Mask Set Errata for Mask 3N45H

This report applies to mask 3N45H for these products:

- MPC5777C

Mask Specific Information

Major mask revision number	1
Minor mask revision number	0
JTAG identifier	0x1837_701D

Table 1. Errata and Information Summary

Erratum ID	Erratum Title
e6990	CJTAG: possible incorrect TAP state machine advance during Check Packet
e10439	CMU: CMU_0 OLR is set out of reset when XOSC used in 8mhz mode, due to wrong RCDIV reset value
e7116	CRC: AutoSAR 4.0 8-bit CRC8 0x2F is not supported in hardware
e8251	DECFIL: timestamp may be lost in edge trigger mode
e9783	DSPI: Frame transfer does not restart after DSI frame matches preprogrammed value
e9664	DSPI: Frame transfer does not restart in case of DSI parity error in master mode
e9656	DSPI: Frame transfer does not restart in case of SPI parity error in master mode
e7352	DSPI: reserved bits in slave CTAR are writable
e10542	DSPI: Transmit, Command, and Receive FIFO fill flags in status register is not cleared when DMA is improperly configured
e1147	DSPI: Using DSPI in DSI mode with MTO may cause data corruption
e7001	EBI: External TA and TEA do not operate properly when internal master does burst aborts to EBI
e7546	EBI: Input signal valid to D_CLKOUT posedge (setup time) spec is not met
e9978	eMIOS: Unexpected channel flag assertion during GPIO to MCB mode transition
e8313	EQADC: TUE specification not met
e9344	eSCI: Late assertion of Transmit Data Ready Interrupt Flag (TXRDY) for Local Interconnect Network (LIN) frame receive (RX) operation
e9001	eSCI: Incorrect behavior while in LIN Standard Bit error detection mode
e9361	eSCI: Timing of TXRDY interrupt flag assertion is incorrect for LIN TX Frame
e9797	eSCI: Unable to send next frame after timeout in LIN mode

Table continues on the next page...

Table 1. Errata and Information Summary (continued)

Erratum ID	Erratum Title
e5642	ETPU2: Limitations of forced instructions executed via the debug interface
e8252	eTPU: ETPU Angle Counter (EAC) Tooth Program Register (TPR) register write may fail
e9090	eTPU: Incorrect eTPU angle counter function under certain conditions
e9809	eTPU: MDU flags(Overflow/Carry) may be set incorrectly
e9338	eTPU: Time base counter values can fail to export properly
e8042	FCCU: EOUT signals are active, even when error out signaling is disabled
e7099	FCCU: Error pin signal length is not extended when the next enabled fault, with its alarm timeout disabled, occurs
e7227	FCCU: FCCU Output Supervision Unit (FOSU) will not monitor faults enabled while already pending
e7223	FCCU: FCCU_IRQ_EN register is writeable in all operating modes
e7230	FCCU: FCCU_IRQ_EN[28] is writeable, but reserved.
e9570	FCCU: FOSU may assert reset when a hardware recoverable fault of width less than one safe clock period occurs
e7869	FCCU: FOSU monitoring of a fault is blocked for second or later occurrence of the same fault
e9670	FCCU: Limitation of error output signal observation test function
e7226	FCCU: the error-out signalling cannot be disabled in non Bi-stable protocols
e8505	FCCU: The FCCU on-chip programmable glitch filter on ERRORIN does not operate correctly
e10011	FCCU: Unexpected faults may be indicated as a result of FCCU reset
e7231	FCCU: When the configuration timeout occurs, FCCU_DELTA_T[T_TRG] is not reset to its default value
e7429	FCCU: Writes to Flash assert FCCU_NCF #17 (FLASH_INIT)
e2340	FEC: slot time is designed for 516 bit times; deviation from the 802.3
e7547	FEC: Transmit signal hold time reduced
e9320	FLASH: (SPC5777C) Address Encode False Report (MCR[AEE] and possible FCCU channels)
e8004	FLASH: Array Integrity with Breakpoints enabled may skip addresses for certain RWSC and APC combinations
e7422	FLASH: Pipeline should not be enabled on the flash
e7991	FLASH: Rapid Program or Erase Suspend fail status
e9595	FlexCAN: Corrupted frame possible if Freeze Mode or Low Power Mode are entered during a Bus-Off state
e7724	FlexCAN: Documentation of ECC registers
e9527	FlexCAN: The transmission abort mechanism may not work properly
e8195	INTC: Interrupt Controller does not work correctly when in hardware vector interrupt mode and FMPERDIV is set to divide-by-4 or divide-by-
e7545	JTAGC: TCK low to TDO data valid spec is not met
e7511	LFAST: Maximum data rate is 240Mbps
e9517	MCAN: Edge filtering may cause incorrect frame reception
e10109	MPC5777C: Enabling pending read optimization (off by default) can cause incorrect operation on the system crossbar
e9770	MPC5777C: Incorrect operation of open-drain outputs
e9856	MPC5777C: Incorrect operation of Software Watchdog Timers in debug mode

Table continues on the next page...

Table 1. Errata and Information Summary (continued)

Erratum ID	Erratum Title
e8164	MPC577xC: Current injection causes leakage path across the DSPI and LFAST LVDS pins
e10580	MPC577xC: Flash HVD may assert during slow input supply ramp
e10714	MPC577xC: Lockstep errors must be ignored when debugging core 1 in lockstep mode
e10737	MPC577xC: Nexus Program Trace Sync Message sometimes missing after queue overflow
e9332	MPC577xC: Performance degradation caused by optimization control bits on Platform Configuration Module
e10453	MPC577xC: Reduced accuracy on EQADC_B channels using VDDEH7 power domain
e9784	MPC577xC: Reset escalation count is reduced by 1 when configured via UTEST DCF record
e10578	MPC577xC: The Temperature Sensor may cause extended reset times
e10610	NPC: Core cannot be halted if NPC is not enabled when entering debug mode
e8340	NPC: EVTO_B toggles instead of remaining asserted when used by the DTS if Nexus is not enabled
e6726	NPC: MCKO clock may be gated one clock period early when MCKO frequency is programmed as SYS_CLK/8.and gating is enabled
e7120	NZxC3: DQTAG implemented as variable length field in DQM message
e9087	PADRING: Input High Voltage (max) and Hysteresis (min) specs not met
e8369	PAD_RING: Reset output (RSTOUT) pin is not driven during Power-On Reset (POR) or Low-Voltage Detect (LVD) assertion
e9250	PASS: JTAG password not working during reset
e10396	PASS: Password challenge to PASS fails while program erase ongoing in any block in memory partition 0
e7904	PASS: Programming Group Lock bit (PGL) can be de-asserted by multiple masters writing the correct password sections to the CINn registers.
e10226	PMC: In SMPS mode, during input supply power down, core voltage can exceed specification limits
e9804	PMC: LVD/HVD Event Status Register does not consistently indicate the reset source
e10135	PMC: Resets that occur during the PMC Self Test process can cause corrupted results.
e5887	PSI5: Detection of a received bit causes an electrical error in specific conditions
e9863	PSI5: Enabling interrupts in the General Interrupt Control Register (PSI5_CH0_GICR) has no effect
e6992	PSI5: IS_DEBUG_FREEZE bit is not documented
e7234	PSI5: No transfer error generated for accesses within the unused range of the PSI5 peripheral window
e5073	PSI5: Possible message reception errors due to incorrect data latency reference point
e6553	PSI5: T bit error Ambiguity is noticed in Synchronous mode
e8368	REACM2: ETPU_C clock does not halt if REACM2 Module Disable (MDIS) is set to 1
e8367	REACM: Register is unexpectedly written after exiting halted state
e10415	SDADC: Additional DMA request generated with slow SDADC SD_CLK clock
e8711	SDADC: Digital filter and FIFO not disabled when MCR[EN] is cleared
e8225	SDADC: FIFO Flush Reset command requires clearing the Data FIFO Full Flag
e10378	SDADC: Incorrect data provided when FIFO is disabled and FIFO overwrite is enabled
e6906	SDADC: Invalid conversion data when output settling delay value is less than 23
e8631	SDADC: low threshold watchdog cannot be used with signed data
e7356	SDADC: The SDADC FIFO does not function correctly when FIFO overwrite option is used

Table continues on the next page...

