

## Mask Set Errata for Mask 2N58R

This report applies to mask 2N58R for these products:

- S32R274

### Mask Specific Information

JTAG identifier	0x298D801D
-----------------	------------

**Table 1. Errata and Information Summary**

Erratum ID	Erratum Title
<a href="#">ERR007274</a>	LINFlexD: Consecutive headers received by LIN Slave triggers the LIN FSM to an unexpected state
<a href="#">ERR007433</a>	JTAGM: Nexus error bit is cleared by successful RWA
<a href="#">ERR007791</a>	SIUL2: Transfer error not generated if reserved addresses within the range of SIUL BASE + 0x100 to 0x23F are accessed
<a href="#">ERR007869</a>	FCCU: FOSU monitoring of a fault is blocked for second or later occurrence of the same fault
<a href="#">ERR007905</a>	PIT: Accessing the PIT by peripheral interface may fail immediately after enabling the PIT peripheral clock
<a href="#">ERR007991</a>	FLASH: Rapid Program or Erase Suspend fail status
<a href="#">ERR008004</a>	FLASH: Array Integrity with Breakpoints enabled may skip addresses for certain RWSC and APC combinations
<a href="#">ERR008340</a>	NPC: EVTO_B toggles instead of remaining asserted when used by the DTS if Nexus is not enabled
<a href="#">ERR008770</a>	FlexRAY: Missing TX frames on Channel B when in dual channel mode and Channel A is disabled
<a href="#">ERR008935</a>	JTAGM: write accesses to registers must be 32-bit wide
<a href="#">ERR009589</a>	e200z420: The Slave Port (back door) of e200z420 DTCM (Data side Tightly Coupled Memory) is not functional in 4:4:2:1 mode with an e200z420 core clock above 120 Mhz.
<a href="#">ERR009590</a>	End to End ECC: An uncorrectable ECC event from a slave to a master can cause invalid data reception
<a href="#">ERR009658</a>	SPI: Inconsistent loading of shift register data into the receive FIFO following an overflow event
<a href="#">ERR009928</a>	FlexPWM: Half cycle automatic fault clearing does not work in PWM submodule 0 under some conditions
<a href="#">ERR010085</a>	SPT: FIR operation writes unnecessary extra data into OPRAM when asynchronous PDMA is concurrently accessing the same OPRAM bank
<a href="#">ERR010086</a>	SPT: Incorrect global maxima result from MAXS command when an asynchronous PDMA is concurrently accessing the same OPRAM bank

*Table continues on the next page...*



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

Erratum ID	Erratum Title
<a href="#">ERR010088</a>	STCU2: Unexpected STCU self-test timeout can occur when a short sequence for external reset is triggered during execution of online self-test
<a href="#">ERR010199</a>	AFE: Device may not exit the reset sequence if a low voltage detection or reset event occurs with Sigma Delta PLL selected as system clock
<a href="#">ERR010259</a>	PMC: Device may not exit the startup reset sequence under certain power sequencing conditions until the STCU watchdog time-out interval elapses
<a href="#">ERR010287</a>	CGM: Under certain conditions the data or instruction crossbars may hang during a change of system clock dividers using the Divider Update Trigger Register
<a href="#">ERR010326</a>	SWT: Out of lockstep fault for instruction and data cache memories is incorrectly indicated when watchdog is configured in fixed address execution service mode
<a href="#">ERR010327</a>	ADC: Incorrect channel under measure is indicated after sampling phase of conversion until conversion completes
<a href="#">ERR010328</a>	JTAGM: Software Reset bit does not auto clear when written with 1 when an external tool is not connected
<a href="#">ERR010329</a>	JTAGM: JTAGM Data out registers are writable regardless of state of the Data Transfer Mode bit
<a href="#">ERR010330</a>	JTAGM: An unexpected interrupt will occur if the Idle Interrupt is enabled after the end of frame transfer
<a href="#">ERR010331</a>	NAL: Trace connections to the device are lost if a device reset occurs
<a href="#">ERR010337</a>	AFE: The AFE_PLLSTS[LOR], AFE_OSCSTS[STS], and MC_ME_GS[S_XOSC] bits do not indicate loss of reference status
<a href="#">ERR010340</a>	NZxC3: ICNT and HIST fields of a Nexus message are not properly reset following a device reset
<a href="#">ERR010345</a>	PMC: The Power Management Controller (PMC) module Analog Front End (AFE) Low Voltage Detect (LVD) interrupt may not work as a valid source of interrupt
<a href="#">ERR010352</a>	SPT: AHB error status can be asserted for the incorrect SPT DMA when performing back to back AHB accesses
<a href="#">ERR010357</a>	PASS: JTAG password match bypasses flash read protection set by PASS.LOCK3[RLx] bits
<a href="#">ERR010365</a>	MC_ME: An invalid mode (Clock Usage) configuration error can occur if the Sigma-Delta PLL and PLL1 are switched on or off independently
<a href="#">ERR010385</a>	e200z4: Incorrect branch displacement at 16K memory boundaries
<a href="#">ERR010391</a>	LFAST: Receiver pads do not meet input hysteresis specification
<a href="#">ERR010393</a>	CGM: CLK_OUT0 and CLK_OUT1 dividers may become stuck if clock selection is changed while dividers with divide by 2 are operational.
<a href="#">ERR010414</a>	PMC: Low Voltage Detect (LVD) self test may incorrectly indicate a self test fail if the supply is outwith its operating range during power up
<a href="#">ERR010436</a>	ZipWire: SIPI can have only one initiator with one outstanding write frame at time
<a href="#">ERR010452</a>	eDMA: When master ID replication is enabled, the stored ID and privilege level will change if read by another master.
<a href="#">ERR010455</a>	ADC: Calibration routine needs to be repeated after any SoC reset that triggers device built-in self-test
<a href="#">ERR010458</a>	RGMI: Ethernet clock output pad 107 can glitch during clock transitions of ENET operation
<a href="#">ERR010459</a>	SPT: Spurious CS_AHB_ERR can be generated in the SPT_DMA_ERR_STATUS register
<a href="#">ERR010479</a>	NPC: Nexus enable required for mode changes when a debugger is attached
<a href="#">ERR010486</a>	PLLDIG: Device may not exit the reset sequence if a reset occurs whilst PLL1 is running with Modulation enabled.

*Table continues on the next page...*

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

Erratum ID	Erratum Title
<a href="#">ERR010490</a>	STCU2: Register values STCU2_PLL_CFG[PLLLDF, PLLODF], STCU2_LBUFM and STCU2_LBRMSW can change at end of self test
<a href="#">ERR010537</a>	AFE: Device may not exit the reset sequence if a low voltage detection on VDD_HV_RAW or VDD_HV_DAC supplies occurs when XOSC is the source of PLL0/1 clock.
<a href="#">ERR010638</a>	NZxC3: Nexus messaging becomes corrupted if more than one master (including a device core) is active and the core is subsequently disabled.
<a href="#">ERR010642</a>	AFE: XOSC overvoltage stress and AFE bandgap overvoltage stress when LBIST partition number 6 is executed during self test
<a href="#">ERR010682</a>	STCU2: Online LBIST of partition 6 incorrectly generates reset when run with AFE LVD unmasked
<a href="#">ERR010750</a>	PMC: Device may not exit the reset sequence if a low voltage detect event occurs during execution of offline or online self-test
<a href="#">ERR010879</a>	SPT: Adaptive scaling using count unoccupied bits from bit 23 down mode can introduce harmonics in the FFT output
<a href="#">ERR010880</a>	SWT: SWT may not get serviced in Fixed and Incremental address mode
<a href="#">ERR010900</a>	FCCU: False indication of a fault state for a single safe clock period can be generated on the error output pin
<a href="#">ERR010997</a>	SPT: Work register contents not deterministic after execution of STOP command
<a href="#">ERR011032</a>	PMC: Unexpected events when an LVD occurs on a supply with its LVD masked
<a href="#">ERR011156</a>	PMC: During online PMC self-test if RESET_B is asserted then a spurious destructive or power on reset may occur and be reported as an LVD/HVD event in the RGM_DES register.
<a href="#">ERR011172</a>	MIPICSI2: Start of Transmission Error can occur at lower speeds
<a href="#">ERR011188</a>	FCCU : Fault NCF[106] may occur when PLL Loss of Lock event occurs during MBIST execution when online STCU-selftest is run.
<a href="#">ERR011245</a>	AFE: AFE_OSCSTS[STS] bit does not get cleared in single-ended and differential bypass mode of XOSC
<a href="#">ERR011297</a>	PMC: Default value of self-test time window (STTW) changed
<a href="#">ERR011306</a>	SAR ADC: Incorrect value of ADC power down exit delay evaluated by the formula given in PDED [PDED] field description
<a href="#">ERR011329</a>	SPT: After completion of a command sequence there is a time window in which work register access by CPU can cause the system to become non-responsive
<a href="#">ERR011394</a>	CTE: Timing table is not executed when the CTE time base clock is set to 80 MHz and the table execution duration is 1 clock cycle
<a href="#">ERR011407</a>	CMU: Sudden loss of clock does not signal the Fault Collection and Control Unit
<a href="#">ERR050090</a>	DSPI/SPI: Incorrect data may be transmitted in slave mode
<a href="#">ERR050119</a>	FlexRay: Disabling of FlexRay Message Buffer during the STARTUP Protocol State takes longer than expected three Slots
<a href="#">ERR050130</a>	PIT: Temporary incorrect value reported in LMTR64H register in lifetimer mode
<a href="#">ERR050246</a>	FlexCAN: Receive Message Buffers may have its Code Field corrupted if the Receive FIFO function is used

**Table 2. Revision History**

Revision	Changes	Date
3	<p>The following errata were added.</p> <ul style="list-style-type: none"> <li>• ERR050130</li> <li>• ERR050246</li> <li>• ERR011329</li> <li>• ERR050090</li> <li>• ERR011394</li> <li>• ERR050119</li> </ul> <p>The following errata were revised.</p> <ul style="list-style-type: none"> <li>• ERR011407</li> <li>• ERR010479</li> <li>• ERR007869</li> </ul>	Feb 2020
2	<p>The following errata were added.</p> <ul style="list-style-type: none"> <li>• ERR011245</li> <li>• ERR011407</li> <li>• ERR011188</li> <li>• ERR010900</li> <li>• ERR011172</li> <li>• ERR011156</li> <li>• ERR011032</li> <li>• ERR010879</li> <li>• ERR010997</li> <li>• ERR010880</li> <li>• ERR011297</li> <li>• ERR011306</li> </ul> <p>The following errata were updated.</p> <ul style="list-style-type: none"> <li>• ERR010436</li> </ul>	Apr 2018
1	Initial revision	Feb 2017

**ERR007274: LINFlexD: Consecutive headers received by LIN Slave triggers the LIN FSM to an unexpected state**

**Description:** As per the Local Interconnect Network (LIN) specification, the processing of one frame should be aborted by the detection of a new header sequence and the LIN Finite State Machine (FSM) should move to the protected identifier (PID) state. In the PID state, the LIN FSM waits for the detection of an eight bit frame identifier value.

