

## Mask Set Errata for Mask 0N78S

This report applies to mask 0N78S for these products:

- MPC5748G
- MPC5747G
- MPC5746G
- MPC5748C
- MPC5747C

### Mask Specific Information

JTAG identifier	0x0988201D
-----------------	------------

**Table 1. Errata and Information Summary**

Erratum ID	Erratum Title
ERR050154	Clocking: Device operation is impacted with a particular MC_CGM system divider configuration.
ERR011287	CMU: Sudden loss of clock does not signal the Fault Collection and Control Unit
ERR050090	DSPI/SPI: Incorrect data may be transmitted in slave mode
ERR010542	DSPI: Transmit, Command, and Receive FIFO fill flags in status register is not cleared when DMA is improperly configured
ERR010385	e200z4: Incorrect branch displacement at 16K memory boundaries
ERR010452	eDMA: When master ID replication is enabled, the stored ID and privilege level will change if read by another master.
ERR011235	EMIOS: Any Unified Channel running in OPWMB or OPWMCB mode may function improperly if the source counter bus is generated by Unified channel in MC mode
ERR050575	eMIOS: Any Unified Channel running in OPWMCB mode may function improperly if the lead or trail dead time insertion features is used and its timebase is generated by Unified channel in MCB mode
ERR011293	EMIOS: For any UC operating in OPWFMB mode the Channel Count register should not be written with a value greater than Channel B Data register value
ERR011295	EMIOS: In OPWFMB mode, A1/B1 registers do not get reloaded with A2/B2 register values if counter value returns 0x1 after counter wrap condition
ERR011294	EMIOS: OPWFMB and MCB mode counter rollover resets the counter to 0x0 instead of 0x1 as mentioned in the specification
ERR009978	eMIOS: Unexpected channel flag assertion during GPIO to MCB mode transition
ERR050395	ENET: Ethernet RX hang when receiving traffic through multiple queues

*Table continues on the next page...*



**Table 1. Errata and Information Summary (continued)**

Erratum ID	Erratum Title
ERR010594	ENET: The ENET1 (Ethernet) module does not function unless the Peripheral control register (MC_ME_PCTL6) is enabled .
ERR010963	Flash: Memory accesses may be corrupted when flash is operating between 33MHz and 75MHz
ERR007991	FLASH: Rapid Program or Erase Suspend fail status
ERR010595	FlexCAN: FLEXCAN1-7 modules will not work unless the Fast External Oscillator (FXOSC) clock source is enabled
ERR050246	FlexCAN: Receive Message Buffers may have its Code Field corrupted if the Receive FIFO function is used
ERR010368	FlexCAN: Transition of the CAN FD operation enable bit may lead FlexCAN logic to an inconsistent state.
ERR010620	FlexRay: The FS80 clock source should not be selected for the FlexRay protocol clock when the MCU clocking is configured for Linear Dynamic Frequency Scaling
ERR050119	FlexRay: Disabling of FlexRay Message Buffer during the STARTUP Protocol State takes longer than expected three Slots
ERR008770	FlexRAY: Missing TX frames on Channel B when in dual channel mode and Channel A is disabled
ERR008180	HSM: e200z0 Nexus interface DQTAG implemented as variable length field in DQM message
ERR010762	HSM: When the Hardware Security Module (HSM) is enabled, execution of selftest will lead to the HSM Destructive Reset Flag being set
ERR010849	IAHB: Programming of PCM Pending Read Enable can lead to a master stalling or receiving incorrect or spurious data
ERR008933	LINFlexD: Inconsistent sync field may cause an incorrect baud rate and the Sync Field Error Flag may not be set
ERR010141	LPU: LPU_RUN mode system clock must be preconfigured for undivided FIRC prior to LPU_STANDBY entry
ERR010609	MC_CGM: CLKOUT_0 and CLKOUT_1 may stop if the clock selection is changed when configured for divide by 2
ERR050195	MEMU_0: MEMU_0 operation is impacted with a particular MC_CGM system divider configuration.
ERR010603	NPC: Nexus Port Controller (NPC) must be enabled to allow mode changes during debug
ERR010723	NPC: Repeated Nexus3 Debug Status messages can be observed if more than one master (including a device core) is active and the core is subsequently disabled
ERR010340	NZxC3: ICNT and HIST fields of a Nexus message are not properly reset following a device reset
ERR010396	PASS: Password challenge to PASS fails while program erase ongoing in any block in memory partition 0
ERR009873	PFLASH: Calibration remap to flash memory not supported on 16KB and 32KB flash blocks in address range 0x00F90000-0x00FBFFFF
ERR010789	PFLASH:EEPROM ECC error suppression is not supported on 16KB and 32KB flash blocks in the address range 0x00F90000-0x00FBFFFF
ERR050130	PIT: Temporary incorrect value reported in LMTR64H register in lifetimer mode
ERR011321	PIT_RTI: Generates false RTI interrupt on re-enabling
ERR010590	PLL: Might remain unlocked when enabled after Standby or power on
ERR050467	PLL: Possible loss of lock when using the PLL bypass calibration mode
ERR011096	SAI: Internal bit clock is not generated when RCR2[BCI]=1 or TCR2[BCI]=1
ERR011150	SAI: Internally generated receive or transmit BCLK cannot be re-enabled if it is first disabled when RCR2[DIV] or TCR2[DIV] > 0

Table continues on the next page...

**Table 1. Errata and Information Summary (continued)**

Erratum ID	Erratum Title
ERR050144	SAI: Setting FCONT=1 when TMR>0 may not function correctly
ERR011306	SAR ADC: Incorrect value of ADC power down exit delay evaluated by the formula given in PDED [PDED] field description
ERR050572	SIRC: Clock output may contain extra clock pulses
ERR050196	STANDBY EXIT: Device may not come out of reset if a short functional reset comes immediately after wakeup event.
ERR010815	STCU2: Offline LBIST execution will toggle GPIO[29], GPIO[30], GPIO[62] and GPIO[63]
ERR010721	STCU: If the Auxiliary Clock 9 Select Control Register CGM_AC9_SC[SELCTL] = 1, offline-BIST will fail leading to a STCU watchdog timeout
ERR010200	STM: Reading the System Timer Module Count register may return an incorrect value
ERR006308	USB: Host non-doubleword –aligned buffer address can cause host to hang on OUT Retry

**Table 2. Revision History**

Revision	Changes
1. February 2017	Initial revision
2, January 2018	<p>The following errata were added.</p> <ul style="list-style-type: none"> <li>• ERR010396</li> <li>• ERR010542</li> <li>• ERR010620</li> <li>• ERR010815</li> <li>• ERR010849</li> <li>• ERR011096</li> <li>• ERR011150</li> </ul> <p>The following errata were revised.</p> <ul style="list-style-type: none"> <li>• ERR010200</li> <li>• ERR010609</li> <li>• ERR010721</li> <li>• ERR010789</li> </ul>
3, March 2021	<p>The following erratum was removed.</p> <ul style="list-style-type: none"> <li>• ERR010577</li> </ul> <p>The following errata were added.</p> <ul style="list-style-type: none"> <li>• ERR010385</li> <li>• ERR010963</li> <li>• ERR011235</li> <li>• ERR011287</li> <li>• ERR011293</li> <li>• ERR011294</li> <li>• ERR011295</li> <li>• ERR011321</li> <li>• ERR050090</li> <li>• ERR050119</li> <li>• ERR050130</li> </ul>