Table 1. Errata and Information Summary (continued)

Erratum ID	Erratum Title
e8710	SDADC: Watchdog Crossover event missed if FM Peripheral Clock frequency is less than or equal to the sigma-delta ADC clock frequency
e7204	SENT: Number of Expected Edges Error status flag spuriously set when operating with Option 1 of the Successive Calibration Check method
e8082	SENT: A message overflow can lead to a loss of frames combined with NUM_EDGES_ERR being set
e7425	SENT: Unexpected NUM_EDGES_ERR error in certain conditions when message has a pause pulse
e10645	SIU: WKPCFG is not applied until after initial reset negation
e9658	SPI: Inconsistent loading of shift register data into the receive FIFO following an overflow event
e7431	STCU2: LBIST does not accept programmable seed value
e7339	STCU2: STCU2 fault injected by FCCU is self clearing
e10636	STCU: Improper behavior in some cases at hot temperatures during offline LBIST and MBIST operation.
e10443	STCU: Improper behavior in some cases when external reset is asserted during LBIST execution
e10025	SWT: System clock source must be set to IRC prior to changing SWT clock selection
e9336	TDM: Erase of TDR flash block may be blocked by the TDM - CSE systems
e7236	XBIC: XBIC may trigger false FCCU alarm
e8310	XBIC: Crossbar Integrity Checker may miss recording information from an initial fault event in the case of back-to-back faults
e8730	XBIC: XBIC may store incorrect fault information when a fault occurs

Table 2. Revision History

Revision	Changes
March 2016	Initial revision
July 2016	The following errata were removed. • e7544 The following errata were added. • e10226 • e10135 • e2340 • e10378 • e10443 • e10453 The following errata were revised. • e8195
August 2016	The following errata were revised. • e10453 • e9595
February 2017	The following errata were removed.

Table 2. Revision History

Revision	Changes
	- e3521 The following errata were added. - e10415 - e10737 - e10396 - e10542 - e10645 - e10439 - e10578 - e10636 - e10580 - e10610 - e10714 The following errata were revised. - e10453

e6990: CJTAG: possible incorrect TAP state machine advance during Check Packet

Description: While processing a Check Packet, the IEEE 1149.7 module (CJTAG) internally gates the TCK clock to the CJTAG Test Access Port (TAP) controller in order to hold the TAP controller in the Run-Test-Idle state until the Check Packet completes. A glitch on the internally gated TCK could occur during the transition from the Preamble element to the first Body element of Check Packet processing that would cause the CJTAG TAP controller to change states instead of remaining held in Run-Test-Idle

If the CJTAG TAP controller changes states during the Check Packet due to the clock glitch, the CJTAG will lose synchronization with the external tool, preventing further communication.

Workaround: To prevent the possible loss of JTAG synchronization, when processing a Check Packet, provide a logic 0 value on the TMS pin during the Preamble element to avoid a possible glitch on the internally gated TCK clock.

e10439: CMU: CMU_0 OLR is set out of reset when XOSC used in 8mhz mode, due to wrong RCDIV reset value

Description: Clock Monitor Unit 0 (CMU_0) asserts “Oscillator frequency less than reference” (OLR) event coming out of reset when the crystal oscillator (XOSC) is used in 8MHz mode, resulting in Fault Collection and Control Unit (FCCU) fault NCF 25 being set. This is because the reset value of the Reference Clock Divider CMU_0_CSR[RCDIV] = 0x0 (instead of 0x3), resulting in XOSC clk (8MHz) < IRC (16MHz).

Workaround: Before enabling FCCU NCF 25 event, clear OLRI bit by writing CMU_0_CSR[RCDIV] to 0x3 first and then CMU_ISR[OLRI] to 1 (the bit is write-1-to-clear).

e7116: CRC: AutoSAR 4.0 8-bit CRC8 0x2F is not supported in hardware

Description: The Cyclic Redundancy Check (CRC) module does not implement the 8-bit CRC-8-H2F required to support the Autosar 4.0 specification. The CRC-8-H2F uses a polynomial generator seed of 0x2F and an equation of $x^5 + x^3 + x^2$+x+1.

Workaround: Do not set the Polynomial selection to 0b11 in the CRC Configuration register (CRC_CFG). The 8-bit CRC-8-H2F function must be written in software to support AuroSAR 4.0.

e8251: DECFIL: timestamp may be lost in edge trigger mode

Description: The Enhanced Queued Analog-to-Digital Converter (eQADC) supports a conversion command that configures it to send a timestamp along with the specified ADC conversion data to the Decimation Filter (DECFIL) for digital processing. The DECFIL receives the data and the timestamp, and updates internal registers with these two values. When the DECFIL is configured for edge triggered output by setting the Triggered Output Result Enable bit in the Module Configuration Register (DECFIL_MCR[TORE]) and setting the Trigger Mode bitfield to either 2b00 or 2b10, and a trigger edge is detected, the DECFILT loads an Internal Output Buffer register (DECFILT_IOB) with the conversion data, and then the timestamp data. This register is intended to hold data to be returned on one of the two Parallel Side Interfaces (PSI0 or PSI1). In the case where a trigger edge occurs and DECFILT_IOB is loaded with the conversion and timestamp data, and then a second trigger edge occurs before any new conversion and timestamp data has been received by the DECFILT, the DECFILT will provide only the initial conversion data, and will not provide the initial timestamp data.

The level triggered mode is not affected by this issue.

Workaround: When the DECFILT has been configured for edge triggered output buffer mode, ensure that the trigger edge rate is slower than the input data rate of the decimation filter. That is, be sure that there is always a new conversion arriving at the DECFILT before any new output trigger edge is detected. If the DECFILT is not receiving timestamps from the eQADC, this limitation is not required.

e9783: DSPI: Frame transfer does not restart after DSI frame matches preprogrammed value

Description: In the Deserial Serial Peripheral Interface module, in the scenario when:

1. Master/slave mode select bit of module configuration register is set (MCR[MSTR]=0b1) to configure the module in master mode

2. Deserial Serial Interface (DSI) communication is selected via DSPI Configuration field (DCONF) in the Module Configuration Register (MCR [DCONF] = 0b01)

3. Preprogrammed value for data match with received DSI frame is configured using DSI De-serialized Data Polarity Interrupt Register (DPIR) and DSI De-serialized Data Interrupt Mask Register (DIMR)

4. Data Match Stop (DMS) bit of DSI configuration register0 is set (DSICR0 [DMS] =0b1) which stops DSI frame transfer in case of a data match with a preprogrammed value

5. DSI frame is received with bits matching preprogrammed value.

Under these conditions, the next frame transfer is stopped, DSI Data Received with Active Bits bit of status register is set (SR [DDIF] =0b1) and the corresponding DDIF interrupt is asserted. Even after the interrupt is serviced and SR [DDIF] is reset, the frame transfer does not restart.