In LINFlexD, if the LIN Slave receives a new header instead of data response corresponding to a previous header received, it triggers a framing error during the new header's reception and returns to IDLE state.

**Workaround:** The following three steps should be followed -

- 1) Configure slave to Set the MODE bit in the LIN Time-Out Control Status Register (LINTCSR[MODE]) to '0'.

2) Configure slave to Set Idle on Timeout in the LINTCSR[IOT] register to '1'. This causes the LIN Slave to go to an IDLE state before the next header arrives, which will be accepted without any framing error.

3) Configure master to wait for Frame maximum time (T Frame\_Maximum as per LIN specifications) before sending the next header.

Note:

$$T_{Header\_Nominal} = 34 * T_{Bit}$$
$$T_{Response\_Nominal} = 10 * (N_{Data} + 1) * T_{Bit}$$
$$T_{Header\_Maximum} = 1.4 * T_{Header\_Nominal}$$
$$T_{Response\_Maximum} = 1.4 * T_{Response\_Nominal}$$
$$T_{Frame\_Maximum} = T_{Header\_Maximum} + T_{Response\_Maximum}$$

where TBit is the nominal time required to transmit a bit and NData is number of bits sent.

### **ERR007433: JTAGM: Nexus error bit is cleared by successful RWA**

**Description:** The JTAG Master module status register includes a Nexus error status bit (JTAGM\_SR[Nexus\_err]) that indicates the status of the last Nexus Read/Write Access (RWA) command. Once this information is latched, it can only be cleared by performing a successful RWA transaction via the same core that caused the error. In addition, if a RWA transaction is performed by a different core, the error bit will not be cleared and it is not possible to determine if the access by the second core RWA was successful or generated another error.

In general, this bit should only be set when the Nexus RWA accesses non-existent or protected memory spaces.

**Workaround:** If the status information is required from a specific core, the user software or tool should read the error bit (ERR) of the e200zx core's Nexus Read/Write Access Control/Status register. To avoid setting the error bit, do not perform illegal memory accesses.

### **ERR007791: SIUL2: Transfer error not generated if reserved addresses within the range of SIUL BASE + 0x100 to 0x23F are accessed**

**Description:** If any reserved register within the System Integration Unit Lite 2 (SIUL2) register range from SIUL2 BASE + 0x100 to 0x23F is accessed then no transfer error will occur.

**Workaround:** Software should not be dependent on the indication of a transfer error occurring from an access within the SIUL2 register range from SIUL2 BASE + 0x100 to 0x23F.

### **ERR007869: FCCU: FOSU monitoring of a fault is blocked for second or later occurrence of the same fault**

**Description:** The Fault Collection and Control Unit (FCCU) Output Supervision Unit (FOSU) will not monitor the FCCU for the second or later occurrence of a given fault in the following cases:

1. Reset is programmed as the only reaction for the fault.
2. Assertion of the fault coincides with the long/short functional reset reaction to a fault previously asserted.

**Workaround:** Enable either Alarm state (NCFTOEx) or at least one other type of Fault-state reaction: Non-maskable Interrupt (NMI) or error out (EOUT) signaling reaction for the faults that have a reset reaction enabled only. Restrictions of combining reset reaction with additional reactions may be written in the chip specific sub-section of the FCCU chapter.

#### **ERR007905: PIT: Accessing the PIT by peripheral interface may fail immediately after enabling the PIT peripheral clock**

**Description:** If a write to the Periodic Interrupt Timer (PIT) module enable bit (PIT\_MCR[MDIS]) occurs within two bus clock cycles of enabling the PIT clock gate in the MC\_CGM (Clock Generation Module) register, the write will be ignored and the PIT will not be enabled.

**Workaround:** After enabling the PIT clock in the MC\_CGM, insert a read of the PIT\_MCR register before writing to the PIT\_MCR register. This guarantees a minimum delay of two bus clocks to guarantee the write is not ignored.

#### **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.

#### **ERR008004: FLASH: Array Integrity with Breakpoints enabled may skip addresses for certain RWSC and APC combinations**

**Description:** For certain combinations of the Flash Read Wait State Control (RWSC) and Address Pipeline Control (APC) settings in the Platform Flash Configuration Register (PFLASH\_PCFR1) the Flash's array integrity (AI) check when run with breakpoints enabled may skip addresses resulting in an incorrect Multiple Input Signature Register (MISR) value or in the case of back to back ECC event errors (EER) or Single Bit Correction (SBC) events, a skipped breakpoint. This occurs for the following combinations:

RWSC=1 and APC=1

RWSC=3 and APC=2

RWSC=5 and APC=3

If breakpoints are enabled and an EER or SBC cause a breakpoint to occur the address after the breakpoint will be skipped, and the resulting MISR will not match expectations. Likewise, if there are back to back errors (EER or SBC) during AI with the above RWSC/APC combinations the 2nd error (and breakpoint) will be missed.

Margin Read (which by specification is a self timed event and is independent of wait states selected) is not affected by this erratum. This erratum only applies to Array Integrity.

**Workaround:** One workaround is to follow the recommended RWSC and APC combinations for given frequencies. If this is done, Array Integrity with Breakpoints feature works as expected. Valid RWSC/APC combinations listed in the specification are:

Flash Operating Frequency	RWSC	APC
30 MHz	0	0
100 MHz	2	1
133 MHz	3	1
167 MHz	4	1
200 MHz	5	2

A second workaround is if the above RWSC and APC combinations (listed in the description) are desired to be checked, do so without enabling breakpoints. In this case, the first EER or SBC event will be logged, and the MISR will correctly reflect the result of all reads being executed.

**ERR008340: NPC: EVTO\_B toggles instead of remaining asserted when used by the DTS if Nexus is not enabled**

**Description:** When the Development Trigger Semaphore (DTS) module asserts its trigger output on the Event Out (EVTO\_B) pin, the EVTO\_B pin will toggle instead of remaining asserted low if the Nexus Port Controller (NPC) Port Configuration Register MCKO Enable (NPC PCR[MCKO\_EN]) bit is set to 0. If the NPC PCR[MCKO\_EN] bit is set to 1, the EVTO\_B pin behaves as expected and remains asserted low as long as the DTS asserts its trigger output.

**Workaround:** Always set the MCKO\_EN bit to 1 when using the DTS trigger out function.

**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.

#### **ERR008935: JTAGM: write accesses to registers must be 32-bit wide**

**Description:** The JTAG Master module (JTAGM) supports only 32-bit write accesses to its registers. A byte write access will be converted into a 32-bit write with the other bytes values at 0x0.

**Workaround:** Perform only 32-bit write accesses on JTAGM registers. Do not use byte writes.

#### **ERR009589: e200z420: The Slave Port (back door) of e200z420 DTCM (Data side Tightly Coupled Memory) is not functional in 4:4:2:1 mode with an e200z420 core clock above 120 Mhz.**

**Description:** Data write/reads to e200z420 DTCM from backdoor from e200z420 data, e200z7 (Core\_1) data, e200z7 (Core\_2) data, SPT, SIPI, Ethernet and DMA are not guaranteed to work when device is configured in 4:4:2:1 mode when e200z420 core clock (CORE0\_CLK) is above 120 Mhz. This could result in failed access attempts to the e200z420 DTCM and/or corruption in accesses.

**Workaround:** User should not attempt to access e200z420 DTCM via backdoor accesses from e200z420 data, e200z7 (Core\_1) data, e200z7 (Core\_2) data, SPT, SIPI, Ethernet and DMA if the device is configured in 4:4:2:1 mode with an e200z420 core clock (CORE0\_CLK) of greater than 120MHz.

#### **ERR009590: End to End ECC: An uncorrectable ECC event from a slave to a master can cause invalid data reception**

**Description:** When an Uncorrectable Error Correction Code (ECC) is received by a master, incorrect data can result in the transaction after the ECC error. There are three scenarios impacting the device:

Case 1 Master DMA:

If an uncorrectable End to End ECC transaction is received during back to back requests from DMA to a slave in the cross bar (RAM/Flash) then the DMA can potentially receive incorrect data.

Case 2 Master Ethernet

If an uncorrectable End to End ECC transaction is received during back to back requests from Ethernet to a slave in the cross bar (RAM/Flash) then the Ethernet can potentially receive incorrect data

Case 3 Signal Processing Toolbox (SPT):

If an uncorrectable End to End ECC transaction is received during back to back requests from SPT to a slave in the cross bar (RAM/Flash) then the SPT can potentially receive incorrect data

**Workaround:** Use one of the two following workarounds:

1. Use the NCF for Uncorrectable ECC Errors on all the slaves to determine that such a fault has occurred and take appropriate action.

NCF[17] - System RAM uncorrectable ECC error, and

NCF[23] - Flash (c55fmc) uncorrectable ECC error can be monitored to determine if this scenario occurred

2. Clear Pending Read Enable bit for the masters

Signal Processing Toolbox – PCM\_IAHB\_BE5[PRE\_SPT] - Set to 0

Ethernet- PCM\_IAHB\_BE2[PRE\_ENET] - Set to 0

DMA – PCM\_IAHB\_BE1[PRE\_DMA]= Set to 0

However this configuration may reduce the performance timing of the system.