*Table continues on the next page...*

**Table 2. Revision History (continued)**

Revision	Changes
	<ul style="list-style-type: none"> <li>• ERR050144</li> <li>• ERR050154</li> <li>• ERR050195</li> <li>• ERR050196</li> <li>• ERR050246</li> <li>• ERR050395</li> <li>• ERR050467</li> <li>• ERR050572</li> </ul> <p>The following erratum was revised.</p> <ul style="list-style-type: none"> <li>• ERR010340</li> </ul>
4, August 2021	<p>The following errata were added.</p> <ul style="list-style-type: none"> <li>• ERR006308</li> <li>• ERR010590</li> <li>• ERR011306</li> <li>• ERR050575</li> </ul> <p>The following errata were revised.</p> <ul style="list-style-type: none"> <li>• ERR010963</li> <li>• ERR011235</li> <li>• ERR050246</li> </ul>
5, March 2022	<p>The following erratum was revised.</p> <ul style="list-style-type: none"> <li>• ERR050575</li> </ul>

**ERR050154: Clocking: Device operation is impacted with a particular MC\_CGM system divider configuration.**

**Description:** BAF issues SWT0 timeout destructive reset if the below conditions are met:

- 1) MC\_CGM divider configuration was programmed in application such that ratio of MC\_CGM\_SC\_DC0[DIV] and MC\_CGM\_SC\_DC5[DIV] is 1 and
- 2) Short functional reset is issued.

**Workaround:** Do not configure any functional reset source in MC\_RGM\_FESS to generate short functional reset in this clocking configuration.

**ERR011287: CMU: Sudden loss of clock does not signal the Fault Collection and Control Unit**

**Description:** The Clock Monitor Unit (CMU) detects when a monitored clock frequency drops below a programmed threshold through the Frequency Less than Low Threshold (FLL) signal. This FLL signal is routed to the Fault Collection and Control Unit ( FCCU ) providing a mechanism to react to the clock fault. Due to it's implementation, the FLL signal will not be triggered when the monitored clock source suddenly stops.

**Workaround:** The CMU has an internal signal which is designed to give an indication that the monitored clock has dropped below 1/4 of the reference clock CLKMTO\_RMN. This provides an alternative means to detect the sudden loss of clock, however since this internal signal is not routed to the FCCU, the user software must periodically poll bitfield [3] of CMU\_ISR register to detect a sudden loss of clock. Write '0b1' to clear the bitfield [3] of CMU\_ISR after enabling CMU.

#### **ERR050090: DSPI/SPI: Incorrect data may be transmitted in slave mode**

**Description:** If the Serial Peripheral Interface (SPI or the Deserial/Serial Peripheral Interface) is operating in slave mode, incorrect or stale data may be transmitted in next transaction without underflow interrupt generation if the set up time of the Peripheral Chip Select (PCS) to the SPI Serial Clock (SCLK) is short and the transmit FIFO may become empty after one transaction.

This can occur if the PCS to SCK is less than:

$$4 \times \text{IPG\_CLOCK\_PERIOD} + 4 \times \text{DSPI\_CLOCK\_PERIOD} + 0.5 \times \text{SCK\_CLOCK\_PERIOD}$$

Where:

IPG\_CLOCK is the internal bus clock ("system" clock)

DSPI\_CLOCK is the protocol clock.

SCK\_CLOCK is the Line-Side Serial Communication Clock.

**Workaround:** When operating in slave mode, software must ensure that the time interval between PCS assertion to start of SCK Clock is greater than  $4 \times \text{IPG\_CLOCK\_PERIOD} + 4 \times \text{DSPI\_CLOCK\_PERIOD} + 0.5 \times \text{SCK\_CLOCK\_PERIOD}$ .

To meet this requirement, the Master SPI can either lengthen the PCS to SCK assertion time or decrease the frequency of the communication interface, or both.

#### **ERR010542: DSPI: Transmit, Command, and Receive FIFO fill flags in status register is not cleared when DMA is improperly configured**

**Description:** The Deserial/Serial Peripheral Interface Transmit, Receive, and Command First In/First Out (FIFO) buffers can request additional information to be transferred via the Direct Memory Access (DMA) module when either the Transmit, Receive, or Command FIFO Fill/Drain Flags are set in the DSPI Status Register (SR[TFFF/RDFD/CMDFFF]). However, the Command/Transmit Fill Flag only indicates that at least 1 location in the FIFO is available to be written. It does not indicate that the FIFO is empty. Similarly, Receive FIFO fill flag only indicates at least 1 location of the FIFO is available to be read. It does not indicate that the FIFO is full. If the DMA is configured to transfer more than 1 FIFO location size of data, the FIFO Fill Flags may not be properly cleared indicating that the FIFO is not full even when the FIFO is actually full (for Transmit and Command FIFO) and not empty when the FIFO is actually empty (for Receive FIFO).

**Workaround:** Properly configure the DMA to fill the Transmit, Receive, and Command FIFOs only one FIFO location, in other words, up to 2 bytes, at a time to each of the FIFOs.

Use the DMA loop to transfer more data if needed.

## **ERR010385: e200z4: Incorrect branch displacement at 16K memory boundaries**

**Description:** The branch target address will be incorrectly calculated in the e200z4 core under the following conditions (all conditions must be matched):

- The first full instruction in a 16 Kbyte section/page of code is a 32-bit long branch with a branch displacement value with the lower 14 bits of the displacement exactly 0x3FFE
- And this branch instruction is located at byte offset 0x0002 in the section/page
- And the preceding instruction is a 32-bit length instruction which is misaligned across the 16K boundary
- And both instructions are dual-issued

Under these conditions, the branch target address will be too small by 32Kbytes.

**Workaround:** After software is compiled and linked, code should be checked to ensure that there are no branch instructions located at address 0x2 of any 16K memory boundary with the lower 14 bits of the displacement equal to 0x3FFE if preceded a 32-bit instruction that crosses the 16K memory boundary. If this sequence occurs, add a NOP instruction or otherwise force a change to the instruction addresses to remove the condition.

A tool is available on [nxp.com](http://nxp.com) that can be run to examine code for this condition, search for `branch_displacement_erratum_10385_checker`.

## **ERR010452: eDMA: When master ID replication is enabled, the stored ID and privilege level will change if read by another master.**

**Description:** When master ID replication is enabled (`DMA_DCHMIDn[EMI]=1`), the `DMA_DCHMIDn[PAL]` and `DMA_DCHMIDn[MID]` fields should reflect the privilege level and master ID respectively of the master that wrote the `DMA_TCDn_CSR[DONE:START]` byte. However, if a different master reads the `DMA_TCDn_CSR[DONE:START]` byte, the master ID and privilege level will incorrectly change to this read access.

**Workaround:** Only allow the intended master ID replication core to access the `DMA_TCDn_CSR[DONE:START]` byte.

## **ERR011235: EMIOS: Any Unified Channel running in OPWMB or OPWMCB mode may function improperly if the source counter bus is generated by Unified channel in MC mode**

**Description:** The Unified channel (UC) configured in Center Aligned Output Pulse Width Modulation Buffered (OPWMCB) or Output Pulse Width Modulation Buffered (OPWMB) modes is not working properly when it is sourced from the UC configured in Modulus Counter (MC) mode by setting the eMIOS Channel Control Register (`EMIOS_CCR`) `MODE` bitfield to 0x10 or 0x11 and any of its pre-scalers (internal or global) divider ratio is higher than 1.

**Workaround:** When a counter bus is generated by the UC set in the MC mode with any pre-scaler (internal or global) divider ration higher than 1, don't use this counter bus for the UC set in OPWMCB or OPWMB mode.