Workaround: DSI frame transfer stop in case of DSI data match condition should be disabled. For this, keep the data match stop bit of DSI configuration register 0 de-asserted (DSICR0 [DMS]=0b0).

e9664: DSPI: Frame transfer does not restart in case of DSI parity error in master mode

Description: In the Serial Peripheral Interface module, in the scenario when:

1. Master/slave mode select bit of module configuration register is set (MCR[MSTR]=0b1) to configure the module in master mode

2. Deserial Serial Interface (DSI) communication is selected via DSPI Configuration field (DCONF) in MCR (MCR[DCONF] = 0b01)

3. Parity reception check on received DSI frame is enabled by setting Parity Enable bit (PE) of DSI configuration register 0 (DSICR0[PE]=0b1)

4. Parity Error Stop (PES) bit of DSI configuration register0 is set (DSICR0[PES]=0b1) which stops DSI frame transfer in case of parity error

5. Parity error is detected on received frame

Then the next frame transfer is stopped, DSI parity error flag bit of status register is set (SR[DPEF] =0b1) and the corresponding DSI parity error interrupt is asserted. Even after the interrupt is serviced and SR [DPEF] is reset, the frame transfer does not restart.

Workaround: DSI frame transfer stop in case of parity error detection should be disabled. For this, keep the parity error stop bit of DSI configuration register0 de-asserted (DSICR0 [PES]=0b0).

e9656: DSPI: Frame transfer does not restart in case of SPI parity error in master mode

Description: In the Deserial Serial Peripheral Interface (DSPI) module, in the scenario when:

1. Master/slave mode select bit (MTSR) of Module Configuration register (MCR) is set (MCR[MSTR]=0b1) to configure the module in master mode

2. SPI communication is selected via DSPI Configuration field (DCONF) in MCR (MCR[DCONF] = 0b00)

3. Parity reception check on received frame is enabled by setting the Parity Enable or Mask tASC delay (PE_MASC) bit of DSPI PUSH FIFO Register In Master Mode (PUSHR), i.e. PUSHR[PE]=0b1.

4. Parity Error Stop bit (PES) of MCR is set (MCR[PES]=0b1) which stops SPI frame transfer in case of parity error

5. Parity error is detected on received frame.

Then the next frame transfer is stopped, the SPI Parity Error Flag bit (SPEF) of the DSPI Status Register (DSPI_SR) is set (SR[SPEF] =0b1) and the corresponding SPI parity error interrupt is asserted. Even after the interrupt is serviced and SR[SPEF] is reset, the frame transfer does not restart.

Workaround: Do not use SPI frame transfer stop in case of parity error detection for SPI transmission in master mode. For this, keep the Parity Error Stop bit of Module Configuration Register de-asserted (MCR[PES] = 0b0).

e7352: DSPI: reserved bits in slave CTAR are writable

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

Workaround: There are two possible workarounds.

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

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

e10542: DSPI: Transmit, Command, and Receive FIFO fill flags in status register is not cleared when DMA is improperly configured

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

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

Use the DMA loop to transfer more data if needed.

e1147: DSPI: Using DSPI in DSI mode with MTO may cause data corruption

Description: Using the DSPI in Deserial Serial Interface (DSI) Configuration (DSPIx_MCR[DCONF]=0b01]) with multiple transfer operation (DSPIx_DSICR[MTOE=1]) enabled, may cause corruption of data transmitted out on the DSPI master if the clock Phase is set for leading edge capture DSPIx_CTARn[CPHA]=0. The first bit shifted out of the master DSPI into the slave DSPI module will be corrupted and will convert a ‘0’ to read as a ‘1’.

Workaround: There are three possible workarounds for this issue.

1) Select CPHA=1 if suitable for external slave devices.

2) Set first bit to ‘1’, or ignore first bit. This may not be a workable solution if this bit is required.

3) Connect SOUT from the master to SIN of the first slave externally instead of using internal signals. This is achieved by setting the DSPI Input Select Register (SIU_DISR) to set the SINSELx field of the first slave DSPI to ‘00’ and configuring this slave’s SIN pin and master SOUT pin as DSPI SIN/SOUT functions respectively. This workaround is suitable only if these two signals are available to be connected externally to each other.

e7001: EBI: External TA and TEA do not operate properly when internal master does burst aborts to EBI

Description: Internal masters such as the e200 cores (with cache enabled), the Enhanced Direct Memory Access modules (eDMAs), and the Fast Ethernet Controller (FEC), can, under certain circumstances, abort a burst access in progress. If this occurs on an EBI access that is using external Transfer Acknowledge (TA) or Transfer Error Acknowledge (TEA), it can cause protocol violations and possibly hang the external bus.

Workaround: Do not use external TA, the TEA pin, or non-chip-select accesses with the e200 cores (with cache enabled for EBI address space), the eDMAs, or the FEC. External TA can be disabled for chip-select accesses by clearing the Select External Transfer Acknowledge (SETA) bit in the corresponding EBI Calibration Base Register (EBI_CAL_BRn) for that chip select.

e7546: EBI: Input signal valid to D_CLKOUT posedge (setup time) spec is not met

Description: The External Bus Interface (EBI) specification for input signal valid to D_CLKOUT posedge (setup time) was originally 5ns. This device requires 7ns.

Workaround: Ensure external devices connected via EBI provide the required input signal setup time.

e9978: eMIOS: Unexpected channel flag assertion during GPIO to MCB mode transition

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

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

(1) Clear the FLAG enable bit in the eMIOS Control register (eMIOS_Cn[FEN] = 0).

(2) Change the channel mode (eMIOS_Cn[MODE]) to the desired MCB mode.

(3) Clear the channel FLAG bit by writing ‘1’ to the eMIOS Channel Status register FLAG field (eMIOS_Sn[FLAG] = 1).

(4) Set the FLAG enable bit (eMIOS_Cn[FEN] = 1) to re-enable the channel interrupt or DMA request reaction.

e8313: EQADC: TUE specification not met

Description: The Enhanced Queued Analog-to-Digital Converter (EQADC) Total Unadjusted Error (TUE) specification in the device datasheet lists separate error levels for ADC clock of 16.5MHz and 33MHz, respectively. Neither the specification of +/- 4 counts at 16.5MHz, nor the specification of +/- 6 counts at 33MHz is met. Specifications for “Gain with Calibration” (GAINWC) and “Offset with Calibration” (OFFWC), which contribute to TUE, are also not met.

Regardless of ADC clock speed, the TUE, GAINWC, and OFFWC specifications are each +/- 8 counts. A future release of the device datasheet will reflect the new specification level.

Workaround: Ensure that application use-cases can tolerate the updated error specifications. If greater accuracy is required, some improvement can be gained by performing multiple conversions and averaging results to reduce noise-related errors.

e9344: eSCI: Late assertion of Transmit Data Ready Interrupt Flag (TXRDY) for Local Interconnect Network (LIN) frame receive (RX) operation

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

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

e9001: eSCI: Incorrect behavior while in LIN Standard Bit error detection mode

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

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

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

e9361: eSCI: Timing of TXRDY interrupt flag assertion is incorrect for LIN TX Frame

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

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

e9797: eSCI: Unable to send next frame after timeout in LIN mode

Description: When generating a Local Interconnect Network (LIN) Transmit (Tx) and Receiver (Rx) frame, the Enhanced Serial Communication Interface (eSCI) module should first send the Header as per the LIN protocol. However, after the Slave Timeout Interrupt Flag (STO) in the Interrupt Flag and Status Register 2 (eSCI_IFSR2) for the previous LIN Rx Frame is asserted (eSCI_IFSR2[STO]=1), the eSCI module is not able to generate the next Header, it remains in a suspended state.

Workaround: Perform the following actions in this order:

(1) Set the LIN Protocol Engine Stop and Reset (LRES) control bit to ‘1’ in the LIN Control Register 1 (eSCI_LCR1).

(2) Wait until the status bits DACT, WACT, LACT, TACT,and RACT in the Interrupt Flag and Status Register (eSCI_IFSR1) are cleared to ‘0’.