### **ERR009658: SPI: Inconsistent loading of shift register data into the receive FIFO following an overflow event**

**Description:** In the Serial Peripheral Interface (SPI) module, when both the receive FIFO and shift register are full (Receive FIFO Overflow Flag bit in Status Register is set (SR [RFOF] = 0b1)) and then the Clear Rx FIFO bit in Module Configuration Register (MCR [CLR\_RXF]) is asserted to clear the receive FIFO, shift register data is sometimes loaded into the receive FIFO after the clear operation completes.

**Workaround:** 1. Avoid a receive FIFO overflow condition (SR[RFOF] should never be 0b1). To do this, monitor the RX FIFO Counter field of the Status Register (SR[RXCTR]) which indicates the number of entries in receive FIFO and clear before the counter equals the FIFO depth.

2. Alternatively, after every receive FIFO clear operation (MCR[CLR\_RXF] = 0b1) following a receive FIFO overflow (SR[RFOF] = 0b1) scenario, perform a single read from receive FIFO and discard the read data.

### **ERR009928: FlexPWM: Half cycle automatic fault clearing does not work in PWM submodule 0 under some conditions**

**Description:** When

a) the EXT\_SYNC signal is selected to cause initialization by setting the Submodule 0 Control 2 Register FlexPWM\_SUB0\_CTRL2[INIT\_SEL] = 11 and

b) a specific FAULTx input is associated with the submodule 0 outputs using the Submodule 0 Fault Disable Mapping Register (FlexPWM\_SUB0\_DISMAP) and

c) the respective bit for that FAULTx is 0 in the FFULL bitfield of the Fault Status Register FlexPWM\_FSTS and

d) the respective bit for that FAULTx is 1 in the FAUTO bitfield of the Fault Control Register FlexPWM\_FCTRL,

then the PWM outputs of submodule 0 will only be re-enabled at the cycle boundary (full cycle) and will not be re-enabled at the cycle midpoint (half cycle).

**Workaround:** When the EXT\_SYNC signal is used to cause initialization in submodule 0 and the submodule 0 PWM outputs are disabled by a specific FAULTx input, use full cycle automatic fault clearing for the specific FAULTx input by setting the corresponding bit of the Fault Status Register FlexPWM\_FSTS[FFULL] to 1.

**ERR010085: SPT: FIR operation writes unnecessary extra data into OPRAM when asynchronous PDMA is concurrently accessing the same OPRAM bank**

**Description:** The FIR command will write extra values into the next OPRAM locations (the next OPRAM column address) following the last expected write location when there is an asynchronous PDMA operation concurrently accessing the same OPRAM bank.

**Workaround:** When using the FIR command with simultaneous asynchronous PDMA, ensure that the commands are operating on different OPRAM banks.

**ERR010086: SPT: Incorrect global maxima result from MAXS command when an asynchronous PDMA is concurrently accessing the same OPRAM bank**

**Description:** The MAXS command can produce an incorrect result for global maxima when an asynchronous PDMA operation is concurrently accessing the same OPRAM bank as the MAXS operation. This occurs only when the input to the MAXS command is complex data with the preprocessing set to calculate magnitude of the input (command bits [119:118]PREPROC = 0b10).

**Workaround:** Use different OPRAM banks for MAXS and asynchronous PDMA if they are to operate concurrently.

**ERR010088: STCU2: Unexpected STCU self-test timeout can occur when a short sequence for external reset is triggered during execution of online self-test**

**Description:** While an online self-test is in progress there is a finite window during the self-test execution during which if an external reset is asserted (RESET\_B pulled low) and this reset is configured as short sequence for external reset by setting the Short Sequence for External Reset bit in the Reset Generation Module Functional Event Short Sequence Register (RGM\_FESS[SS\_EXR] = 1b1), or if another functional reset source is triggered during this window, the time after which the part waits for self-test to complete is longer than expected. This time-out value is governed by the watchdog time-out value set in the Watchdog End of Count Timer field in the STCU2 Watchdog Register Granularity register (STCU2\_WDG[WDGEOC]). Further, the self-test does not issue a hardware abort (STCU2 Error Register On-line Hardware Abort Flag (STCU2\_ERR\_STAT [ABORTHW]) will not be set to 1b1) but signals a time-out (STCU2\_ERR\_STAT [WDTOSW] = 1b1). If the online self-test is being run with PLL enabled then an unexpected PLL unlock event is also observed (STCU2\_ERR\_STAT[LOCKESW] = 1b1)

**Workaround:** To avoid the longer than expected duration for self-test completion, allow the online self-test to complete without applying external reset when the external reset is configured as a short sequence for external reset. The other functional resets must also not be triggered during the online self-test execution.

**ERR010199: AFE: Device may not exit the reset sequence if a low voltage detection or reset event occurs with Sigma Delta PLL selected as system clock**

**Description:** If the device is configured with the 160 MHz clock from Sigma Delta PLL (SDPLL\_CLK160) selected as the source of system clock (MC\_CGM\_SC\_SS[SELSTAT] = 0b0011) there is a chance that the device may not exit the reset sequence if any of the following events occur:

- Low voltage detection on any supply
- Long external reset (causes entry to the reset sequence at PHASE1)
- Phase 3 destructive reset (causes entry to the reset sequence at PHASE3[DEST])
- Phase 3 functional reset (causes entry to the reset sequence at PHASE3[FUNC])

**Workaround:** There are two possible workarounds that can be applied:

1. Use an external watchdog circuit and power supply monitor to determine if any low voltage event on an external power supply has occurred and if the MCU has become non responsive. The external circuitry should be configured to assert the device power-on reset signal (VREG\_POR\_B) while the power supplies are outside of the specified operational ranges. This will ensure that the device can safely return to normal operation once the supplies are back within the specified ranges. This is as per the recommendations of the device safety manual.
2. Do not select SDPLL\_CLK160 as the source of system clock.

**ERR010259: PMC: Device may not exit the startup reset sequence under certain power sequencing conditions until the STCU watchdog time-out interval elapses**

**Description:** During startup, if the VDD\_HV\_PMU and VDD\_HV\_IO supplies trigger their respective low-voltage detects (LVDs) without the power-on reset signal (VREG\_POR\_B) asserted, the device may not exit the reset sequence until the Self Test Control Unit (STCU) watchdog timer (WDG) interval elapses and performs a device reset. This time is factory programmed to 1.5 seconds but can be user defined by programming the STCU\_WDG DCF record in UTEST flash. The watchdog timer counter will elapse and bring the part out of reset regardless of whether the STCU is programmed to run offline self-test.

**Workaround:** To prevent the issue from occurring, VREG\_POR\_B should be asserted anytime the VDD\_HV\_IO and VDD\_HV\_PMU supplies are below the operating range stated in the device data sheet (this is already a requirement for external regulation mode).

**ERR010287: CGM: Under certain conditions the data or instruction crossbars may hang during a change of system clock dividers using the Divider Update Trigger Register**

**Description:** When the Divider Update Trigger Register (MC\_CGM\_DIV\_UPD\_TRIG) is used to enable the pre-loaded system clock configurations in the Clock Generation Divider Registers (MC\_CGM\_SC\_DCx) when the core is accessing the same slave (SRAM or TCM) through both data and instruction crossbars (AXBS), there is a possibility that either AXBS may hang, requiring a full system reset to clear. This scenario can only occur when program code resides in the device SRAM or TCM. This also applies if the CGM dividers are being reloaded with the same value that is currently loaded.

**Workaround:** Set the AXBS frequency/divider ratios during initialization by configuring the MC\_CGM\_SC\_DCx registers to the desired settings, and do not change the dividers during execution of the application. If the clock divider settings must be changed during execution of the application, then the application must be run such that the program code resides in flash and the data in SRAM/TCM, and all masters apart from CPU are not active.

**ERR010326: SWT: Out of lockstep fault for instruction and data cache memories is incorrectly indicated when watchdog is configured in fixed address execution service mode**

**Description:** An out of lockstep fault indication for instruction and data cache memories(NCF[10]) will be incorrectly raised if all of the following conditions are true:

- The safety core (e200z420) is configured in lockstep execution mode.
- The safety core Software Watchdog Timer(SWT\_0) Service Mode field in SWT Control Register is set to 'Fixed Address Execution'(SWT\_CR[SMD] = 0b10).
- SWT\_0 is enabled.

If the above conditions are true, then if an attempt is made to change the value of the Instruction Address Compare 8 (IAC8) register in the safety core and the value of the IAC8 register is utilized by either storing the value of the IAC8 register to memory or by servicing a watchdog time-out event, then NCF[10] fault will incorrectly assert.

**Workaround:** 'Fixed Address Execution' service mode should not be used for SWT\_0 when the safety core is configured in lockstep execution mode.

**ERR010327: ADC: Incorrect channel under measure is indicated after sampling phase of conversion until conversion completes**

**Description:** The Main Status Register Channel under measure address field (ADC\_MSR[CHADDR]) indicates which ADC channel is currently performing a conversion. This field indicates the correct channel during the sampling phase of conversion, but will display an incorrect value in the subsequent phases until conversion is complete.

**Workaround:** User must only consider ADC\_MSR[CHADDR] to be valid when the ADC is in the sample phase of conversion. The Main Status Register Status of the ADC field shows when the ADC is in the sample phase (ADC\_MSR[ADCSTATUS] = 0b100).

**ERR010328: JTAGM: Software Reset bit does not auto clear when written with 1 when an external tool is not connected**

**Description:** The JTAGM module Software Reset bit (JTAGM\_MCR[SWRESET]) when set to 0b1 resets the state machine and counters inside the JTAGM module and should auto clear to 0b0. However, the auto clear of JTAGM\_MCR[SWRESET] only occurs with an external tool connected and will not auto clear when no external tool is connected, unless the Data Transfer Mode bit is set (JTAGM\_MCR[DTM]= 0b1) and that a valid TCK selection is made via JTAGM\_MCR[TCKSEL] bits.