**ERR050575: eMIOS: Any Unified Channel running in OPWMCB mode may function improperly if the lead or trail dead time insertion features is used and its timebase is generated by Unified channel in MCB mode**

**Description:** The Unified channel (UC) configured in Center Aligned Output Pulse Width Modulation Buffered (OPWMCB) mode is not working properly when:

1. It's timebase is sourced from the UC configured in Modulus Counter Buffered (MCB) mode.
2. The lead or trail dead time insertion features is used.
3. Its channel prescaler is different than timebase channel prescaler.

**Workaround:** Channel configured in OPWMCB mode with lead or trail dead time insertion features enabled must have channel prescaler equal to the timebase channel prescaler configured in MCB mode.

**ERR011293: EMIOS: For any UC operating in OPWFMB mode the Channel Count register should not be written with a value greater than Channel B Data register value**

**Description:** For any Unified Channel (UC) running in Output Pulse-Width and Frequency Modulation Buffered (OPWFMB) mode, Channel Control Register MODE bitfield = 7'h1011000 or 7'h1011010, the internal counter runs from 0x1 to Channel B Data register value.

The internal counter can be overwritten by software using the Channel Count register during 'freeze' operation.

If a counter wrap occurs due to overwriting of the counter with a value greater than its expiry value (B Data Register value); than the output signal behavior cannot be guaranteed.

**Workaround:** For any UC operating in OPWFMB mode the Channel Count register should not be written with a value greater than Channel B Data register value.

**ERR011295: EMIOS: In OPWFMB mode, A1/B1 registers do not get reloaded with A2/B2 register values if counter value returns 0x1 after counter wrap condition**

**Description:** In Output Pulse-Width and Frequency Modulation Buffered (OPWFMB) mode, A1/B1 registers do not get reloaded with A2/B2 register values if counter value returns 0x1 after counter wrap condition.

In order to avoid the counter wrap condition make sure internal counter value is within the 0x1 to B1 register value range when the OPWFMB mode is entered. Also overwriting of Channel Count register by forcing 'freeze' in OPWFMB mode should not take internal counter outside 0x1 to B register value.

**Workaround:** In order to avoid the counter wrap condition:

1. Make sure internal counter value is within the 0x1 to (B1 register) value range when the OPWFMB mode is entered.
2. Overwrite of Channel Count register by forcing 'freeze' in OPWFMB mode should not be outside the range of 0x1 to (B register) value.

## **ERR011294: EMIOS: OPWFMB and MCB mode counter rollover resets the counter to 0x0 instead of 0x1 as mentioned in the specification**

**Description:** When the enhanced Modular Input/Output System (eMIOS) is used in Output Pulse-Width and Frequency Modulation Buffered (OPWFMB) or Modulus Counter Buffered (MCB) modes, when the counter rolls over, the counter returns to 0x0 instead of 0x1 as specified in the reference manual.

**Workaround:** In order to avoid the counter wrap condition:

1. Make sure internal counter value is within the 0x1 to (B1 register) value range when the OPWFMB mode is entered.
2. Overwrite of Channel Count register by forcing 'freeze' in OPWFMB mode should not be outside the range of 0x1 to (B register) value.

## **ERR009978: eMIOS: Unexpected channel flag assertion during GPIO to MCB mode transition**

**Description:** When changing an Enhanced Modular IO Subsystem (eMIOS) channel mode from General Purpose Input/Output (GPIO) to Modulus Counter Buffered (MCB) mode, the channel flag in the eMIOS Channel Status register (eMIOS\_Sn[FLAG]) may incorrectly be asserted. This will cause an unexpected interrupt or DMA request if enabled for that channel.

**Workaround:** In order to change the channel mode from GPIO to MCB without causing an unexpected interrupt or DMA request, perform the following steps:

- (1) Clear the FLAG enable bit in the eMIOS Control register (eMIOS\_Cn[FEN] = 0).
- (2) Change the channel mode (eMIOS\_Cn[MODE]) to the desired MCB mode.
- (3) Clear the channel FLAG bit by writing '1' to the eMIOS Channel Status register FLAG field (eMIOS\_Sn[FLAG] = 1).
- (4) Set the FLAG enable bit (eMIOS\_Cn[FEN] = 1) to re-enable the channel interrupt or DMA request reaction.

## **ERR050395: ENET: Ethernet RX hang when receiving traffic through multiple queues**

**Description:** Two or more applications are enabled to share the same Ethernet module by using different queues. At least 2 queues are configured to receive packets, with flushing enabled (RX\_FLUSHx). When queues become full, packets are normally flushed, but under certain conditions of traffic, a lock-up of the Rx path can happen instead. When this occurs, the buffer descriptor for the last received packet contains an incorrect packet size (equal to the maximum buffer size). Packets cannot be received anymore, but the TX path remains unaffected. To recover the RX path, the ENET hardware block must be reset and re-configured.

**Workaround:** Unless the use case demands it, disable flushing to ensure the problem does not happen.

Or if reset is acceptable:

To recover the RX path, the ENET hardware block must be reset and re-configured



## **ERR010594: ENET: The ENET1 (Ethernet) module does not function unless the Peripheral control register (MC\_ME\_PCTL6) is enabled .**

**Description:** The Ethernet module ENET1 does not function unless the MLB Peripheral control register (MC\_ME\_PCTL6) is enabled. This does not apply to ENET0.

**Workaround:** Enable MC\_ME\_PCTL6 if using the ENET1 module.

## **ERR010963: Flash: Memory accesses may be corrupted when flash is operating between 33MHz and 75MHz**

**Description:** Error Correction Code (ECC) errors may be generated in the flash due to corrupted data when all of the following conditions are met:

- The flash operating frequency is greater than 33MHz and less than or equal to 75MHz
- Read Wait State Control (PFLASH\_PFCR1[RWSC]) = 2 and Address Pipeline Control (PFLASH\_PFCR1[APC]) = 1
- Pipelined reads which are executed between the UTEST flash block and any other flash block. Pipelined reads are overlapping reads, where before the previous read is completed a new read is requested.

Device has UTEST memory space and remaining flash area is the main array memory space. This issue condition occurs if pipelined reads are executed between UTEST and the main array space. If pipelined reads are executed between UTEST and UTEST memory space or between main array and main array memory space, this issue condition will not be seen.

**Workaround:** When the flash clock frequency is greater than 33MHz and less than or equal to 75MHz configure PFLASH\_PFCR1[RWSC] = 2 and PFLASH\_PFCR1[APC] = 0. The configuration for all other flash clock frequencies is specified in the MCU datasheet.

## **ERR007991: FLASH: Rapid Program or Erase Suspend fail status**

**Description:** If a flash suspend operation occurs during a 5us window during a verify operation being executed by the internal flash program and erase state machine, and the suspend rate continues at a consistent 20us rate after that, it is possible that the flash will not exit the program or erase operation. A single suspend during a single program or erase event will not cause this issue to occur.

Per the flash specification, a flash program or erase operation should not be suspended more than once every 20 us, therefore, if this requirement is met, no issue will be seen. IF the suspend rate is faster than 20 us continuously, a failure to program/erase could occur.

**Workaround:** When doing repeated suspends during program or erase ensure that suspend period is greater than 20us.

## **ERR010595: FlexCAN: FLEXCAN1-7 modules will not work unless the Fast External Oscillator (FXOSC) clock source is enabled**

**Description:** FLEXCAN modules 1-7 will not work unless the Fast External Oscillator (FXOSC) clock source is enabled on the device.

