECH is triggered immediately, the trigger is processed and the channel is restored and after sampling/ conversion, a second ECH interrupt occurs.

In scan mode, the second ECH does not occur if the trigger arrives during the conversion phase. In one-shot mode, a trigger arriving during the conversion phase of the last channel restarts the whole conversion chain and the next ECH occurs at completion of that chain.

Workaround: The application should check for valid data using the Channel Data Register, Internal Channel Data Register or Test Channel Data Register (CDR) status bits or the CEOCFRx registers to ensure all expected channels have converted. This can be tested by running a bitwise AND and an XOR with either the Channel Conversion Mask Register (JCMRx) or the Channel Conversion Mask Register (NCMRx and the CEOCFRx registers during the JECH handler. Any non-zero value for (xCMRx & (xCMRx $\oplus$ CEOCFRx)) / (SARADC_ICxCMRn & (SARADC_ICxCMRn $\oplus$ SARADC_IPICRn.EOC_CHx)) indicates that a channel has been missed and conversion should be requested again.

Spurious ECH interrupts can be detected by checking the NSTART/JSTART flags in the SARADC_MSR Module Status Registers – if the flag remains set during an ECH interrupt then another interrupt will follow after the restored channel or chain has been sampled and converted.

The spurious ECH workaround above applies to single-shot conversions. In single-shot mode, NSTART changes from 1 to 0. Therefore, the user can rely on checking the NSTART bit to confirm if a spurious ECH has occurred. However, for scan mode, the NSTART bit will remain set during normal operation, so it cannot be relied upon to check for the spurious ECH issue. Consequently, if the CTU is being used in trigger mode, the conversions must be single-shot and not scan mode.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP 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. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2021 NXP B.V.