(3) Clear LRES in eSCI_LCR1 to ‘0’.

(4) Begin transmission of the LIN Header for the next frame.

e5642: ETPU2: Limitations of forced instructions executed via the debug interface

Description: The following limitations apply to forced instructions executed through the Nexus debug interface on the Enhanced Time Processing Unit (ETPU):

1- When a branch or dispatch call instruction with the pipeline flush enabled (field FLS=0) is forced (through the debug port), the Return Address Register (RAR) is updated with the current program counter (PC) value, instead of PC value + 1.

2- The Channel Interrupt and Data Transfer Requests (CIRC) instruction field is not operational.

Workaround: Workaround for limitation #1 (branch or dispatch call instruction):

Increment the PC value stored in the RAR by executing a forced Arithmetic Logic Unit (ALU) instruction after the execution of the branch or dispatch call instruction.

Workaround for limitation #2 (CIRC):

To force an interrupt or DMA request from the debugger:

1- Program a Shared Code Memory (SCM) location with an instruction that issues the interrupt and/or DMA request. Note: Save the original value at the SCM location.

2- Save the address of the next instruction to be executed.

3- Force a jump with flush to the instruction position.

4- Single-step the execution.

5- Restore the saved value to the SCM location (saved in step 1).

6- Force a jump with flush to the address of the next instruction to be executed (saved in step 2).

NOTE: This workaround cannot be executed when the eTPU is in HALT_IDLE state.

e8252: eTPU: ETPU Angle Counter (EAC) Tooth Program Register (TPR) register write may fail

Description: When the TPR is written with the Insert Physical Tooth (IPH) bit set to 1, and a physical tooth arrives at near the same time, the buffering of a second write to the TPR may fail, even if the required wait for one microcycle after the IPH write is observed.

Workaround: Wait at least two microcycles between consecutive writes to the TPR register, if the first write sets the IPH bit.

e9090: eTPU: Incorrect eTPU angle counter function under certain conditions

Description: The eTPU Angle Counter (EAC) can function incorrectly in some scenarios when all of the following conditions apply:

- EAC Tooth Program Register (TPR), Angle Ticks Number in the Current Tooth field (TICKS) = 0 [TPR.TICKS = 0]

and

- Tick Rate Register (TRR) and the eTPU Engine Time Base Configuration Register prescaler field [eTPU_TBR_TBCR_ENGn.TCRnP] satisfy the following condition:

(TRR – 1)*(TCRnP + 1) < 3, where TRR is the non-zero 15-bit integer part (the 15 most significant bits).

When the above conditions are met, three possible scenarios can cause the EAC to function incorrectly:

Scenario 1:

1. The EAC is in High Rate Mode, TRR = 1, and TPR Missing Tooth Counter field = 0 [TPR.MISSCNT = 0]

2. On an EAC transition from High Rate Mode to Normal mode, a positive value is written to TPR.MISSCNT

3. The first microcycle in Normal Mode coincides with a tick timing and either

a. A tooth does not arrive

or

b. A tooth arrives

Expected EAC behavior:

a. Nothing happens

or

b. The EAC transitions back to High Rate Mode

Actual (incorrect) EAC behavior:

a. The EAC transitions to Halt Mode, even though TPR.MISSCNT > 0

or

b. The EAC stays in Normal Mode, even though a tooth arrived before expected and TPR.MISSCNT > 0. The values of TPR.MISSCNT and TPR.LAST are reset, even though the EAC does not transition to High Rate Mode.

Scenario 2:

TCRnP = 0, TRR = 1 (integer part) and a new value is written to TPR.MISSCNT when the EAC transitions from High Rate Mode to Normal Mode. In this scenario, TPR.MISSCNT decrements on every microcycle, but the time the EAC takes to transition to Halt Mode is determined by the previous

TPR.MISSCNT value, so that one of the following unique situations is observed:

a. TPR.MISSCNT reaches zero, but the EAC transitions to Halt Mode only after a number of microcycles equal to the TPR.MISSCNT value before the write.

b. EAC transitions to Halt Mode with TPR.MISSCNT > 0 while, decrementing MISSCNT one more time. If TPR.MISSCNT > 1 during the mode transition, the EAC will stay in Halt mode with a non-zero value of TPR.MISSCNT.

Scenario 3:

1. The EAC transitions to Normal mode from High Rate or Halt Mode

2. The EAC enters Normal mode with TPR.LAST = 1

3. A tooth is received on the second or third microcycle after the EAC transitions to Normal mode. The tooth may be either a physical tooth or a dummy physical tooth generated by setting the Insert Physical Tooth (IPH) field of the TPR register (TPR.IPH = 1).

Observed result:

The EAC resets the values of TPR.LAST, TPR.IPH and the eTPU Engine Time Base2 (TCR2) register, but the EAC goes to Halt mode.

If a new TPR.TICKS value is written with the EAC in Normal mode, the value is effective after a new tooth is received in Halt mode, with TCR2 counting from 0.

Workaround: Limit the angle tick period to a minimum value that satisfies the condition (TRR – 1)* (TCRnP + 1) > 2, where TRR is the non-zero 15-bit integer part (the 15 most significant bits).

e9809: eTPU: MDU flags(Overflow/Carry) may be set incorrectly

Description: The MAC Carry (MC) & MAC Overflow (MV) flags can be incorrectly set on a MAC instruction if it is the first MDU operation in a thread and the last MDU operation in previous thread was aborted/terminated (thread ended before the operation finished).

Workaround: There are 2 workarounds:

(1) Do not abort/terminate a MDU operation

or

(2) Do not use a MAC instruction as the first MDU operation in a thread

e9338: eTPU: Time base counter values can fail to export properly

Description: Time base counters from the Enhanced Time Processor Unit (eTPU) may sometimes fail to export on the Shared Time and Angle Counter (STAC) bus to other eTPU engines or to the Enhanced Modular Input Output Subsystem (eMIOS). This issue can occur when the clock ratio between eTPU and eMIOS is switched from 1:1 to 2:1 (in other words, when the eTPU operates at twice the frequency of the eMIOS).

Workaround: The reset default clock ratio for eTPU:eMIOS is 1:1. If the application requires switching to 2:1 clock ratio, perform this clock ratio switch early in the application initialization procedure while the system clock is still being driven by the Internal RC Oscillator (IRCOSC) and before any other changes are made to any Phased Lock Loop (PLL) or System Integration Unit (SIU) system clock tree divider settings.

Perform a single write to the SIU System Clock Register eTPU Divider (SIU_SYSDIV[ETPUDIV] = 1). This switches the clock ratio from 1:1 to 2:1 in such a way that the STAC bus logic can properly register the change in clock ratios. Once this change has been made, ensure that no other subsequent writes to the SIU_SYSDIV register change the value of the ETPUDIV field from 1 back to 0 until the next reset event. Writes to other fields in SIU_SYSDIV are allowed and will not affect this workaround.

e8042: FCCU: EOUT signals are active, even when error out signaling is disabled

Description: Every time the Fault Collection and Control Unit (FCCU) moves into fault state caused by an input fault for which the error out reaction is disabled (FCCU_EOUT_SIG_ENn[EOUTENx]=0), the Error Out 1 and 2 (EOUT[0] and EOUT[1]) will become active for a duration of 250 us plus the value programmed into the FCCU Delta Time register (FCCU_DELTA_T[DELTA_T]). EOUT is not affected if the FCCU moves into the alarm state that generates an interrupt (IRQ), if the Fault is cleared before the alarm timeout.

This erratum does not affect the outputs of other pins (for example, for communication modules like CAN/Flexray). Only the EOUT signal is impacted.

Workaround: There are three possible workarounds:

1) Enable EOUT signaling for all enabled error sources.