**Workaround:** The FXOSC clock should be enabled before using FLEXCAN1-7 modules by setting the Oscillator Enable bit (FXOSCON) in the active mode configuration register (MC\_ME\_xxxx\_MC).

## **ERR050246: FlexCAN: Receive Message Buffers may have its Code Field corrupted if the Receive FIFO function is used**

**Description:** If the Code Field of a Receive Message Buffer is corrupted it may deactivate the Message Buffer, so it is unable to receive new messages. It may also turn a Receive Message Buffer into any type of Message Buffer as defined in the Message buffer structure section in the device documentation.

The Code Field of the FlexCAN Receive Message Buffers (MB) may get corrupted if the following sequence occurs.

- 1- A message is received and transferred to an MB (i.e. MBx)
- 2- MBx is locked by software for more than 20 CAN bit times (time determines the probability of erratum to manifest).
- 3- SMB0 (Serial Message Buffer 0) receives a message (i.e. message1) intended for MBx, but destination is locked by the software (as depicted in point 2 above) and therefore NOT transferred to MBx.
- 4- A subsequent incoming message (i.e. message2) is being loaded into SMB1 (as SMB0 is full) and is evaluated by the FlexCAN hardware as being for the FIFO.
- 5- During the message2, the MBx is unlocked. Then, the content of SMB0 is transferred to MBx and the CODE field is updated with an incorrect value.

The problem does not occur in cases when only Rx FIFO or only a dedicated MB is used (i.e. either RX MB or Rx FIFO is used). The problem also does not occur when the Enhanced Rx FIFO and dedicated MB are used in the same application. The problem only occurs if the FlexCAN is programmed to receive in the Legacy FIFO and dedicated MB at the same application.

**Workaround:** This defect only applies if the Receive FIFO (Legacy Rx FIFO) is used. This feature is enabled by RFEN bit in the Module Control Register (MCR). If the Rx FIFO is not used, the Receive Message Buffer Code Field is not corrupted.

If available on the device, use the enhanced Rx FIFO feature instead of the Legacy Rx FIFO. The Enhanced Rx FIFO is enabled by the ERFEN bit in the Enhanced Rx FIFO Control Register (ERFCR).

The defect does not occur if the Receive Message Buffer lock time is less than or equal to the time equivalent to 20 x CAN bit time.

The recommended way for the CPU to service (read) the frame received in a mailbox is by the following procedure:

1. Read the Control and Status word of that mailbox.
2. Check if the BUSY bit is deasserted, indicating that the mailbox is not locked. Repeat step 1) while it is asserted.
3. Read the contents of the mailbox.
4. Clear the proper flag in the IFLAG register.
5. Read the Free Running Timer register (TIMER) to unlock the mailbox

In order to guarantee that this procedure occurs in less than 20 CAN bit times, the MB receive handling process in software (step 1 to step 5 above) should be performed as a 'critical code section' (interrupts disabled before execution) and should ensure that the MB receive handling occurs in a deterministic number of cycles.

## **ERR010368: FlexCAN: Transition of the CAN FD operation enable bit may lead FlexCAN logic to an inconsistent state.**

**Description:** The activation or deactivation of the CAN FD operation by setting or clearing the FDEN bit of the CAN\_MCR register or by setting the FlexCAN soft reset bit (SOFTRST) of the CAN\_MCR register when the FDEN bit is enabled may cause an internal FlexCAN register to become metastable. As result, the first CAN frame, transmitted or received, may have corrupted data (ID and payload). However, even though the data is corrupted, a valid CAN frame is transmitted because the Cyclic Redundancy Check (CRC) calculation is based on the corrupted data. During reception the data is corrupted internally after the CRC bits have been checked and therefore this corrupted data may be stored in a reception message buffer. After the first CAN frame, all subsequent frames are transmitted and received correctly.

**Workaround:** Perform the following steps to set the FDEN bit:

1. If FlexCAN is already in freeze mode, go to step 3, otherwise set the HALT and FRZ bits of the CAN\_MCR register.
2. Wait the FRZACK bit of the CAN\_MCR register to be set by the hardware.
3. Set the LPB (Loop Back Mode) bit of the CAN\_CTRL1 register.
4. Configure only one message buffer to be transmitted. The frame should be a classical one (non-FD) with IDE =0, RTR =1 DLC =0x5 and STD\_ID =0x682.
5. Set the FDEN bit of the CAN\_MCR register.
6. Clear the HALT bit of the MCR register to leave freeze mode.
7. Wait the FRZACK bit of the CAN\_MCR register to be cleared by the hardware.
8. Wait the respective bit of the CAN\_IFLAG register to be set (successfully transmission in loop back mode).
9. Clear the respective bit of the CAN\_IFLAG register by writing 1.
10. Set the HALT and FRZ bits of the CAN\_MCR register.
11. Wait the FRZACK bit of the CAN\_MCR register to be set by the hardware.
12. Clear the LPB (Loop Back Mode) bit of the CAN\_CTRL1 register.

Perform the following steps to apply a soft reset or clear the FDEN bit:

1. If FlexCAN is already in freeze mode, go to step 3, otherwise set the HALT and FRZ bits of the CAN\_MCR register.
2. Wait the FRZACK bit of the CAN\_MCR register to be set by the hardware.
3. Set the SOFTRST bit of the CAN\_MCR register.
4. Wait the SOFTRST bit of the CAN\_MCR register to be cleared by the hardware.
5. Set again the SOFTRST bit of the CAN\_MCR register.
6. Wait the SOFTRST bit of the CAN\_MCR register to be cleared by the hardware.

## **ERR010620: FlexRay: The FS80 clock source should not be selected for the FlexRay protocol clock when the MCU clocking is configured for Linear Dynamic Frequency Scaling**

**Description:** The FlexRay module protocol clock can be selected from the FXOSC clock (default) or the FS80 clock and this is configured at FR\_MCR[CLKSEL]. When the MCU clock configuration is changed from the default state to the Linear DFS (Dynamic Frequency Scaling) clock mode, the FS80 must not be selected as the source for the FlexRay protocol clock.

**Workaround:** Prior to configuring the Linear DFS clock mode the user must select FXOSC for the FlexRay protocol clock.

## **ERR050119: FlexRay: Disabling of FlexRay Message Buffer during the STARTUP Protocol State takes longer than expected three Slots**

**Description:** Disabling of FlexRay Message Buffer takes longer than the expected three Slots. This is observed, when software application tries to disable the Message Buffer during the FlexRay STARTUP protocol state (vPOC!State = POC:startup) when vPOC!StartupState = "initialize schedule" or "integration consistency check".

In this scenario, FlexRay Communication Controller keeps the specific Message buffer search results until the availability of next cycle start/segment start/slot start events and therefore prevent the disabling of Message Buffer.

Note:

1. All Message Buffers can be disabled immediately if FlexRay protocol state (vPOC!State) is in following States: "POC:default config", "POC:config", "POC:wakeup", "POC:ready", "POC:halt", "POC:startup" and (vPOC!StartupState = "POC:integration listen" or "POC: ColdStart-Listen").

2. All Message Buffers can be disabled within three slots, if FlexRay protocol state (vPOC!State) is in following states: "POC: Normal-Active" or "POC: Normal-Passive".