**Workaround:** To use the Software Reset bit with no tool connected, user must ensure that the Data Transfer Mode bit is set (JTAGM\_MCR[DTM]= 0b1) and that a valid TCK selection is made via JTAGM\_MCR[TCKSEL] bits. This will ensure that the JTAGM module is correctly clocked with no external tool and therefore the Software Reset bit will clear.

**ERR010329: JTAGM: JTAGM Data out registers are writable regardless of state of the Data Transfer Mode bit**

**Description:** The Data Transfer Mode bit (JTAGM\_MCR[DTM]) is intended to control the ability of application software to write JTAG data to the JTAGM Data Output Registers (JTAGM\_DORn). However JTAGM\_DOR0, JTAGM\_DOR1, JTAGM\_DOR2 and JTAGM\_DOR3 are writable regardless of the state of the JTAGM\_MCR[DTM] bit.

**Workaround:** If application software generated data is required to be written to the JTAGM Data Output Registers then the permissions to write to these registers should be handled within the software to ensure no genuine JTAG Data required is overwritten.

**ERR010330: JTAGM: An unexpected interrupt will occur if the Idle Interrupt is enabled after the end of frame transfer**

**Description:** After a JTAGM frame has completed when the JTAGM Idle Interrupt was not enabled (JTAGM\_MCR[IIE] = 0b0), if the JTAGM module is then configured to give an interrupt (JTAGM\_MCR[IIE] = 0b1), an unexpected interrupt will be seen.

**Workaround:** Before enabling a JTAGM idle interrupt by setting JTAGM\_MCR[IIE] = 0b1, set the Idle bit (JTAGM\_SR[Idle]) to 0b1 to clear any previous interrupts.

**ERR010331: NAL: Trace connections to the device are lost if a device reset occurs**

**Description:** During reset, the Nexus Aurora transmit pins (TXxN/TXxP, where x is 0 through 3) are put into a high impedance state until either self-test completes or the Self-Test Control Unit (STCU) determines that self-test is disabled. In addition, the system clock frequency is reset and the system clock is set to the Internal RC oscillator. Therefore, if the device is reset, the system clock is reset to a frequency that is less than 1/10 of the Nexus trace clock, the connection to a tool will be disconnected. The tool will have to establish a new connection with the device.

**Workaround:** The user should expect the trace connection to be lost through a reset. Tools will have to be reconnected following the reset and the pins re-enabled.

## **ERR010337: AFE: The AFE\_PLLSTS[LOR], AFE\_OSCSTS[STS], and MC\_ME\_GS[S\_XOSC] bits do not indicate loss of reference status**

**Description:** The Sigma Delta PLL (SDPLL) status register's Loss of Reference bit AFE\_PLLSTS[LOR], Oscillator Status Register Fail bit and Status bit (AFE\_OSCSTS[FAIL], AFE\_OCSTS[STS]), and Mode Entry Global Status External Oscillator Status bit (MC\_ME\_GS[S\_XOSC]) do not provide the correct status of the SDPLL loss of reference when SDPLL reference clock is removed in the system. The incorrect bit states are listed below:

AFE\_PLLSTS[LOR] improperly indicates 0  
AFE\_OSCSTS[STS] improperly indicates 1  
MC\_ME\_GS[S\_XOSC] improperly indicates 1

**Workaround:** Use the Clock Monitor Unit 0 (CMU\_0) to detect the status of the oscillator clock.

## **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, the ICNT and the Branch History field (HIST), if Branch History trace is enabled, 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) or Boot Assist Module (BAM) after reset (in other words, the System Status and Configuration Module [SSCM] boots directly to user code if a valid Reset Configuration Half-Word is found), perform an indirect branch instruction shortly after reset to reset the ICNT (and HIST if Branch History mode is enabled). 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 the BAF or BAM, an indirect branch will occur during the BAF/BAM execution and the tool trace will be re-synchronized prior to the execution of user code.

## **ERR010345: PMC: The Power Management Controller (PMC) module Analog Front End (AFE) Low Voltage Detect (LVD) interrupt may not work as a valid source of interrupt**

**Description:** The Power Management Controller (PMC) module interrupt (interrupt request number (IRQ) 477) is used to signal Low Voltage Detect (LVD) events on the internal AFE voltage regulators (VREGs). If these are enabled by setting the associated bits for each individual VREG in the AFE Interrupt Enable Register (PMC\_AFE\_INTR\_ENA[AFE\_INT\_EN\_VREGx] = 0b1 and PMC\_AFE\_INTR\_ENA[IE\_EN] = 0b1) then any interrupt request generated only occurs for two peripheral bridge clock cycles. If another interrupt is being serviced when the request is raised then the AFE LVD interrupt subroutine will sometimes not be executed, meaning it cannot be used reliably.

**Workaround:** Users should not consider the PMC module interrupt (IRQ 477) to be a valid source of interrupt. However, if an AFE VREG LVD event is configured as a source of destructive reset (by setting the corresponding bit in the MCB\_AFE\_LVD\_MASK register) then the Reset Generation Module (MC\_RGM) will latch the request for destructive reset in the case of the LVD event.

## **ERR010352: SPT: AHB error status can be asserted for the incorrect SPT DMA when performing back to back AHB accesses**

**Description:** When back to back Data Crossbar AHB accesses are performed by different Signal Processing Toolbox (SPT) DMA masters (Command Sequencer DMA (CS-DMA), Programmable DMA (PDMA) or Sample DMA (SDMA)) and there is an AHB error response in the data phase of the last access beat of the first DMA, and the second DMA access has simultaneously entered the address phase then the error response captured is recorded by the SPT for the second DMA access. This causes the incorrect error AHB Error Response status bit to be set in the DMA Error Status Register (SPT\_DMA\_ERR\_STATUS).

**Workaround:** The memory address associated with a Data Crossbar AHB error response is reported for System Memory Protection Unit (SMPU) protected memory regions and ECC errors. These errors will be reported and logged in the Memory Error Management Unit (MEMU) error reporting table. This address can be compared with the particular application DMA configuration to identify which DMA truly caused the error response (by comparing the error address against the configured DMA operating address ranges) and it's memory access behaviour can be corrected. Accesses made to reserved memory areas will not be logged in the MEMU error reporting tables so the user must ensure that none of the SPT DMAs are configured to attempt an access into these illegal areas.

## **ERR010357: PASS: JTAG password match bypasses flash read protection set by PASS.LOCK3[RLx] bits**

**Description:** On a censored device, setting the region lock bits in the Pass Lock Register 3 (PASS.LOCK3[RLx]) of the Password and Device Security Module (PASS) should protect the contents of flash memory blocks from being read when a debug tool is attached and the block is enabled for any flash region that has the region lock set. Debug enable may be achieved either through clearing the PASS.LOCK3[DBL] bit or providing a matching JTAG password.

However, providing a matching JTAG password bypasses the region lock control of the PASS.LOCK3[RLx] bits and allows all flash regions to be read, regardless of the PASS.LOCK3[RLx] bit settings.

**Workaround:** It is not possible to maintain the flash read protection on locked regions if the JTAG password is used to enable the debug interface.

In order to maintain the read protection on locked regions of the flash, only enable the debug interface by clearing the Debug Interface Lock bit (PASS.LOCK3[DBL]) bit. The PASS.LOCK3[DBL] bit can be cleared by providing the correct password sequence to a password challenge/response, by other external interface, such as CAN using software executing on the MCU.

For parts that support using specific JTAG passwords to invalidate the JTAG password comparison, an invalid JTAG password can be used in UTEST flash to disable this feature. Otherwise, programming a random value for the JTAG password match, which is not communicated to users, can greatly reduce the chances of enabling the debug interface and unlocking flash read access.

### **ERR010365: MC\_ME: An invalid mode (Clock Usage) configuration error can occur if the Sigma-Delta PLL and PLL1 are switched on or off independently**

**Description:** Performing a system mode transition which switches the Sigma-Delta PLL (SDPLL) and PLL1 to different states (using the PLL control bits in the Mode Entry Module Mode Configuration Registers (MC\_ME\_[mode]\_MC[SDPLLON] and MC\_ME\_[mode]\_MC[PLL1ON])) with any enabled peripherals selecting either of these as their clock source can cause an Invalid mode configuration interrupt (Clock Usage) (Interrupt Status Register MC\_ME\_IS[I\_ICONF\_CU] = 0b1).

**Workaround:** User must ensure any mode transition that switches on or switches off SDPLL and PLL1 sets both to the same on or off state in the target mode. This can be done by setting MC\_ME\_[mode]\_MC[SDPLLON] and MC\_ME\_[mode]\_MC[PLL1ON] both to zero or both to one.

### **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`.

### **ERR010391: LFAST: Receiver pads do not meet input hysteresis specification**

**Description:** The LVDS Fast Asynchronous Transmission (LFAST) receiver pads do not meet the data sheet specification for input hysteresis ( $V_{\text{hys\_drf}}$ ). The LFAST electrical characteristics state a minimum of 25mV for this parameter but some devices do not meet this specification.

**Workaround:** Do not expect each device to meet the data sheet specification for LFAST input hysteresis.

### **ERR010393: CGM: CLK\_OUT0 and CLK\_OUT1 dividers may become stuck if clock selection is changed while dividers with divide by 2 are operational.**

**Description:** If clock out functionality is enabled on either CLK\_OUT0 and/or CLK\_OUT1 and there is a divide by 2 divider operational on these clocks (via CGM\_AC14\_DC0[DE] and/or CGM\_AC9\_DC0[DE] = 0b1), then if the clock selection for CLK\_OUT is changed via the MCB\_CLKOUT\_SEL register or a Phase 1,2 or 3 reset occurs then the dividers may become stuck, causing no clock to be output from the divider. This will not clear until a power on reset occurs. This is only true if the divider is using divide by 2 (CGM\_AC14\_DC0[DIV] and/or CGM\_AC9\_DC0[DIV] = 0b1).