2) In case external device (which evaluates EOUT) can communicate with the MCU, the following procedure could be used:

a) Program any duration of EOUT as per application needs (FCCU_DELTA_T[DELTA_T])

b) For faults requiring error out reaction, the software shall validate EOUT via separate communication channel (like I2C) while EOUT is asserted.

c) External device shall implement a timeout mechanism to monitor EOUT validation by separate channel.

d) Following scenarios shall be considered as valid EOUT reactions:

d1) Validation is performed while EOUT is asserted d2) Timeout occurs but no validation and EOUT is still asserted.

3) In case external device (which evaluates EOUT) cannot communicate with the MCU, following procedure could be used:

a) Program the error out duration to a duration x (FCCU_DELTA_T[DELTA_T]).

b) For faults requiring error out reaction, clear the fault after the pin has continued to be asserted for a longer duration (for example 2*duration x). This will artificially create a long pulse on EOUT.

c) For faults which do not require error out reaction, clear the fault within duration x. This will artificially create a short pulse on EOUT.

d) External device should ignore short pulse of duration x while recognizing longer pulses as valid reaction.

e) While clearing the fault, the associated software shall check the pending faults.

e7099: FCCU: Error pin signal length is not extended when the next enabled fault, with its alarm timeout disabled, occurs

Description: In the Fault Collection and Control Unit (FCCU), when the following conditions are met:

- two faults occur
- the second fault arrises with a delay (T_delay) from the first error
- the second fault has its alarm timeout disabled
- T_delay is lower than the FCCU error pin minimum active time (T_min, defined in the Delta T register (FCCU_DELTA_T))

Then the error output signal is not extended and its duration is only T_min, if the faults are cleared before the timer expires.

The expected behavior is to have the error output signal duration of T_min + T_delay, if the faults are cleared before the timer expires.

Workaround: Take into account that the error out signal duration will only be T_min, if the faults are cleared before the timer expires.

The timer count is meaningful only when the Error pin is driven low, which can be checked by reading the pin status FCCU_STAT[ESTAT].

e7227: FCCU: FCCU Output Supervision Unit (FOSU) will not monitor faults enabled while already pending

Description: The Fault Collection and Control Unit (FCCU) Output Supervision Unit (FOSU) will not monitor the FCCU reaction to fault inputs that are enabled with an already pending notification.

The FOSU monitoring is triggered by an edge from a fault input. The edge detection will be blocked in following cases:

1) When a fault input is disabled in the FCCU and a fault occurs,

2) When a fault input is enabled in the FCCU and a fault occurs in the CONFIG state. FOSU edge detection remains blocked until it gets initialized by a FCCU reaction or a destructive reset.

Workaround: Apply the following procedure when enabling fault inputs in the FCCU in order to ensure correct monitoring by the FOSU:

1) Check for FCCU pending faults and clear them.

2) Configure the FCCU as desired. In addition enable fault input for interrupt reaction (software recovery mode) to an injected error on this input.

3) Immediately on exiting the CONFIG state, check for FCCU pending faults. If there is a fault status set then initiate a destructive reset.

4) Clear the FOSU status by injecting a fault on the FCCU fault input configured for software recovery mode. This will generate a FCCU reaction that will clear the FOSU edge detection logic.

Apply the following procedure when exiting FCCU CONFIG state in order to ensure correct monitoring by the FOSU:

- Check for FCCU pending faults. If there is a fault status set then initiate a destructive reset.

e7223: FCCU: FCCU_IRQ_EN register is writeable in all operating modes

Description: In the Fault Collection and Control Unit (FCCU), the FCCU Interrupt Enable register (FCCU_IRQ_EN) is writable (and readable) in all states (NORMAL, CONFIG, FAULT and ALARM) while in some revisions of the documentation it is stated “This register is writable only in the CONFIG state”.

Workaround: Take into account that FCCU_IRQ_EN register can be written in all the states of the FCCU.

Please ignore the following text in the description of FCCU_IRQ_EN register if you find it in the documentation revision in hand:

"This register is writable only in the CONFIG state."

e7230: FCCU: FCCU_IRQ_EN[28] is writeable, but reserved.

Description: The Fault Collection and Control Unit (FCCU) Interrupt (IRQ) Enable Register (FCCU_IRQ_EN) bit position 28 is writable, even though the bit is documented as reserved.

Workaround: Expect the FCCU_IRQ_EN[28] bit to be writable, not reserved. For future compatibility software should set this bit to the reset value of 0b0 any time this register is written.

e9570: FCCU: FOSU may assert reset when a hardware recoverable fault of width less than one safe clock period occurs

Description: The Fault Collection and Control Unit Output Supervision Unit (FOSU) may issue a reset if all of the following conditions are present:

• An input fault is programmed as hardware recoverable in a FCCU Non-Critical Fault Configuration Register (FCCU_NCF_CFGn)

• The only reaction programmed for this fault is FCCU Error Output signaling (FCCU_EOUT_SIG_ENn)

- The source of the fault signal is asserted for less than one safe clock period. The safe clock for this device is the internal RC oscillator (IRC).

Workaround: (1) Program “Request reset pulse” in the associated FCCU Non-Critical Fault State Configuration (FCCU_NCFS_CFGn) register for that hardware recoverable fault in addition to the EOUT signaling.

(2) Alternatively, if the reset reaction is not desired, configure the fault as software recoverable.

e7869: 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: There are two possible workarounds. Either one can be used with same effectiveness.

1. In addition to the reset reaction, enable either the interrupt (IRQ) or Non-maskable Interrupt (NMI) or error out signaling reaction for the faults that have a reset reaction enabled.

2. Apply the following procedure during the FCCU configuration after a reset and in the fault service routine while clearing the fault status inside the FCCU.

i. Check for FCCU pending faults and clear them.

ii. Configure the FCCU as desired.

iii. Enable a fault as software recoverable by setting its corresponding bit in the NCF Configuration Register (FCCU_NCF_CFGn)

iv. Inject a fake fault to the fault set up in step “iii” by writing the corresponding code into the NCF Fake Register (FCCU_NCFF)

v. Check that there are no pending faults else clear the pending faults and repeat steps “iv” and “v”

vi. Reconfigure the fault that was configured for software recovery mode.

e9670: FCCU: Limitation of error output signal observation test function

Description: The Fault Collection and Control Unit (FCCU) I/O Control Register (FCCU_EINOUT) provides an option to observe the EOUT[1:0] signals (ERROR1 and ERROR0 pins) in input mode in either the Test0 or Test2 fault output modes.

The ERROR[1:0] signals on this device are available on multi-function pins whose function is meant to be selectable either by Device Configuration Format (DCF) record or by System Integration Unit (SIU) registers. However, due to this issue, the observation of error signals provided by FCCU_EINOUT only functions properly when the pins have been configured by DCF record and does not function correctly if the pin function has been selected via the SIU Pad Configuration Register (PCR) instead.

Workaround: In order to use the FCCU_EINOUT features to properly observe the error signals in input mode, the Error Functions Pads Configuration DCF record must be programmed in the user DCF records section of user test (UTEST) flash. Do not use Pad Configuration Registers SIU_PCR117 or SIU_PCR125 to select the error pin functions.

e7226: FCCU: the error-out signalling cannot be disabled in non Bi-stable protocols

Description: In the Fault Collection and Control Unit (FCCU) module, the error out signalling can only be disabled when using the Bi-stable protocol and not all of the protocols.

The Error Out (EOUT) Signaling Enable Register (FCCU_EOUT_SIG_ENx) registers, which can be used for disabling error out signalling, are applicable only for bi-stable mode and do not affect the other protocols. (Various protocols like Bi-stable, Dual rail and other protocols are selected by programming the Fault Output Mode (FOM) field of the Configuration Register (FCCU_CFG)).