**Workaround:** Do not disable Message Buffer, while FlexRay is in STARTUP protocol State

## **ERR008770: FlexRAY: Missing TX frames on Channel B when in dual channel mode and Channel A is disabled**

**Description:** If the FlexRay module is configured in Dual Channel mode, by clearing the Single Channel Device Mode bit (SCM) of the Module Control register (FR\_MCR[SCM]=0), and Channel A is disabled, by clearing the Channel A Enable bit (FR\_MCR[CHA]=0) and Channel B is enabled, by setting the Channel B enable bit (FR\_MCR[CHB]=1), there will be a missing transmit (TX) frame in adjacent minislots (even/odd combinations in Dynamic Segment) on Channel B for certain communication cycles. Which channel handles the Dynamic Segment or Static Segment TX message buffers (MBs) is controlled by the Channel Assignment bits (CHA, CHB) of the Message Buffer Cycle Counter Filter Register (FR\_MBCCFRn). The internal Static Segment boundary indicator actually only uses the Channel A slot counter to identify the Static Segment boundary even if the module configures the Static Segment to Channel B (FR\_MBCCFRn[CHA]=0 and FR\_MBCCFRn[CHB]=1). This results in the Buffer Control Unit waiting for a corresponding data acknowledge signal for minislot:N in the Dynamic Segment and misses the required TX frame transmission within the immediate next minislot:N+1.

**Workaround:** 1. Configure the FlexRay module in Single Channel mode (FR\_MCR[SCM]=1) and enable Channel B (FR\_MCR[CHB]=1) and disable Channel A (FR\_MCR[CHA]=0). In this mode the internal Channel A behaves as FlexRay Channel B. Note that in this mode only the internal channel A and the FlexRay Port A is used. So externally you must connect to FlexRay Port A.

2. Enable both Channel A and Channel B when in Dual Channel mode (FR\_MCR[CHA]=1] and FR\_MCR[CHB]=1). This will allow all configured TX frames to be transmitted correctly on Channel B.

**ERR008180: HSM: e200z0 Nexus interface DQTAG implemented as variable length field in DQM message**

**Description:** The Hardware Security Module (HSM) core (e200z0) implements the Data Tag (DQTAG) field of the Nexus Data Acquisition Message (DQM) as a variable length packet instead of an 8-bit fixed length packet. This may result in an extra clock (“beat”) in the DQM trace message depending on the Nexus port width selected for the device.

**Workaround:** Tools should decode the DQTAG field as a variable length packet instead of a fixed length packet.

**ERR010762: HSM: When the Hardware Security Module (HSM) is enabled, execution of selftest will lead to the HSM Destructive Reset Flag being set**

**Description:** When the HSM is enabled, a Logic Built in Self Test (LBIST) execution will cause the flag indicating an HSM Destructive Reset event in the Destructive Event Status Register (MC\_RGM\_DES[F\_HSM\_DEST]) to be set even though there has been no actual HSM destructive reset event triggered. Only the flag is set during LBIST, no actual reset occurs.

**Workaround:** Since the HSM does not function during LBIST, user software should not treat the the setting of the MC\_RGM\_DES[F\_HSM\_DEST] flag during LBIST execution as a security/safety issue. Users should clear this flag as part of the post LBIST software routine.

**ERR010849: IAHB: Programming of PCM Pending Read Enable can lead to a master stalling or receiving incorrect or spurious data**

**Description:** Overriding the default configuration of bus traffic optimization for specific masters on the device (Ethernet, eDMA, USB\_0, USB\_1, MLB, and uSDHC,Z2) can cause those masters to stall, receive wrong read data, or get a spurious read access when uncorrectable ECC errors are received from slaves.

**Workaround:** To avoid this situation, following configuration fields in the Platform Configuration Module (PCM) should not be modified:

PCM\_IAHB\_BE0[PRE\_ENET]  
PCM\_IAHB\_BE0[PRE\_DMA]  
PCM\_IAHB\_BE1[PRE\_uSDHC]  
PCM\_IAHB\_BE1[PRE\_USB\_1]  
PCM\_IAHB\_BE1[PRE\_USB\_0]

PCM\_IAHB\_BE1[PRE\_MLB]  
PCM\_IAHB\_BE2[PRE\_Z2\_DATA]  
PCM\_IAHB\_BE2[PRE\_Z2\_INST]

### **ERR008933: LINFlexD: Inconsistent sync field may cause an incorrect baud rate and the Sync Field Error Flag may not be set**

**Description:** When the LINFlexD module is configured as follows:

1. LIN (Local Interconnect Network) slave mode is enabled by clearing the Master Mode Enable bit in the LIN Control Register 1 (LINC1[MME] = 0b0)
2. Auto synchronization is enabled by setting LIN Auto Synchronization Enable (LINC1[LASE] = 0b1)

The LINFlexD module may automatically synchronize to an incorrect baud rate without setting the Sync Field Error Flag in the LIN Error Status register (LINESR[SFEF]) in case Sync Field value is not equal to 0x55, as per the Local Interconnect Network (LIN) specification.

The auto synchronization is only required when the baud-rate in the slave node can not be programmed directly in software and the slave node must synchronize to the master node baud rate.

**Workaround:** There are 2 possible workarounds.

Workaround 1:

When the LIN time-out counter is configured in LIN Mode by clearing the MODE bit of the LIN Time-Out Control Status register (LINTCSR[MODE]= 0x0):

1. Set the LIN state Interrupt enable bit in the LIN Interrupt Enable register (LINIER[LSIE] = 0b1)
2. When the Data Reception Completed Flag is asserted in the LIN Status Register (LINSR[DRF] = 0b1) read the LIN State field (LINSR[LINS])
3. If LINSR[LINS]= 0b0101, read the Counter Value field of the LIN Time-Out Control Status register (LINTCSR[CNT]), otherwise repeat step 2
4. If LINTCSR[CNT] is greater than 0xA, discard the frame.

When the LIN Time-out counter is configured in Output Compare Mode by setting the LINTCSR[MODE] bit:

1. Set the LIN State Interrupt Enable bit in the LIN Interrupt Enable register (LINIER[LSIE])
2. When the Data Reception Completed flag bit is asserted in the LIN Status Register (LINSR[DRF] = 0b1), read the LINSR[LINS] field
3. If LINSR[LINS]= 0b0101, store LINTCSR[CNT] value in a variable (ValueA), otherwise repeat step 2
4. Clear LINSR[DRF] flag by writing LINSR[LINS] field with 0xF
5. Wait for LINSR[DRF] to become asserted again and read LINSR[LINS] field
6. If LINSR[LINS] = 0b0101, store LINTCSR[CNT] value in a variable (ValueB), else repeat step 4
7. If ValueB – ValueA is greater than 0xA, discard the frame

Workaround 2:

Do not use the auto synchronization feature (disable with LINCR1[LASE] = 0b0) in LIN slave mode.

### **ERR010141: LPU: LPU\_RUN mode system clock must be preconfigured for undivided FIRC prior to LPU\_STANDBY entry**

**Description:** If the LPU\_RUN mode system clock is selected to be FXOSC or divided-FIRC when LPU\_STANDBY mode is entered then the MCU may not return to LPU\_RUN mode on a wake-up event.

In LPU\_RUN mode the FXOSC or divided-FIRC can be used as the system clock, but the user must ensure that the undivided FIRC is selected as the system clock before the LPU\_STANDBY mode transition is initiated.

**Workaround:** Prior to entering LPU\_STANDBY select undivided FIRC as the LPU System Clock by configuring LPU\_RUN\_CF[SYS\_CLK\_SEL] = 0 and FIRC\_CTL[FIRCDIV] = 5'b0.