**Workaround:** In order to avoid the divider becoming stuck when performing a device reset or changing clock selection when using divide by 2, the user is recommended to follow the steps below:

1. Configure the divider value to divide by 3 by setting CGM\_AC14\_DC0[DIV] and/or CGM\_AC9\_DC0[DIV] = 0b10, or any value not equal to 0b0 or 0b1
2. Wait for the divider update to complete, shown by the divider update status bit CGM\_DIV\_UPD\_STAT[17]=0b0 and/or CGM\_DIV\_UPD\_STAT[22]=0b0
3. Disable the divider by setting CGM\_AC14\_DC0[DE] and/or CGM\_AC9\_DC0[DE] = 0b0
4. Wait for CGM\_DIV\_UPD\_STAT[17]=0b0 and/or CGM\_DIV\_UPD\_STAT[22]=0b0
5. Change the clock selection required in the MCB\_CLKOUT\_SEL register
6. Wait for CGM\_DIV\_UPD\_STAT[17]=0b0 and/or CGM\_DIV\_UPD\_STAT[22]=0b0
7. Enable the divider by setting CGM\_AC14\_DC0[DE] and/or CGM\_AC9\_DC0[DE] = 0b1
8. Wait for CGM\_DIV\_UPD\_STAT[17]=0b0 and/or CGM\_DIV\_UPD\_STAT[22]=0b0
9. Configure the divider value to the desired division factor by setting CGM\_AC14\_DC0[DIV] and/or CGM\_AC9\_DC0[DIV] accordingly
10. Wait for CGM\_DIV\_UPD\_STAT[17]=0b0 and/or CGM\_DIV\_UPD\_STAT[22]=0b0
11. If usage of CGM\_AC14\_DC0[DIV] = 0b1 and/or CGM\_AC9\_DC0[DIV] = 0b1 is only required temporarily then it is advised to re-configure the divider after use to CGM\_AC14\_DC0[DIV] and/or CGM\_AC9\_DC0[DIV] = 0b10 or any value not equal to 0b0 or 0b1

If it is expected that a device reset can occur whilst the dividers are enabled it is recommended not to use the divide by 2 option to ensure the clock out dividers do not hang.

If the clock monitoring unit (CMU) is being used to monitor CLK\_OUT0 (via CMU\_6) then this must also be disabled before CGM\_AC14 is disabled during the steps above.

## **ERR010414: PMC: Low Voltage Detect (LVD) self test may incorrectly indicate a self test fail if the supply is outwith its operating range during power up**

**Description:** The Power Management Controller (PMC) executes a self test of the Low Voltage Detect (LVD) and High Voltage Detect (HVD) mechanisms during device start up (at Phase 3 of the reset sequence) after a power on reset or a reset due to an LVD event. If a scenario occurs during this self test where a ramping-up supply is out of the operating range when that supply's monitor is being tested, the PMC self test would indicate a failure correctly once self test execution has completed (indicated by `PMC_SELF_TEST_UM_VD_REG[ST_RESULT]=0b0` when `PMC_SELF_TEST_UM_VD_REG[ST_DONE]=0b1`). This means that if a failure is evident after testing it is not certain that a genuine failure has occurred or whether the supply may have been outwith the operating ranges whilst the test was executing

**Workaround:** In the event of a PMC LVD/HVD self test failure (`PMC_SELF_TEST_UM_VD_REG[ST_RESULT]=0b0`), the user should wait until all supplies return to the correct operating range, and then perform a software initiated PMC self test through `PMC_SELF_TEST_UM_VD_REG[ST_MODE]=0b01`. Once the software initiated self test has completed it will now indicate the correct status via the result register `PMC_SELF_TEST_UM_VD_REG[ST_RESULT]`.

## **ERR010436: ZipWire: SIPI can have only one initiator with one outstanding write frame at time**

**Description:** The Serial Inter-processor Interface (SIPI) module of the Zipwire interface only supports one initiator and one outstanding write frame at a time.

If a new write is initiated (by setting `SIPI_CCRn[WRT] = 0b1`, where n is the respective channel number for the transmission), or a new streaming write is initiated (by setting `SIPI_CCRn[ST] = 0b1`) with acknowledgement of a previous frame pending, then the initiator node may get a timeout error (indicated by `SIPI_ERR[TOEn]=0b1`). The previous write frame last byte may also be corrupted at the target node.

This also means that the target node cannot initiate a write transfer while the initiator node is in the process of a write transfer.

**Workaround:** The initiator should maintain only one outstanding write/streaming write frame to the target node at any one time.

The user must ensure that before initiating a new write request or initiating a new streaming write that it has received an acknowledgement for the previous write transaction (indicated by `SIPI_CSRn[ACKR] = 0b1`). The write acknowledgement interrupt can be enabled by setting `SIPI_CIRn[WAIEn]=0b1`.

Implement a protocol that ensures both sides of the link cannot initiate a transfer at the same time. For example, a token-passing protocol could be implemented using the SIPI trigger command feature. Send a trigger command to pass the token to the other end of the link. Upon receipt of the trigger command, either initiate a write transfer if one is pending, or pass the token back by sending a trigger command. If a write transfer is initiated, wait until ACK is received and then send a trigger command to pass the token back. In this manner, if each side agrees only to initiate a transfer when it obtains the token, there will be no simultaneous transfers that can cause the problem described.

**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.

**ERR010455: ADC: Calibration routine needs to be repeated after any SoC reset that triggers device built-in self-test**

**Description:** The Successive Approximation Register Analog-to-Digital Converter (SAR-ADC) Calibration, BIST Control and Status register (ADC\_CALBISTREG) and Offset and Gain User register (ADC\_OFSGNUSR) are used to configure the settings used in SAR-ADC's calibration routines and should retain any user programmed values after short and long functional resets.

However it is found that the values of these registers along with calibration routine generated internal values are reset after any SoC reset that triggers offline or online built-in self-test (BIST), which includes long functional reset from an external source (RESET\_B) when RGM\_FESS[SS\_EXR] = 0b0.

The value of these registers goes to the reset value (at POR) even if self-test bypass is configured.

**Workaround:** The user should ensure that if the SAR-ADC is used in the application then the high accuracy mode ADC calibration routine should be re-run after any reset that causes BIST to be executed. If non-default calibration routine settings are used then the ADC\_CALBISTREG and ADC\_OFSGNUSR registers should be re-programmed before re-running the calibration routine.

**ERR010458: RGMII: Ethernet clock output pad 107 can glitch during clock transitions of ENET operation**

**Description:** When changing protocol speed in the Ethernet MAC (ENET) controller in RGMII mode it in turn will change the speed of the clock generated on pad 107 (PG[11]). Depending on the mode selected via the ENET\_RCR[RMII\_MODE], ENET\_RCR[RMII\_10T], ENET\_RCR[RGMII\_EN], and ENET\_ECR[SPEED] registers this pad will change the ENET RMII Reference clock (RMII\_CLK) or the GTX Clock for RGMII (TXC). It has been found that speed transitions such as the following:

RGMII mode enabled and clock on TXC reduced from 125 Mhz to 25 Mhz for 100 Mbps operation or vice versa

RGMII mode enabled and clock on TXC reduced from 125 Mhz to 2.5 Mhz for 10 Mbps operation or vice versa

RGMII mode enabled and clock on TXC reduced from 25 Mhz to 2.5 Mhz for 10 Mbps operation or vice versa

can cause the TXC clock output on pad 107 to not meet the Tcyc Min specification under 'RGMII Switching Section' of the device datasheet and could manifest as a glitch. This is limited to the clock transitioning edge.

It is expected that in most cases the ethernet physical interface IC (PHY) will be able to ignore this glitch on the clock supplied from pad 107 since no transmission will be occurring during the initialisation or clock speed changes, but the impact has to be determined by the individual PHY in use.

**Workaround:** Since this issue only occurs during speed changes or module initialisation it is not expected to cause any issues since there will be no data being transmitted during this time, however there are two potential workarounds:

1) Ensure that no external clock is supplied to the input of pad 130 (REF\_CLK function on PI[2]) when performing initialisation / speed switching on the ENET module. This will ensure that no clock is generated on the output of pad 107 and therefore no glitch would occur. Depending on the PHY used however this may not be possible.

2) Ensure that the external PHY will not be affected by any glitch on the clock by either powering down the PHY or assert reset to the PHY whilst the ENET module is being initialised / speed changes are being made.

Since the 125MHz input clock to pad 130 for RGMII mode is supplied from an external source, users should also note that any glitch on this supplied clock at any time will also be propagated to the output clock on pad 107.

## **ERR010459: SPT: Spurious CS\_AHB\_ERR can be generated in the SPT\_DMA\_ERR\_STATUS register**

**Description:** The Signal Processing Toolbox (SPT) command sequence is provided by application software and written to a buffer in the system RAM or flash, where command sequencer DMA (CSDMA) fetches or alternatively transfers the list of instructions to the command queue.

It has been found that the following sequence can occur:

- Whilst the CSDMA is performing a 2-part burst access a high priority high speed crossbar (AHB) master Programmable DMA or Sample DMA (PDMA or SDMA) access occurs.
- The CSDMA will complete its first burst access, then the AHB will service the PDMA/SDMA access. The pending second burst of the CSDMA is stored at this point.
- Once the PDMA/SDMA access completes, the CSDMA will resume its second burst access, but if the SPT code being executed at this time is a JUMP or NEXT instruction in the same cycle as the second access resumes, then the address on the AHB BUS can get corrupted.
- This means that additional instruction fetches from unwanted locations would occur.
- If the address fetched resides in protected or un-implemented memory then a spurious CS AHB Error (SPT\_DMA\_ERR\_STATUS[CS\_AHB\_ERR]=0b1) would occur without any illegal opcode error (SPT\_CS\_STATUS1[ILL\_OPCODE]=0b1). If the address fetched resides in normal memory space no error will be generated.
- In no cases (even if valid code is read from the incorrectly updated address) will these instructions be executed by the command sequencer, since the JUMP / NEXT instruction will invalidate the tags for the remaining instructions. The program flow would continue on as normal.