Workaround: Do not expect and program the EOUT_SIGN_ENx register to support anything other than bi-stable protocol.

The EOUT Signaling Enable (EOUTENx) field description of FCCU_EOUT_SIG_EN should be read as follows:

0 = EOUT signaling is disabled for error Recoverable Fault (RF) source x for bi-stable protocol. Error pins behave as if in Non-Faulty state.

1 = EOUT signaling is enabled for error RF source x for bi-stable protocol.

e8505: FCCU: The FCCU on-chip programmable glitch filter on ERRORIN does not operate correctly

Description: The Fault Collection and Control Unit (FCCU) external error input (ERRORIN) can be filtered by the FCCU on-chip programmable glitch filter but is routed directly to the FCCU Output Supervision Unit (FOSU) without passing through the glitch filter. In addition, when the glitch filter is programmed using the FCCU Control Register (FCCU_CTRL) FILTER_WIDTH field, the effective duration may vary depending on the ERRORIN signal arrival instant.

As a result of these issues, it is possible for the FOSU to recognize an event on ERRORIN that is ignored by the FCCU resulting in a reset when the FOSU timeout period expires.

Workaround: Bypass the FCCU on-chip glitch filter by writing a 1 to FCCU Control Register FILTER_BYPESS field (FCCU_CTRL[FILTER_BYPASS]). Use an external glitch filter instead of the on-chip filter to ensure that only valid external error events are recognized by the FCCU.

e10011: FCCU: Unexpected faults may be indicated as a result of FCCU reset

Description: When a Fault Collection and Control Unit (FCCU) reset occurs (as a result of a fault that is configured for reset reaction), there may be additional faults incorrectly recorded in the FCCU status registers.

Workaround: Write software such that it does not depend on the FCCU status registers to contain only real faults after reset. Ignore interrupt-reaction faults when reading FCCU status registers after an FCCU reset. If multiple reset-reaction faults are recorded, be aware that likely only one is a valid fault.

e7231: FCCU: When the configuration timeout occurs, FCCU_DELTA_T[T_TRG] is not reset to its default value

Description: The Fault Collection and Control Unit (FCCU) must be configured while in the CONFIG state using the FCCU Control (FCCU_CTRL) register. To prevent unwanted modifications to the FCCU configuration, the CONFIG state is protected with a watchdog that only allows modification to the FCCU during a configurable time window. This window is configured on the Configuration Timeout Register (FCCU_CFG_TO) register.

In the case of a timeout, the FCCU will exit CONFIG mode and return to NORMAL mode. All the registers that were being modified return to their default values.

The T_TRG bitfield on the FCCU_DELTA_T register presents the issue as it will not return to its default value even if the transition from CONFIG to NORMAL stage was caused by a timeout. The T_TRG bitfield is used to configure the EOUT pin semi-period that is used in case the FCCU goes in FAULT mode.

Incorrect T_TRG bitfield configuration may cause the EOUT pin to toggle too slowly or too fast. This may cause safety problems if the external safety circuit connected to the EOUT pin does not receive the error signal in the expected frequency.

Workaround: Expect T_TRG bitfield to retain the programmed value even if the transition from CONFIG to NORMAL was caused by a configuration timeout.

e7429: FCCU: Writes to Flash assert FCCU_NCF #17 (FLASH_INIT)

Description: When programming Flash, Fault Control and Collection Unit event #17, “Flash Initiation Error” may be incorrectly asserted when no actual error has occurred.

Workaround: By default there is no reaction to this event. The user must not program a reaction for this fault channel when flash programming is required. It is intended only for debug purposes.

e2340: FEC: slot time is designed for 516 bit times; deviation from the 802.3

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

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

Workaround: No software workaround is needed or available.

e7547: FEC: Transmit signal hold time reduced

Description: The Fast Ethernet Controller (FEC) Media Independent Interface (MII) transmit signal hold time specification is reduced from 5ns to 4.5ns on this device.

Workaround: Ensure Ethernet physical interface (PHY) devices connected via MII are able to tolerate the reduced hold time specification.

e9320: FLASH: (SPC5777C) Address Encode False Report (MCR[AEE] and possible FCCU channels)

Description: During Flash Read while Write operations (RWW), it is possible for a Program or Erase operation to corrupt the Address Encode feature of the flash, and falsely give an Address Encode Error (AEE) event. The false AEE event only occurs for RWW operations to partitions in the Low, Mid or High address spaces, and may only occur if the read and write occur both in Even RWW partitions (i.e. 0, 2, or 4), or if the read and write occur both in Odd RWW partitions (i.e. 1, 3, or 5).

Reads to even numbered RWW partitions while writing to odd numbered RWW partitions will not trigger this false AEE condition. Likewise, reads to odd numbered RWW partitions while writing to even numbered RWW partitions will not trigger this false AEE condition.

Reads and Writes to 256K blocks, do not show the address encode corruption issue.

Read Data and ECC Parity bits returned for these Reads while writing are valid and not corrupted.

Workaround: Disable the Fault Collection and Control Unit (FCCU) Failure input channels 31 and 32 by clearing the corresponding bits in the Non-Recoverable Fault Enable Registers 0 and 1 (FCCU_NCF_E0 and FCCU_NCF_E1). FCCU input channel 31 is the Encoding Error Flash, and covers the C55FMC_MCR[AEE] event indication in addition to Flash Read Voltage Error (C55FMC_MCR[RVE]) and Flash Read Reference Error (C55FMC_MCR[RRE]) indications and is controlled by the FCCU_NCF_E0[bit 0]. FCCU input channel 32 is the flash controller address feedback event indication and is controlled by the FCCU_NCF_E1[bit 31].

Address Encode Error (AEE) fault recognition is a safety mechanism used to detect potential permanent and transient faults in the flash address decode logic. Even with AEE detection disabled, most faults that would be detected by AEE are still detected by other mechanisms as part of the MPC57xxM redundant safety concept. Consequently, disabling the AEE fault notifications in the FCCU has no impact to the overall functional safety integrity of the device, and the ISO26262 ASIL D target is achieved.

With the FCCU channels 31 and 32 disabled, the Read Voltage Error and Read Reference error may be monitored by reading the fields from the Flash Module Configuration register (C55FMC_MCR[RVE], C55FMC_MCR[RRE]).

e8004: 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.

e7422: FLASH: Pipeline should not be enabled on the flash

Description: Enabling of pipelining of the Flash Address Pipeline Control (APC) bits in the Platform Flash Configuration Register 1 (PFLASH_PFCR1) does not give optimum system performance. Pipelining the accesses will actually decrease the performance of the flash when there are multiple flash masters. The APC should be set to a value of 0x4.

Workaround: Disable pipelined flash accesses by properly setting the Address Pipeline Control bits in the Platform Flash Configuration Register 1 to Pipelined accesses disabled and one wait state (PFLASH_PFCR1[APC] = 0x4).

e7991: 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.

e9595: FlexCAN: Corrupted frame possible if Freeze Mode or Low Power Mode are entered during a Bus-Off state

Description: In the Flexible Controller Area Network (FlexCAN) module, if the Freeze Enable bit (FRZ) of the Module Configuration Register (MCR) is asserted and the Freeze Mode is requested by asserting the Halt bit (HALT) of the MCR register during the Bus Off state, the transmission after exiting the Bus-Off condition will be corrupted. The issue occurs only if a transmission is pending before the freeze mode request. In addition, the same issue can happen if Low-Power Mode is requested instead of Freeze Mode.

Workaround: The workaround depends on whether the bus-off condition occurs prior to requesting Freeze mode or low power mode.