### **ERR010609: MC\_CGM: CLKOUT\_0 and CLKOUT\_1 may stop if the clock selection is changed when configured for divide by 2**

**Description:** If the clock out functionality is enabled on either CLKOUT\_0 and/or CLKOUT\_1 and is configured for divide by 2 (via MC\_CGM\_AC6\_DC0[DE] and/or MC\_CGM\_CLKOUT1\_DC0[DE] = 0b1), then if the clock selection for CLKOUT\_0/CLKOUT\_1 is changed via MC\_CGM\_AC6\_SC[SELCTL]/MC\_CGM\_CLKOUT1\_SC[SELCTL] register respectively or a Destructive, Functional (long/short) reset occurs then the clock out may stop. The following clock sources when selected are affected:

- FXOSC
- FXOSC divided
- FXOSC ANA Clk
- SXOSC
- SXOSC divided
- SIRC
- SIRC divided
- PLL\_CLKOUT1
- PLL\_CLKOUT2
- RTC\_CLK
- CAN0 CHI clk (when driven by FXOSC, not affected when driven by FS80)
- CAN0 PE clk (when driven by FXOSC, not affected when driven by F40)

**Workaround:** Changing CLKOUT\_0/CLKOUT\_1 clock source selection value via software, resets all its corresponding dividers and recovers them.

Apply the following sequence after each reset for enabled CLKOUT\_0/CLKOUT\_1 clock dividers that are to be configured to divide by 2 for the application.

1. Disable the CLKOUT\_0 and/or CLKOUT\_1 clock divider by writing to MC\_CGM\_AC6\_DC0[DE] and/or MC\_CGM\_CLKOUT1\_DC0[DE] = 0b0
2. Change the CLKOUT\_0 and/or CLKOUT\_1 clock source selection to FIRC (MC\_CGM\_AC6\_SC[SELCTL] = 0b0001 and/or MC\_CGM\_CLKOUT1\_SC[SELCTL] = 0b1001).

3. Select the desired clock source as the CLKOUT\_0 and/or CLKOUT\_1 clock source (e.g. for FXOSC: MC\_CGM\_AC6\_SC[SELCTL] = 0b0000 and/or MC\_CGM\_CLKOUT1\_SC[SELCTL] = 0b1000).

4. Configure and enable the corresponding CLKOUT\_0 and/or CLKOUT\_1 clock divider by writing to MC\_CGM\_AC6\_DC0[DE] and/or MC\_CGM\_CLKOUT1\_DC0[DE] = 0b1.

**ERR050195: MEMU\_0: MEMU\_0 operation is impacted with a particular MC\_CGM system divider configuration.**

**Description:** Register read-write access on MEMU\_0 may not happen correctly if the ratio of MC\_CGM\_SC\_DC0[DIV] and MC\_CGM\_SC\_DC5[DIV] is 1.

**Workaround:** Do not perform register read write access on MEMU\_0 or disable MEMU\_0 when this clocking configuration is used.

**ERR010603: NPC: Nexus Port Controller (NPC) must be enabled to allow mode changes during debug**

**Description:** The Nexus Port Controller (NPC) must be enabled to allow mode changes via the Mode Entry module in debug mode. The e200zx core generates some Nexus trace messages automatically even when trace is not enabled if a Nexus Enable instruction is executed (typically used for Nexus read/Write access of memory by a tool). As a result, if the NPC is not enabled, the core will still see messages pending and never complete the requested mode change.

**Workaround:** Enable the NPC by enabling the Message Clock Output (MCKO\_EN = 1) in the NPC Port Configuration Register (NPC\_PCR).

**ERR010723: NPC: Repeated Nexus3 Debug Status messages can be observed if more than one master (including a device core) is active and the core is subsequently disabled**

**Description:** This errata applies to the condition where there is more than one master active on the Nexus Port Controller (NPC) module, and one or more of these masters is a device core. In this situation, if a mode transition is initiated to a mode where that device core is disabled, with the clock gated (as configured in the relevant core control register MC\_ME\_CCTLx for the requested mode) then message data can be left pending on the interface until the core clock resumes. This causes status message to be repeated several times and no other message from any other Nexus3 client can be transmitted causing potential debugger problems.

**Workaround:** While transitioning to a low power mode(STOP, STANDBY, LPU\_RUN), use the NPC Handshake by clearing NPC\_1 PCR [LP1\_SYNC] bit. The debugger can then disable the Nexus3 tracing of the core before it acknowledges that the transition into a low-power mode may proceed. For a non-low power mode transition (DRUN, RUNx), do not disable device core but instead use the Power Architecture 'wait' instruction to move the device core to the wait state.

Alternatively, transmit repeated or more than one TCODE messages from the active masters.



## **ERR010340: NZxC3: ICNT and HIST fields of a Nexus message are not properly reset following a device reset**

**Description:** Following reset, if instruction trace is enabled in the Nexus e200zx core Class 3 trace client (NZxC3), the e200zx core transmits a Program Trace – Synchronization Message (PT-SM). The PT-SM includes the full execution address and the number of instructions executed since the last Nexus message (ICNT) information. However, if Branch History trace is enabled, the ICNT and the Branch History (HIST) fields are not properly cleared when this message is transmitted. This may cause unexpected trace reconstruction results until the next Nexus Program Trace Synchronization Message (Program Trace – Direct Branch Message with Sync, Program Trace – Indirect Branch Message with Sync, or Program Trace – Indirect Branch History Message with Sync).

In Branch History mode, the first indirect branch following the reset (and the initial PT-SM) will contain the branch history prior to the reset plus the branch history after reset. However, there is no way to determine which branches occurred prior to reset and which followed reset.

**Workaround:** If not using branch history trace mode, to recreate the proper trace, the tool should take into account that the ICNT field is not cleared by the first PT-SM. The previous ICNT will be added to new ICNT value in the subsequent Nexus message. This may require extra processing by the tool.

If using branch history mode, then an accurate reconstruction of the executed code just before and just after reset may not be possible. Trace reconstruction can be recovered after the next indirect branch message.

On devices that bypass the Boot Assist Flash (BAF), Boot Assist Module (BAM), or BootROM after reset, perform an indirect branch instruction shortly after reset to reset the ICNT (and HIST if Branch History mode is enabled). Also, A full program trace synchronization message will be generated after 256 direct branches even if there is no indirect branches. This will allow the tool to recover the trace reconstruction from that point onward.

On devices that always execute boot from boot ROM firmware, the BAF or BAM, an indirect branch will occur during the boot ROM/BAF/BAM execution and the tool trace will be re-synchronized prior to the execution of user code.

## **ERR010396: PASS: Password challenge to PASS fails while program erase ongoing in any block in memory partition 0**

**Description:** If the device is in a Censored state (enabled by programming the censorship DCF in UTEST) and a JTAG password is configured to enable device debug access, then the password challenge to the PASS module would be initiated by programming the Challenge Selector Register (PASS\_CHSEL) to determine the password group, then programming the Challenge Input Registers (PASS\_CINn) with the correct password. Programming the correct password would then allow enabling of debug interface access.

However, this operation will fail if a program or erase operation is ongoing on any flash block in memory partition 0, since this is shared with the UTEST block where the JTAG password resides.

**Workaround:** Users should ensure that no program or erase operations are occurring on any memory partitions shared with the UTEST block before initiating a password challenge. This can be monitored through the flash module configuration register program and erase status bits (C55FMC\_MCR[PGM], C55FMC\_MCR[ERS]).