This sequence of events is considered extremely unlikely to occur.

**Workaround:** CS AHB Error should only be checked in the SPT\_DMA\_ERR\_STATUS register when an Illegal Opcode indication in SPT\_CS\_STATUS1 register occurs. Otherwise isolated CS AHB Errors cannot be considered as valid.

#### **ERR010479: NPC: Nexus enable required for mode changes when a debugger is attached**

**Description:** If the Nexus interface is enabled in the the e200zx cores, even if trace (program, data, ownership, watchpoint, data acquisition) is not enabled, the Nexus Port Controller (NPC) tracing must be enabled to allow mode changes via the Mode Entry module if debug mode is enabled (debugger connected to the MCU) since some Nexus trace messages are automatically generated regardless whether any trace mode is disabled. Nexus is enabled in the core if any Nexus feature is accessed by a tool (executing the Nexus\_enable command to use the Nexus Read/Write Access feature to access memory).

**Workaround:** NPC tracing must be enabled by enabling the Message Clock Output (MCKO) in the NPC Port Configuration Register (NPC\_PCR) when a debugger is connected to allow messaged to exit the core Nexus module. In addition, the Full Port Mode bit should also be set.

#### **ERR010486: PLLDIG: Device may not exit the reset sequence if a reset occurs whilst PLL1 is running with Modulation enabled.**

**Description:** PLL1 has ability to run with modulation enabled, which is controlled by PLLDIG\_PLL1FM[MODEN] bit. The PLL1 nominal frequency (fp11\_phi) output must be configured such that the output is within the operating frequency as outlined in the Reference manual, this includes the Modulation Depth (MD) configured . When the PLL1 has locked and is running with modulation enabled and a reset (from any source) is applied to the system then the PLL1 output can overshoot for a short duration which can cause the system to become stuck in reset.

**Workaround:** There are 3 potential workarounds:

1. Run PLL1 with Modulation disabled (PLLDIG\_PLL1FM[MODEN] = 0b0)
2. Use an external watchdog circuit and power supply monitor to determine if the MCU has become unresponsive. The external circuitry should be configured to assert the device power-on reset signal (VREG\_POR\_B) if the device becomes unresponsive. This will ensure that the device can safely return to normal operation as per the recommendations of the device safety manual.
3. Configure the PLL1 output frequency such that the MD of the mode configured (PLLDIG\_PLL1FM[MODSEL]) plus the PLL nominal frequency (fp11\_phi) is 10% lower than the frequency specified in the device reference manual for maximum clock frequencies for CORE1\_CLK/CORE2\_CLK.

**ERR010490: STCU2: Register values STCU2\_PLL\_CFG[PLLLDF, PLLODF], STCU2\_LBUFM and STCU2\_LBRMSW can change at end of self test**

**Description:** If the value of the Self Test Control Unit (STCU) PLL configuration register PLL output division factor and PLL loop division factor bits (STCU2\_PLL\_CFG[PLLODF, PLLLDF]), STCU Logic BIST (LBIST) unrecoverable fault mapping register (STCU2\_LBUFM) or the STCU On-Line LBIST reset management register (STCU2\_LBRMSW) are programmed to a value other than zero, then after completion of self test these bits can change to zero.

STCU2\_PLL\_CFG register controls the PLL configuration during offline self test and is loaded from values pre programmed in flash dcf records. The values programmed in the flash dcf records are correctly used for programming the PLL during offline self test, but the value that is read post self test may not reflect the value programmed from the original flash dcf record.

STCU2\_LBUFM controls the fault mapping of faults that occur during self test and are reported at end of self test. Each LBIST partition can be configured to generate either a Recoverable fault (Fault Collection and Control Unit (FCCU) NCF[7] - Non-critical fault indication from STCU) or a Non-Recoverable fault (FCCU NCF[6] - Critical fault indication from STCU). For offline it is programmed from flash dcf records and before running online self test these can be programmed. However the programming of these bits cannot be guaranteed to correctly map the fault, in which case it will return to a default of 'Recoverable Fault mapping' which will raise the NCF[7] flag in the FCCU.

STCU2\_LBRMSW is intended to provide reset management control for the online self test. However this is not used in the device and regardless of the programmed value the device would generate a reset at end of online self test.

It should be noted that in all cases the offline self test will complete correctly, so no functional safety impact is expected.

**Workaround:** For offline self test user should ignore the values of STCU2\_PLL\_CFG[PLLLDF, PLLODF] after self test completion and do not update STCU2\_LBUFM to a value other than zero via UTEST.

For online self test do not program STCU2\_LBUFM and STCU2\_LBRMSW to a value different from its reset value (0x0).

The fault reactions for online self test can be programmed using NCF[7] non-critical fault indication from the FCCU module.

**ERR010537: AFE: Device may not exit the reset sequence if a low voltage detection on VDD\_HV\_RAW or VDD\_HV\_DAC supplies occurs when XOSC is the source of PLL0/1 clock.**

**Description:** If a low voltage detection (LVD) event occurs on the VDD\_HV\_RAW or VDD\_HV\_DAC supplies then the default reaction is not to perform a destructive reset of the device (reset reaction for LVD on AFE supplies is enabled by writing to MCB\_AFE\_LVD\_MASK). However it has been found that when the XOSC is used as the source for PLL0 or PLL1 clock then following an LVD event on these supplies there is a chance that the device may become unresponsive if no reset reaction is enabled, or may not exit the reset sequence if the reset reaction is enabled. This occurrence is considered to be extremely rare, but is theoretically possible.

**Workaround:** Use an external watchdog circuit to determine when the MCU is not responding because of brownout conditions on external power supplies and adhere to the Safety Manual guidelines if the stuck in reset condition occurs. Asserting the device power-on reset signal (VREG\_POR\_B) when the MCU becomes unresponsive or when any power supply is outwith the operating range will bring the device back into normal operation.

**ERR010638: NZxC3: Nexus messaging becomes corrupted 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 core is disabled, with the clock gated (as configured in the relevant core control register MC\_ME\_CCTLx for the requested mode) then the Nexus3 interface issues a core debug status message indicating the transition into core STOP mode (core disabled with clock gated). It is possible during this message transmission, that the core Nexus clock gets disabled. This blocks the Nexus interface, and the message data can be left pending on the interface until the core clock resumes. This causes overflow of the NAL (Nexus Aurora Link) FIFO and corrupts the Aurora protocol.

**Workaround:** Possible workarounds are:-

- 1) If Nexus traces are enabled then mode transitions that disable device cores should not be executed.
- 2) If Nexus traces are enabled and a core is needed to be stopped, then that core should use PowerPC 'wait' instruction instead of stop mode.

**ERR010642: AFE: XOSC overvoltage stress and AFE bandgap overvoltage stress when LBIST partition number 6 is executed during self test**

**Description:** During the execution of self test on LBIST partition number 6 the analog front end (AFE) XOSC internal termination resistors are disabled (default state is enabled). This is true for both offline self test and online self test. If the device is configured with an external LVDS differential XOSC input which is active during the time that the termination is disabled then the XOSC inputs will overshoot causing an overvoltage stress on the XOSC pads.

For online self test this also limits the usage of the XOSC in differential bypass mode as a source of clock for the PLL if LBIST partition number 6 is executed during online self test.

It has also been found that if the SD\_R external resistor is not present in the board design then the internal AFE voltage regulators will also overshoot, causing a voltage overstress on the internal LV supplies when LBIST partition number 6 is executed.

**Workaround:** There are 2 workarounds available:

1. Exclude LBIST partition number 6 execution in both online and offline self test modes by removing this partition from the configuration. Analysis has shown that there will be minor safety impact caused by the removal of LBIST partition 6 from self test.
2. If LBIST partition 6 is required to be executed in offline or online BIST then ensure that any external LVDS clock source used is removed during LBIST partition 6 execution. It is also required that the 40.2Kohm SD\_R resistor recommended in the device datasheet is present in

the hardware. If the application does not use the device SDADC (Sigma-Delta ADC) and DAC (Digital to Analog Converter) then the SD\_R resistor value can be relaxed to within the range of 40-47 Kohm.

**ERR010682: STCU2: Online LBIST of partition 6 incorrectly generates reset when run with AFE LVD unmasked**

**Description:** During online logic built in self-test (LBIST), if the analog front end (AFE) low voltage detect (LVD) are configured as a source of reset (either of MCB\_AFE\_LVD\_MASK[AFE\_LVD4\_MASK], MCB\_AFE\_LVD\_MASK[AFE\_LVD3\_MASK] or MCB\_AFE\_LVD\_MASK[AFE\_LVD2\_MASK] set to 1b1), then execution of LBIST on partition 6 (LB6) results in a destructive reset.

If MCB\_AFE\_LVD\_MASK[AFE\_LVD9\_MASK] is set to 1b1 before running online LBIST, then after online self-test RGM\_DES[F\_AFE\_LVD9] bit is set but no reset occurs.

**Workaround:** Any one of below workarounds can be applied:

1) Run online LBIST with AFE LVDs masked as a source of reset (All of MCB\_AFE\_LVD\_MASK[AFE\_LVD4\_MASK], MCB\_AFE\_LVD\_MASK[AFE\_LVD3\_MASK], MCB\_AFE\_LVD\_MASK[AFE\_LVD2\_MASK] and MCB\_AFE\_LVD\_MASK[AFE\_LVD9\_MASK] set to 1b0).

OR

2) Do not run LBIST on partition 6 during online self-test.

**ERR010750: PMC: Device may not exit the reset sequence if a low voltage detect event occurs during execution of offline or online self-test**

**Description:** The device may not exit the reset sequence if a low voltage detect (LVD) event occurs during offline or online Self Test Control unit (STCU2) built-in self-test (BIST) execution, on any LVD monitored supply other than VDD\_HV\_IO.

**Workaround:** Use an external watchdog circuit to determine when the MCU is not responding because of brownout conditions on external power supply and adhere to the Safety Manual guidelines when the stuck in reset condition occurs. Asserting the device power-on reset signal (VREG\_POR\_B) when the MCU becomes non-responsive or when any power supply is outside the operating range will bring the device back into normal operation.