A) Procedure to enter Freeze Mode:

1. Set the Freeze Enable bit (FRZ) in the Module Control Register (MCR).

2. Check if the Module Disable bit (MDIS) in MCR register is set. If yes, clear the MDIS bit.

3. Poll the MCR register until the Low-Power Mode Acknowledge (LPMACK) bit in MCR is cleared (timeout for software implementation is 2 CAN Bits length).

4. Read the Fault Confinement State (FLTCONF) field in the Error and Status 1 Register (ESR1) to check if FlexCAN is in bus off state. If yes, go to step 5A. Otherwise, go to step 5B.

5A. Set the Soft Reset bit (SOFTRST) in MCR.

6A. Poll the MCR register until the Soft Reset (SOFTRST) bit is cleared (timeout for software implementation is 2 CAN Bits length).

7A. Poll the MCR register until the Freeze Acknowledge (FRZACK) bit is set (timeout for software implementation is 2 CAN Bits length).

8A. Reconfigure the Module Control Register (MCR).

9A. Reconfigure all the Interrupt Mask Registers (IMASKn).

5B. Set the Halt FlexCAN (HALT) bit in MCR.

6B. Poll the MCR register until the Freeze Acknowledge (FRZACK) bit is set (timeout for software implementation is 178 CAN Bits length).

NOTE: The time between step 4 and step 5B must be less than 1353 CAN bit periods.

B) Procedure to enter in Low-Power Mode:

1. Enter in Freeze Mode (execute the procedure A).

2. Request the Low-Power Mode.

3. Poll the MCR register until the Low-Power Mode Acknowledge (LPMACK) bit in MCR is set (timeout for software implementation is 2 CAN Bits length).

e7724: FlexCAN: Documentation of ECC registers

Description: FlexCAN ECC registers listed below have issues in their descriptions.

1) Error Injection Address Register (CAN_ERRIAR):

FlexCAN registers and data structures located in RAM are 32-bit (4 bytes) wide, therefore, the two least sinificant bits in INJADDR bit field are don’t care. Read attempts to this register will result in address values multiple of 0x4.

2) Error Report Data Register (CAN_RERRDR):

RDATA bit field is read-only.

3) Error Report Syndrome Register (CAN_RERRSYNR):

BE3, BE2, BE1, BE0, SYND3, SYND2, SYND1 and SYND0 bit fields are read-only.

Workaround: No workaround needed.

e9527: FlexCAN: The transmission abort mechanism may not work properly

Description: The Flexible Controller Area NEtwork (FlexCAN) is not able to abort a transmission frame and the abort process may remain pending in the following cases:

a) If a pending abort request occurs while the FlexCAN is receiving a remote frame.

b) When a frame is aborted during an overload frame after a frame reception.

c) When an abort is requested while the FlexCAN has just started a transmission.

d) When Freeze Mode request occurs and the FlexCAN has just started a transmission.

Workaround: Use the Mailbox Inactivation mechanism instead of the transmission abort mechanism. The Abort Enable bit (AEN) of the Module Configuration Register should be kept cleared and the abort code value “0b1001" should not be written into the CODE field of the Message Buffer Control and Status word.

e8195: INTC: Interrupt Controller does not work correctly when in hardware vector interrupt mode and FMPERDIV is set to divide-by-4 or divide-by-

Description: Hardware vector interrupt mode is selected for a e200z7 core by setting the corresponding Hardware Vector Enable bit in the Interrupt Controller (INTC) Module Configuration Register (INTC_MCR[HVEN_PRC0/1]. If this mode is used in conjunction with a Frequency Modulated Peripheral Clock Divider (FMPERDIV) value of divide-by-4 or divide-by-8, multiple interrupt requests may be indicated when only a single interrupt event has occurred. This affects any peripheral interrupt request.

INTC software vector mode is not affected and works correctly regardless of FMPERDIV setting.

Workaround: Use software interrupt vector mode, if possible.

If hardware interrupt vector mode is required, use only FMPERDIV setting of divide-by-2.

e7545: JTAGC: TCK low to TDO data valid spec is not met

Description: The datasheet specification for Test Clock (TCK) low to Test Data Out (TDO) valid is not met on this device. The original spec is 10ns. This device requires 14ns.

Workaround: Use modified timing to latch data on the falling edge of TCK.

e7511: LFAST: Maximum data rate is 240Mbps

Description: The LVDS Fast Asynchronous Serial Transmission (LFAST) module in this device will operate at a maximum data rate of 240Mbps instead of the 320Mbps maximum rate specified in the device datasheet.

Workaround: Do not operate the LFAST module at any data rate greater than 240Mbps.

e9517: MCAN: Edge filtering may cause incorrect frame reception

Description: The MCAN may fail to correctly receive the first bit of the frame when edge filtering is activated in the CAN Core Control Register's Edge Filtering during Bus Integration field (CCCR.EFBI = '1') and when the end of the integration phase coincides with a falling edge at the receive (Rx) input pin. In this case, the Cyclic Redundancy Check (CRC) will detect that the received frame is faulty and an error frame will be sent.

When that frame is re-transmitted, the integration phase has finished and the frame is received correctly. Edge filtering is only applied during the integration phase. It is never used during normal operation.

The MCAN enters integration phase under the following conditions:

- When the Initialization control field (CCCR.INIT) is set to '0' after start-up

- After a protocol exception event where protocol exception handling is enabled (CCCR.PXHD = 0)

Workaround: Disable edge filtering or expect re-transmissions when this issue occurs.

e10109: MPC5777C: Enabling pending read optimization (off by default) can cause incorrect operation on the system crossbar

Description: The default setting for pending read optimization is disabled. If pending read optimization is enabled, this can cause the masters on the system crossbar (XBAR) to stall, receive wrong read data, or perform a spurious read access when uncorrectable Error Correcting Code (ECC) errors are reported by a XBAR slave. For a read transaction following an uncorrectable ECC error, these masters can stall or receive the wrong data.

Workaround: Do NOT enable pending read optimization (leave PCM_IAHB_BEn[PRE*] = 0, the default value) for all XBAR masters.

Core0: Clear bits PRE_CORE0_I and PRE_CORE0_D in register PCM_IAHB_BE1.

Core1: Clear bits PRE_CORE1_I and PRE_CORE1_D in register PCM_IAHB_BE1.

eDMA_A: Clear bit PRE_DMA_A in register PCM_IAHB_BE2.

eDMA_B: Clear bit PRE_DMA_B in register PCM_IAHB_BE2.

CSE2/SIPI: Clear bit PRE_M6 in register PCM_IAHB_BE2.

FEC: Clear bit PRE_FEC in register PCM_IAHB_BE2.

Performance note: Pending read optimization has an effect on performance of back-to-back reads issued by the given master. A general statement about performance impact of disabling this optimization cannot be made as it depends on data traffic patterns for all involved masters in the system. However, the following steps can be taken to reduce or eliminate any impact of disabling the pending read optimization for these masters:

For Core0 and Core1, enable instruction cache and data cache.

For Enhanced Direct Memory Access masters (eDMA_A and eDMA_B), program Transfer Control Descriptors (TCDs) where source size (ssize) and destination size (dsize) parameters are set to the same value. There is no performance impact in this case as the DMA does not perform back to back reads.

Cryptographic Services Engine (CSE2), Serial Interprocessor Interface (SIPI), and Fast Ethernet (FEC) masters typically do not benefit from pending read optimization and are therefore usually unaffected by disabling the feature.

e9770: MPC5777C: Incorrect operation of open-drain outputs

Description: When pins are configured as open-drain the high-drive for the output buffer is normally disabled. However, due to this Issue, during the transition to or from this non-driven state, the output may momentarily drive high. The duration of this drive is 0-10ns.