### **ERR009873: PFLASH: Calibration remap to flash memory not supported on 16KB and 32KB flash blocks in address range 0x00F90000-0x00FBFFFF**

**Description:** The PFLASH module supports calibration remapping of a flash access to another on-chip flash address. UTEST flash, BAF, and secure flash blocks cannot be remapped nor can accesses to other flash blocks be rerouted to addresses in UTEST flash, BAF, or secure flash. Flash blocks of size 16kB and 32KB in address range 0x00F90000-0x00FBFFFF do not support calibration remap to flash memory. All other flash blocks of size 32KB, 64KB 256KB in address range 0x00FC0000-0x0157FFFF can be overlaid using the mirrored address range.

**Workaround:** When using the calibration remapping of flash feature, the user must select flash blocks of size 32KB, 64KB 256KB in address range 0x00FC0000-0x0157FFFF.

### **ERR010789: PFLASH:EEPROM ECC error suppression is not supported on 16KB and 32KB flash blocks in the address range 0x00F90000-0x00FBFFFF**

**Description:** The PFLASH module supports the suppression of ECC event reporting on secure data flash blocks in the address range 0x00F80000 – 0x00F87FFF. For more information see MPC5748G Reference Manual section “ECC on data flash accesses”. Flash blocks of size 16KB and 32KB in address range 0x00F90000 – 0x00FBFFFF do not support suppression of error reporting on ECC events. When reading from the address range 0x00F90000-0x00FBFFFF any non-correctable ECC error will be reported as a bus error to the requesting master. Both correctable and non-correctable ECC errors are reported to the MEMU.

**Workaround:** The application software must handle bus errors due to non-correctable ECC errors in the flash memory region 0x00F90000 – 0x00FBFFFF.

### **ERR050130: PIT: Temporary incorrect value reported in LMTR64H register in lifier mode**

**Description:** When the Programmable interrupt timer (PIT) module is used in lifier mode, timer 0 and timer 1 are chained and the timer load start value (LDVAL0[TSV] and LDVAL1[TSV]) are set according to the application need for both timers. When timer 0 current time value (CVAL0[TVL]) reaches 0x0 and subsequently reloads to LDVAL0[TSV], then timer 1 CVAL1[TVL] should decrement by 0x1.

However this decrement does not occur until one cycle later, therefore a read of the PIT upper lifier timer register (LTMR64H) is followed by a read of the PIT lower lifier timer register (LTMR64L) at the instant when timer 0 has reloaded to LDVAL0[TSV] and timer 1 is yet to be decremented in next cycle then an incorrect timer value in LTMR64H[LTH] is expected.

**Workaround:** In lifier mode if the read value of LTMR64L[LTL] is equal to LDVAL0[TSV] then read both LTMR64H and LTMR64L registers one additional time to obtain the correct lifetime value.

### **ERR011321: PIT\_RTI: Generates false RTI interrupt on re-enabling**

**Description:** A false Real-Time Interrupt (RTI) may be observed when the RTI module is re-enabled if, after servicing an RTI interrupt (by clearing TFLGn[TIF]), the clocks to the RTI module are disabled.

This occurs only if the RTI module clock is disabled within four RTI clock cycles of an RTI Interrupt being cleared.

**Workaround:** Option 1: The user should check the RTI interrupt flag, TFLGn[TIF] before servicing the interrupt, this flag won't be set for the false/spurious interrupts.

Option 2: Ensure that the module clock to the RTI module is not disabled within four RTI clock cycles after servicing an RTI interrupt. Consult the chip-specific documentation to determine the clock period of the RTI module and implement a time delay of at least five times this period before disabling the RTI module clock.

### **ERR010590: PLL: Might remain unlocked when enabled after Standby or power on**

**Description:** When enabled after an event where the following conditions are met, the PLL might fail to lock:

- 1.- VDD\_LV (1.2 V) was disabled
- 2.- A Pre-divider of 1 (PLLDIG\_PLLDV[PREDIV] = 0 or 1) is used
- 3.- The PLL is clocked from the fast internal oscillator (FXOSC)

Common situations in which VDD\_LV might be disabled are Standby and power-off of the device.

**Workaround:** There are three ways to avoid this:

- 1.- Avoid using a pre-divider of 1 (PLLDIG\_PLLDV[PREDIV] = 0 or 1) and instead use a value of 2.
- 2.- During Standby leave the supply to VDD\_LV enabled.
- 3.- Initialize the PLL using the fast internal oscillator (FIRC) as source clock and then switch to use the fast external oscillator (FXOSC).

### **ERR050467: PLL: Possible loss of lock when using the PLL bypass calibration mode**

**Description:** A momentary PLL loss of lock may occur shortly after enabling the PLL if using the PLL bypass calibration mode (PLLDIG\_PLLCAL1[BYPICAL] = 1). Note that the PLLDIG\_PLLCAL1[BYPICAL] bit is set by default after exiting the low power STANDBY mode.

**Workaround:** Do not use the PLL bypass calibration mode and instead ensure PLLDIG\_PLLCAL1[BYPICAL] is cleared before enabling the PLL.

### **ERR011096: SAI: Internal bit clock is not generated when RCR2[BCI]=1 or TCR2[BCI]=1**

**Description:** When the SAI transmitter or receiver is configured for internal bit clock with BCI = 1, the bit clock is not generated for either of the following two configurations:

- a) SYNC = 00 and BCS = 0
- b) SYNC = 01 and BCS = 1

**Workaround:** When the SAI transmitter or receiver is configured for internal bit clock with BCI=1, use only one of the following two configurations:

a) SYNC = 01 and BCS = 0

b) SYNC = 00 and BCS = 1

**ERR011150: SAI: Internally generated receive or transmit BCLK cannot be re-enabled if it is first disabled when RCR2[DIV] or TCR2[DIV] > 0**

**Description:** If the receive or transmit bit clock (BCLK) is internally generated, enabled with DIV > 0 and is then disabled, due to software or Stop mode entry, and the BCLK is enabled again, the clock is not generated.

**Workaround:** If the receive or transmit BCLK is internally generated and a DIV value greater than 0 is used, the SAI must be reset before the BCLK is re-enabled. This is achieved by writing the SR bit in the respective RCSR or TCSR register first to 1 and then immediately to 0.

**ERR050144: SAI: Setting FCONT=1 when TMR>0 may not function correctly**

**Description:** When FCONT=1 the transmitter will recover after a FIFO error when the FIFO is no longer empty and starting again from the same word in the following frame where the error occurred.

Configuring TMR > 0 will configure one or more words in the frame to be masked (nothing transmitted during that slot). If anything other than the last word(s) in the frame are masked when FCONT=1 and a FIFO Error Flag is set, then the transmitter will not recover and will set FIFO Error Flag during each frame.

**Workaround:** To avoid this issue, set FCONT in TCR4 to be 0.

**ERR011306: SAR ADC: Incorrect value of ADC power down exit delay evaluated by the formula given in PDEDR [PDED] field description**

**Description:** The formula in the register field ADC\_PDEDR [PDED] provides the delay between the power down bit reset and start of conversion value in number of clock cycles of the ADC module clock, however the given formula of PDED x 1/[ADC\_clock\_frequency] is incorrect. This gives a calculated value that is too short by 1 cycle of ADC Bus clock and 1 cycle of ADC clock (AD\_clk).