**ERR010879: SPT: Adaptive scaling using count unoccupied bits from bit 23 down mode can introduce harmonics in the FFT output**

**Description:** When adaptive scaling is enabled using count unoccupied bits from bit 23 downwards saturation can occur in the FFT calculation. This results in harmonics being introduced in the FFT result.

**Workaround:** There are two possible workarounds:

1. Use adaptive scaling in the mode where unoccupied bits are counted from bit 15 downwards by clearing the ADPTV\_SHFT (bit 113) bit in the RDX4 instruction.

2. If adaptive scaling is used counting from bit 23 downwards then the value of the twiddle factors should be divided by  $\text{SQRT}(2)$ . For this workaround to be valid twiddle factor quadrature extension cannot be used and the maximum FFT size is 512 points.

### **ERR010880: SWT: SWT may not get serviced in Fixed and Incremental address mode**

**Description:** The "Fixed Address Execution" and "Incremental Address Execution" modes may not work properly for servicing the Software Watchdog Timer (SWT). When a core tries to service its corresponding SWT by generating a pulse for the SWT upon execution of the code at the address loaded into the IAC8 register, then the SWT may miss the pulse and therefore not get serviced.

**Workaround:** Use the "Fixed Service Sequence" or "Keyed Service Sequence" modes to service the SWT.

### **ERR010900: FCCU: False indication of a fault state for a single safe clock period can be generated on the error output pin**

**Description:** The error out pin from the Fault Collection and Control Unit (FCCU) may pulse to a logic low (0b0) when the following conditions are fulfilled:

- software changes the error out protocol from a toggling protocol to a not-toggling protocol, and programs the `FCCU_CFG.FCCU_SET_AFTER_RESET` bit to 0b1
- software switches the Fault Collection and Control Unit (FCCU) state machine from CONFIG to NORMAL state

The duration of the glitch is equal to a single clock period of the Internal RC oscillator and there is a 50% of probability of the pulse occurring.

**Workaround:** Split the configuration of the FCCU in 2 phases.

During the first phase, software should do the following:

- 1) move the FCCU to the CONFIG state
- 2) configure the FCCU including the error out protocol, but without setting the `FCCU_CFG.FCCU_SET_AFTER_RESET` flag to 0b1 (leave as 0b0)
- 3) exits to the NORMAL state

During the second phase, software should do the following:

- 4) move the FCCU to the CONFIG state
- 5) set the `FCCU_CFG.FCCU_SET_AFTER_RESET` flag to 0b1
- 6) exit to the NORMAL state

**Note:** The default (after reset) error out protocol is the Dual Rail. Since this is a toggling protocol, the software must execute the above steps each time the user wants to switch to a non-toggling error out protocol.

## **ERR010997: SPT: Work register contents not deterministic after execution of STOP command**

**Description:** When the SPT (Signal Processing Toolbox) executes a STOP command at the end of a command sequence the work register (WR) values are not deterministic until they are re-initialized. Values read from the work registers before they have been re-initialized are not reliable.

**Workaround:** The CPU can re-initialize the WRs after STOP command has concluded SPT command execution by writing any value to the WRs. The CPU should not rely on read access to the work registers (WR) between a STOP command and the launching of a new command sequence, instead the WAIT and EVT commands should be used to manage the timing of accesses when both SPT and CPU must interact with work registers during command sequence execution.

## **ERR011032: PMC: Unexpected events when an LVD occurs on a supply with its LVD masked**

**Description:** Device low voltage detect (LVD) can be masked/disabled in the following situations:

- a) Optionally during normal functional mode using the PMC\_PMCCR register.
- b) During online and offline STCU self-test, by default all LVDs and HVDs are masked except VDD\_HV\_IO LVD using the DCL\_IPS\_0 control in UTEST.
- c) During PMC self test the LVD/HVD for the supply under PMC self test and the subsequent supply to be under self test are automatically masked

The following symptoms occur if an LVD occurs on a supply whose LVD has been masked/disabled:

- 1) If LVD occurs on VDD\_HV\_PMC while it is masked: During online STCU, PMC self-test or normal operation, reset will not be masked and a reset event (corresponding to External Reset) would occur. During offline STCU operation where RESET\_B is already asserted, LBIST 4 partition will fail self-test and the FCCU and MC\_RGM status will not reflect the LVD event.
- 2) If LVD occurs on VDD\_HV\_IO while it is masked: The 16MHz IRCOSC will stop working and its output may glitch, but will restart when supply recovers above LVD level. This could lead to the device becoming stuck in reset.
- 3) If LVD occurs on VDD\_HV\_ADC while it is masked: The SAR-ADCs will stop converting, but recommence when supply recovers above LVD level. The ADC cannot be used during STCU execution but should not be relied upon during PMC self-test execution or if PMCCR[LVD\_ADC\_ENABLE] has been written to 0b0 and the supply can fall below operating level.
- 4) If LVD occurs on VDD\_HV\_FLA while it is masked: flash reads and write/erase operations can be impacted.

**Workaround:** It is recommended to keep all supplies within specification and/or assert VREG\_POR\_B if they fall out of spec, or expect the above symptoms to occur if a supply goes below LVD level while the LVD is masked. During functional operation the PMCCR register ideally should not be used to disable LVDs/HVDs.

## **ERR011156: PMC: During online PMC self-test if RESET\_B is asserted then a spurious destructive or power on reset may occur and be reported as an LVD/HVD event in the RGM\_DES register.**

**Description:** When the device is configured to run PMC (power management controller) online self-test (via `PMC_SELF_TEST_UM_VD_REG[ST_MODE] = 0b1`) or single VD test (via `PMC_SELF_TEST_UM_VD_REG[ST_MODE] = 0b10`) there are two scenarios and configurations where if an external reset is applied via `RESET_B` (when configured as long functional reset via `RGM_FESS[SS_EXR] = 0b0`) during the PMC online self test of any supply that has the HVD/LVD unmasked in the `dcl_ips_0` register will cause the CGM (clock generation module) dividers to be bypassed, causing an unexpected destructive or POR reset. These scenarios arrive since the LVD Self Test Time Window Register (`PMC_STTW`) value will be written with a value based on a certain clock speed, but the CGM being bypassed results in a faster clock being applied, effectively meaning the self test time window decreases. The 2 cases are described as follows:

1. PMC module clock configured to be less than 16MHz. In this case, when reset occurs, the CGM clock divider gets bypassed and the PMC module clock changes to 16MHz, and this causes PMC self-test timing window to reduce, which in turn causes failures and a resulting reset (destructive or POR) to occur.
2. PLL and PCFS is enabled and PMC module clock frequency are configured to be higher than 16MHz. In this case, when RESET occurs, the PCFS ramp down slowly reduces the frequency, but as the CGM divider again gets bypassed, the system clock increases (e.g before reset, with PMC module clock of 60MHz when reset occurs this will become higher by the factor of CGM divider and then gradually reduce due to PCFS). This increase in frequency causes the similar effect as in 1st case.

Either scenario will be reported as an LVD/HVD event in the `RGM_DES` register with bits for either the supply that was under test, the subsequent supply to be tested, or both.

**Workaround:** To avoid any unwanted reset during PMC online self test ensure that when self-test is initiated via `PMC_SELF_TEST_UM_VD_REG[ST_MODE] = 0b1`, or single VD test via `PMC_SELF_TEST_UM_VD_REG[ST_MODE] = 0b10` the following clock configuration rules are applied:

1. PMC module clock frequency is greater than or equal to 16MHz or if the PMC module clock is configured to be less than 16MHz then programming the `PMC_STTW` register with a value of `0x172` (corresponding to 16MHz + 10% value) then no issue will occur.
2. PCS ramp-down is disabled on the selected clock source for the PMC module.

## **ERR011172: MIPICSI2: Start of Transmission Error can occur at lower speeds**

**Description:** At data rates < 333Mbps per lane, the high speed data plus high speed trail (`Ths_trail`) duration must be > 4 bytes (per lane). I.E., if a single byte of HS data is sent per lane, then the `Ths_trail` must be of duration >  $3 \cdot 8 \cdot UI$ , where `UI` is the Unit Interval at the associated data rate. If this condition is not met, a start of transmission error can occur. The start of transmission error may happen on the current high speed burst or subsequent bursts.

**Workaround:** The `Ths_trail` duration is typically a configurable value in the transmitter. When using data rates < 333Mbps, configure the transmitter in such way that the `Ths_trail` duration is always >  $3 \cdot 8 \cdot UI$ .

## **ERR011188: FCCU : Fault NCF[106] may occur when PLL Loss of Lock event occurs during MBIST execution when online STCU-selftest is run.**

**Description:** If a PLL loss of lock (LOL) event occurs during the Direct Memory Access (DMA) RAM Memory BIST (MBIST) portion of the online STCU Self Test then the Self Test will be aborted and the Fault Collection and Control Unit (FCCU) will report a Non-Critical Fault NCF[7] (STCU NCF). This will be indicated by NCF[7] being set in the FCCU\_NCF\_S0 register. It has been found that in addition to this the FCCU may unexpectedly also report a DMA RAM Alarm fault. This will be indicated by NCF[106] being set in the FCCU\_NCF\_S3 register.

**Workaround:** If the application makes use of the DMA module and the DMA RAM Alarm NCF[106] occurs then the software reaction to this fault should first ensure that the alarm is genuine by ensuring that no STCU online self test was in operation and was aborted due to a PLL LOL event. This can be determined by checking the STCU2\_ERR\_STAT[LOCKESW] bit. If STCU2\_ERR\_STAT[LOCKESW] is not equal to 0b1 then the alarm is genuine and software can take appropriate action to this alarm, otherwise it can be cleared (by writing 0x400 to FCCU\_NCF\_S3) and ignored. If STCU online self test is not used in the application then no action need be taken and if NCF[106] occurs it can be considered genuine and be dealt with accordingly.