**Workaround:** The correct formula that should be used to calculate the value for the ADC\_PDEDR[PDED] register is -

$$(1/\text{ADC Bus clock}) + ((\text{PDED}+1) \times 1/[\text{ADC\_clock\_frequency}])$$

Where:

ADC\_clock\_frequency = Frequency of ADC clock (AD\_clk)

ADC Bus clock= Module interface clock for register access (ADC\_CLK)

## **ERR050572: SIRC: Clock output may contain extra clock pulses**

**Description:** The 128 kHz Slow Internal RC (SIRC) oscillator may generate an extra clock pulse per clock period. The occurrence of the potential extra clock pulse may vary for each clock period ranging from no extra clock pulse to one extra clock pulse per period. Factors that affect the occurrence rate are part to part variations, temperature, and core voltage.

The SIRC output clock may be selected as the clock source for the Real Time Clock (RTC) via the RTC\_RTCC[CLKSEL] register, as a clock to monitor in the Clock Monitor Unit (CMU) via the CMU\_CSR[CLKSEL1] register, or as the clock source for the Software Watchdog Timer (SWT) timers (SWTx, HSM\_SWT). The RTC, CMU, and SWT counters/timers may get an extra clock per SIRC clock period causing a higher than expected count (which may affect the RTC count/wakeup duration, the CMU measured SIRC frequency, or the SWT time-out period) or may cause a corrupted count value. The SIRC clock may also be selected to the CLKOUT\_0 and CLKOUT\_1 pins in which the extra pulse may or may not be observable for a divide by 1 configuration and for non-divide by 1 configurations may affect the divided clock duty cycle or may corrupt or stall the divided clock waveform.

**Workaround:** Select other clock sources for the RTC. The RTC supports other clock sources via the RTC\_RTCC[CLKSEL] register which include selections for the 32 kHz Slow External Crystal Oscillator (SXOSC), the 16 MHz Fast Internal RC Oscillator (FIRC), or the 8-40 MHz Fast External Crystal Oscillator (FXOSC). If the RTC uses the SIRC clock, then the application needs to manage possible unexpected counter values such as count values that are double the expected value.

If the CMU monitors the SIRC clock, then the application needs to manage possible unexpected measured SIRC frequencies based on possible unexpected counter values such as frequencies that are double the expected value.

The application needs to account for the possibility of a quicker SWT timeout for any SWT that is enabled. Other timer sources such as a PIT timer may be used to compliment the SWT counts.

If the CLKOUTx pin selects the SIRC clock source, then the application needs to manage the possible unexpected CLKOUT waveforms.

## **ERR050196: STANDBY EXIT: Device may not come out of reset if a short functional reset comes immediately after wakeup event.**

**Description:** The device may get stuck in STANDBY exit sequence if a short functional reset is observed between wakeup event and low power mode exit.

A destructive or POR reset after short functional reset will recover the device out of the stuck condition.

**Workaround:** Do not configure any functional reset source in MC\_RGM\_FESS to generate short functional reset before entering low power mode.

## **ERR010815: STCU2: Offline LBIST execution will toggle GPIO[29] ,GPIO[30], GPIO[62] and GPIO[63]**

**Description:** When the Self Test Control Unit (STCU2) is configured to execute Offline Logic Built-in Self Test (LBIST) GPIO[29] ,GPIO[30], GPIO[62] and GPIO[63] will toggle during the LBIST execution phase. All other GPIO are static during LBIST execution. Note MPC5748G 0N78S is factory programmed with STCU2 DCF records to execute offline MBIST (Memory Built-in Self Test) and LBIST by default. During MBIST execution all GPIO are static.

**Workaround:** The user should select the most appropriate workaround for offline LBIST:

1. Disable offline MBIST and LBIST. The STCU2\_CFG DCF can be programmed with the NIL pointer. The user should program the following DCF {0x7F000000, 0x0008000C} = (STCU2\_CFG data, STCU2\_CFG DCF client address).
2. Disable offline LBIST only, MBIST will continue to execute. The STCU2\_MB\_CTRL75 can be programmed with the NIL pointer. The user should program the following DCF {0x7F000000, 0x0008072C} = (STCU2\_MB\_CTRL75 data, STCU2\_MB\_CTRL75 DCF client address).
3. The user can assess the impact of the GPIO[29] ,GPIO[30], GPIO[62] and GPIO[63] toggles to understand whether it will affect the application. If no impact the LBIST can execute.

Note: User DCF records should be programmed to the first erased memory location in the UTEST flash region beyond 0x00400\_0300.

Note: Option (1) and (2) should not be used in safety-critical applications.

## **ERR010721: STCU: If the Auxiliary Clock 9 Select Control Register CGM\_AC9\_SC[SELCTL] = 1, offline-BIST will fail leading to a STCU watchdog timeout**

**Description:** When STCU offline-BIST is enabled (default) and the Auxiliary Clock 9 Select Control Register CGM\_AC9\_SC[SELCTL] = 1 (default = 0) the offline-BIST will not complete leading to an STCU watchdog timeout. STCU watchdog timeout duration depends on STCU\_WDG DCF record programmed before the STCU\_RUN DCF record.

**Workaround:** To avoid the STCU offline-BIST failure leading to the STCU watchdog timeout the user must select 1 of the following workarounds:

1. If there is a requirement for CGM\_AC9\_SC[SELCTL] = 1 (FXOSC) the user can avoid STCU offline-BIST failure by disabling FlexCAN\_0 MBIST by programming STCU\_MB\_CTRL40 DCF (0xBA00000000806A0). This will reduce BIST coverage as FlexCAN\_0 MBIST is excluded.
2. If STCU offline-BIST is required to be executed without a reduction in coverage the user must select CGM\_AC9\_SC[SELCTL] = 0 (FS80).
3. If there is a requirement for CGM\_AC9\_SC[SELCTL] = 1 (FXOSC) the user can disable STCU offline-BIST by programming STCU\_CFG DCF (0x7F000000008000C).

## **ERR010200: STM: Reading the System Timer Module Count register may return an incorrect value**

**Description:** The erratum may only occur when the STM is configured to use the FXOSC clock source selected at STM\_CR[CSL]. For the case application software reads the STM Count register (STM\_CNT) the value returned may be incorrect. However, the user should be assured that STM interrupts will continue to be triggered at the expected STM\_CNT value.

Note the default clock source for the STM is the FS80 (divided system clock) and this configuration is not impacted by this erratum.

**Workaround:** To avoid the erratum condition the user should select the FS80 clock source for the STM. However, for the case the FXOSC is required to clock the STM and the STM Count register is to be read, the following sequence must be executed:

1. Disable the STM via STM\_CR[TEN]
2. Read STM Counter register STM\_CNT
3. Re-enable the STM via STM\_CR[TEN]

## **ERR006308: USB: Host non-doubleword –aligned buffer address can cause host to hang on OUT Retry**

**Description:** The USB host core operating in streaming mode may underrun while sending the data packet of an OUT transaction. The host then retries the OUT transaction according to the USB specification.

This issue occurs during the OUT retry. The USB host may hang on OUT retry if the data buffer start address is not 4-byte aligned. This applies to both the host controller and the OTG controller in host mode.

- Workaround:**
- Set the host TXFIFO threshold to a large value (TXFIFOTHRES in the TXFILLTUNING register). This increases the tolerance to bus latency and avoids a FIFO underrun.
  - Set the Stream Disable bit (SDIS) to 1 in the USBMODE register. This forces the controller to load an entire packet in the FIFO before starting to transmit on the USB bus. Hence, the FIFO never underruns. This somewhat reduces the maximum bandwidth of the USB, because there is idle time when the the controller waits for the entire packet to be loaded.

**How to Reach Us:****Home Page:**[nxp.com](http://nxp.com)**Web Support:**[nxp.com/support](http://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](http://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,  $\mu$ Vision, 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.

© 2022 NXP B.V.