It should be noted that following a failed STCU online self test due to the conditions described the STCU2\_ERR\_STAT[LOCKESW] bit will remain set until a subsequent self test is run. This needs to be considered as if no action to the failing self test (e.g re-run self test) is taken then this would mean that a genuine NCF[106] occurrence could be interpreted as a false flag.

## **ERR011245: AFE: AFE\_OSCSTS[STS] bit does not get cleared in single-ended and differential bypass mode of XOSC**

**Description:** The device AFE (Analog Front End) Oscillator Status Register is intended to reflect the selection and status of the XOSC. The AFE\_OSCSTS[STS] bit becomes set when the XOSC becomes active (and has completed the transition count set in AFE\_OSCDLY register) and is cleared when the XOSC is disabled using the MC\_ME module. The rising edge of the AFE\_OSCSTS[STS] bit will also raise an interrupt (IRQ# 746) if AFE\_OSCCTRL[IE] = 0b1, this interrupt is clearable by writing 0b1 to AFE\_OSCSTS[STS].

However it has been found that when the XOSC is configured in single-ended or differential bypass modes then the AFE\_OSCSTS[STS] bit and subsequent interrupt can only be considered valid the first time the XOSC is enabled in these modes. If the XOSC is disabled in the MC\_ME module this will not be reflected by AFE\_OSCSTS[STS], and no further interrupt will occur if the XOSC is re-enabled.

**Workaround:** When using single-ended or differential bypass modes of the XOSC, the user should use the MC\_ME\_GS[S\_XOSC] bit instead of the AFE\_OSCSTS[STS] bit for accurate status of the oscillator. If the AFE\_OSCSTS[STS] interrupt is to be used, and if XOSC needs to be disabled after initial enabling then before enabling the XOSC again, the user needs to first clear the AFE\_OSCCTRL[EN\_EXT] bit and then set this bit again.

## ERR011297: PMC: Default value of self-test time window (STTW) changed

**Description:** The Device Configuration Format (DCF) record that controls the Power Management Controller (PMC) self-test time window (PMC\_STTW) has a default of 0xA0. With the current default DCF clock configuration applied via the dcl\_ips0/1/2 records the PMC self-test will run from the 16MHz IRC clock with a divider of 1, therefore the value 0xA0 equates to a 10us timeout value. The datasheet specification for the STTW setting is 21us (0x150). It has been found that at the minimum operating voltage specification of any of the device supplies (excluding VDD\_LV\_CORE if external regulation mode is used) a small number of parts could fail to complete the power up PMC self-test within the shorter 10us window, which results in the part becoming stuck in an infinite reset loop.

**Workaround:** If the incorrect self-test time out value of 0xA0 is present (this can be determined by software reading the PMC\_STTW register after device power up) and the power supply used in the application could vary by more than 3% below the typical required value for each supply then the user should update the reset PMC\_STTW value to 0x150 via the dcl\_pmc\_self\_test\_time\_gap DCF record. This has the effect of increasing STTW time window from 10us to 21us (datasheet specification value) with a 16MHz clock. The updated dcl\_pmc\_self\_test\_time\_gap DCF should be programmed as a triple voted, spread address client by programming the following values to the next free DCF location in the UTEST flash:

0x000C015004000344

0x0003FEAF04000348

0x000600A804000350

## 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  $PDED \times 1/[ADC\_clock\_frequency]$  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 -

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

Where:

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

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

## ERR011329: SPT: After completion of a command sequence there is a time window in which work register access by CPU can cause the system to become non-responsive

**Description:** When the SPT completes execution of a command sequence there is a short time window in which access to an SPT work register by CPU can potentially cause peripheral bus 1 (PBRIDGE\_1/AIPS\_1) to become non-responsive and the CPU to stop executing instructions

because it has stalled waiting on the peripheral bus transaction to complete. Any other core that attempts to access peripheral memory over peripheral bus 1 will also stall and the system can only be recovered by asserting power on reset (POR).

Execution of a STOP command terminates the SPT command sequence execution and the SPT finite state machine (FSM) transitions from RUN state to STOP state for one SPT clock cycle then START state. A 'soft reset' signal is generated for one SPT clock cycle at the STOP to START state transition but only once the Command Sequencer DMA (CS DMA) has completed all command fetches. If a work register access is requested by the CPU at the same clock cycle as the 'soft reset' signal generation then the SPT will never acknowledge the access request and the peripheral bus will wait forever, making it non-responsive and causing any cores that attempt access to hang. The scenario is also possible if the SPT FSM transitions to STOP state then START state by software assertion of CS\_MODE\_CTRL[STOP] bit or if the FSM transitions to ASYNCSTOP state then START state by software assertion of CS\_MODE\_CTRL[ASYNCSTOP] bit.

Practically, the only way this can occur is if the CPU polls for assertion of SPT\_CS\_STATUS0[PS\_STOP] or SPT\_CS\_STATUS0[PS\_ASYNCSTOP] then immediately makes repeated work register accesses so that one of these accesses coincides with SPT 'soft reset' signal assertion. Access to the SPT status registers cannot cause the failure.

The duration of the time window depends on how long the CS DMA is active after execution of the STOP command. Assuming configuration:

- SPT DMA bus master has highest priority on the XBAR.
- Burst size of INCR16 is set for CS DMA.
- There is no SPT acquisition in progress meaning the sample DMA (SDMA) cannot pre-empt the CS DMA between bursts.
- SPT instructions stored in SRAM (if instructions are fetched from flash memory the window duration will always be greater and subject to arbitration delay at the flash controller port.)
- XBAR clock (SYS\_CLK) at 133 MHz (MPC577xK/N) or 120 MHz (S32R274/S32R372).

The possible worst-case duration can be calculated:

- 1024-bit transfer (64-bit \* 16) takes 18 cycle for INCR16.
- $18 \text{ cycle} * (1/133 \text{ MHz}) = 135.3\text{ns}$ .  $18 \text{ cycle} * (1/120 \text{ MHz}) = 151.2\text{ns}$ .
- SPT CS DMA always fetches 16 instructions, in worst case the STOP command is the first.  $128\text{-bit} * 16 = 2048\text{-bit}$ .
- Then 2 bursts are needed. At 133 MHz XBAR this is  $2 * 135.3\text{ns} = 270.6\text{ns}$ . At 120 MHz XBAR this is  $2 * 151.2\text{ns} = 302.4\text{ns}$ .

**Workaround:** An interrupt service routine (ISR) should be used to react to SPT command sequence completion because the latency between the interrupt request being raised on assertion of CS\_STATUS0[PS\_STOP] and execution of the first instruction in the ISR is greater than the worst-case time window duration for assumed configuration. However if CPU polling must be used then measures must be taken before safely accessing the work registers upon completion of the command sequence:

- Check CS\_STATUS0[PS\_STOP] = 0b1 (This bit indicates that SPT state machine has been in STOP state and so has executed the STOP command.)
- Once CS\_STATUS0[PS\_STOP] assertion has been detected the user must clear it by writing 0b1.
- Check CS\_STATUS3[CS\_DMA\_ON] = 0b0 (Ensures that all Command Sequencer fetches have been completed.)

Alternatively, enforcing a time delay greater than the worst-case window duration between execution of STOP command and accessing a work register would ensure the time window has elapsed. The delay must begin from the detection of CS\_STATUS0[STOP] = 0b1.

#### **ERR011394: CTE: Timing table is not executed when the CTE time base clock is set to 80 MHz and the table execution duration is 1 clock cycle**

**Description:** The CTE time base (datapath) clock is derived from the 80 MHz CTE module clock. It is possible to configure the time base clock frequency to an undivided 80 MHz by setting CTE Control Register 1 CTE Clock divider field to 0 or 1 (CTE\_CNTRL1[CTECK\_DV] = 0 or 1). If this is done and the Timing Table (TT) execution duration is set to 1 clock cycle by programming TT0/TT1 Execution Duration Register to 1 (CTE\_LUT\_DUR = 1 or CTE\_LUT\_DUR1 = 1 depending on whether TT0 or TT1 is to be executed) then the CTE will not execute the timing table and no timing signals will be generated.

**Workaround:** To execute a timing table for 1 clock cycle the CTE time base clock frequency must be less than 80 MHz. This can be achieved by setting CTE\_CNTRL1[CTECK\_DV] field to any value greater than 1 which results in a divided CTE time base clock less than 80 MHz.

#### **ERR011407: CMU: Sudden loss of clock does not signal the Fault Collection and Control Unit**

**Description:** The Clock Monitor Unit (CMU) detects when the frequency of a monitored clock drops below a programmed threshold and asserts the Frequency Less than Low Threshold (FLL) signal if this occurs. The FLL signal is routed to the Fault Collection and Control Unit (FCCU) providing a mechanism to react to the clock fault but due to the monitoring implementation the FLL signal will not be triggered when the monitored clock suddenly stops.

**Workaround:** Each of the CMU monitored clocks has been analysed for the system level failure effect upon loss of the monitored clock and the safety mechanisms present to detect this. From this analysis it is concluded that loss of all the monitored clocks can be detected by other existing safety mechanisms in the system.

Further, since the CMU monitored clocks are derived from one of the system clock sources (IRC\_CLK, XOSC\_CLK, PLL0\_PHI\_CLK, PLL1\_PHI\_CLK or SDPLL\_CLK) if the loss of the monitored clock is caused by the loss of the source clock then this will be detected and reported to FCCU by existing source clock loss detection mechanisms.

CMU\_5 is an exception because it is possible to source the monitored LFAST\_CLK from external LFAST\_REF\_CLK input instead of a system clock source. In the externally provided LFAST\_CLK use-case the loss of LFAST\_REF\_CLK can only be detected by loss of the LFAST frame transmit/receive functionality which leads to interruption in the LFAST communication.

#### **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.

### **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

### **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 lifier value.

## **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.

In case a customer does use Rx FIFO only or dedicated MB only, (i.e. either Rx MB or Rx FIFO is used) the problem doesn't occur. In case a customer does use the Enhanced Rx FIFO and dedicated MB at the same application, the problem does not occur also. So bottom line the problem does only occur 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.

**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.

© 2020 NXP B.V.