Workaround: Strong external drivers connected to the pin need to ensure they do not attempt to drive the pin within 10ns of the MCU transitioning to or from the non-driven state. This will prevent contention between the internal and any external drive sources.

e9856: MPC5777C: Incorrect operation of Software Watchdog Timers in debug mode

Description: While the device is in debug mode, if one e200z7 core is single-stepped while the other core is halted, the Software Watchdog Timers (SWT) for both cores will start and then continue to run instead of stopping after the step is executed. If both cores are single-stepped at the same time, the SWTs will halt as expected after the step.

Workaround: During debug mode, an e200z7 core should not be allowed to single-step while the other core is halted. The debug tool should be updated to always step both cores at the same time when a step is requested on either core by the user.

e8164: MPC577xC: Current injection causes leakage path across the DSPI and LFAST LVDS pins

Description: The General Purpose Input/Output (GPIO) digital pins (including all digital CMOS input or output functions of the pin) connected to the differential LVDS drivers of the Deserial/Serial Peripheral Interface (DSPI), high-speed debug, and LVDS Fast Asynchronous Serial Transmit Interface (LFAST) do not meet the current injection specification given in the operating conditions of the device electrical specification. When the LVDS transmitter or receiver is disabled and current is positively or negatively injected into one pin of the GPIO pins connected to the differential pair, a leakage path across the internal termination resistor of the receiver or through the output driver occurs potentially corrupting data on the complementary GPIO pin of the differential pair. All LFAST and DSPI LVDS receive and transmit GPIO pairs on the MPC577xC exhibit the current injection issue.

There is an additional leakage path for the LFAST receive pins through the loopback test path when current is negatively injected into a GPIO pin connected to an LFAST pair. In this case current will be injected into the same terminal of the GPIO pin connected through the loopback path (positive terminal to positive terminal, negative terminal to negative terminal). The pins affected by the loopback path on the MPC577xC are: SIPI_TXP to/from SIPI_RXP, and SIPI_TXN to/from SIPI_RXN.

There is no leakage issue when the pins are operating in normal LVDS mode (both LVDS pairs of the LFAST interface configured as LVDS).

Workaround: As long as the GPIO pad pins are operated between ground (VSS) and the Input/Output supply (VDDEHx) then no leakage current between the differential pins occurs. If the GPIO pad is configured as an input buffer then DC current injection must be limited to a maximum of 2.5mA. In this case, the adjacent CMOS pin will see a shift in the VOH/VOL levels and A/C timing if configured as an output. If the GPIO pad is configured as an output care should be taken to prevent undershoot/overshoot/ringing during transient switching of capacitive loads. This can be done by carefully configuring the output drive strength to the capacitive load and ensuring board traces match the characteristic impedance of the output buffer to critically damp the rising and falling edges of the output signal.

e10580: MPC577xC: Flash HVD may assert during slow input supply ramp

Description: During power up, under slow input ramp rates (10ms or slower to reach operating voltage level), a High-Voltage Detection (HVD) event may be observed. This HVD event is generated by the detector monitoring the flash regulator output (HVD_FLASH).

This may have the effect of prolonging the reset duration until the supply voltage reaches operational levels but does not lead to any incorrect operation of the device.

Workaround: Use a ramp rate faster than 10ms to reach operational voltage levels. If a slower ramp rate is used, check and clear Flash HVD status after each power up so that subsequent voltage detection events can be properly indicated.

e10714: MPC577xC: Lockstep errors must be ignored when debugging core 1 in lockstep mode

Description: When debugging core 1 in lockstep mode, Fault Collection and Control Unit (FCCU) faults NCF 10 and/or NCF 11 (Safety core out of sync) may be indicated. These are not real faults in the hardware but are instead due to debug tool access.

Workaround: In the FCCU, do not configure NCF 10 and NCF 11 reactions to generate a reset. Instead, use an interrupt reaction and program the interrupt handler to ignore these faults when debug mode is in use.

e10737: MPC577xC: Nexus Program Trace Sync Message sometimes missing after queue overflow

Description: When the core nexus trace functionality is enabled, the expected Program Trace Sync or Data Trace Sync Message following a FIFO overflow error message may not be reported, causing a larger window of uncorrelated trace information until the next sync message is transmitted.

Workaround: Avoid queue overflow by using the core Nexus stall feature. If using the stall feature is not desirable, the following actions may help reduce the occurrence of the condition.

- Use Branch History Mode to reduce the frequency of messages.
- Use triggers to disable and re-enable trace in areas of code where continuity of trace is not essential.

e9332: MPC577xC: Performance degradation caused by optimization control bits on Platform Configuration Module

Description: The “Pending Read Enable” (PRE_x) optimization option on the Platform Configuration Module (PCM) Bus Bridge Configuration Registers (PCM_IAHB_BEn) may introduce a performance degradation in some cases.

On this device, the Cryptographic Services Engine (CSE) and Serial Interprocessor Interface (SIPI) bus masters will always exhibit a degradation if the pending read optimization is enabled for the master port 6 concentrator (PCM_IAHB_BE2[PRE_M6] = 1). The Fast Ethernet Controller (FEC) will exhibit the problem when the Frequency Modulation Peripheral Clock Divider is set to divide-by-two (SIU_SYSDIV[FMPERDIV] = 0) and the pending read optimization is enabled for FEC (PCM_IAHB_BE2[PRE_FEC] = 1).

These pending read enable bits (PCM_IAHB_BEn[PRE_x]) are enabled (set) by default.

Workaround: Disable the pending read optimization for the M6 concentrator (PCM_IAHB_BE2[PRE_M6] = 0) if using the CSE and/or SIPI. Disable the pending read optimization for the FEC (PCM_IAHB_BE2[PRE_FEC] = 0) if the FM peripheral clock divider is set to divide-by-2. Other FM peripheral clock ratios are unaffected.

e10453: MPC577xC: Reduced accuracy on EQADC_B channels using VDDEH7 power domain

Description: Analog inputs for the Enhanced Queued Analog to Digital Converter B (EQADC_B) which are on the VDDEH7 power domain, including channels ANB[8:23] and ANB[40-45] may experience higher Total Unadjusted Error (TUE) due to noise coupling in the pad ring. This effect can occur when other input/output pins on the same VDDEH7 power domain are being driven. These may be output signals from internal peripheral modules or general purpose Input/Output. Input signals, if driven to the minimum or maximum allowable voltage levels may also cause the effect.

At 16MHz ADC sampling frequency, there is no impact to the TUE specification.

At 32MHz ADC sampling frequency, the impact to TUE specification is as follows:

- Worst case condition is cold temperature (-40C), where the TUE impact is an additional 6 counts (total of 14 counts).

- At room(25C) or hot (150C) junction temperature, the impact is lessened, with the TUE affected by an additional 4 counts (total of 12 counts).

The magnitude of the additional error is application-specific and depends on the usage of VDDEH7 domain input/output signals.

Workaround: If possible, use a sampling frequency of 16MHz or less with these inputs. If not, use averaging of multiple samples on these channels to reduce error impact and/or assign inputs with less need for accuracy to these channels. Move those requiring more sensitive measurements to EQADC_A channels or to unaffected EQADC_B channels that are on the VDDA_EQ supply domain.

e9784: MPC577xC: Reset escalation count is reduced by 1 when configured via UTEST DCF record

Description: The Reset Escalation Threshold (RET) value is stored in the System Integration Unit Reset Control Register (SIU_RCR). This value may be initialized either by writing to this register in normal operation or by Device Configuration Format (DCF) record stored in UTEST flash.

However, there is a difference in the effective reset escalation threshold depending on whether the SIU_RCR[RET] field is loaded via DCF record or written directly by code executing on the device. Writing SIU_RCR[RET] directly will cause the reset escalation counter to expire in the expected (RET+1) counts as stated in the Reference Manual.

If instead the SIU_RCR[RET] value is loaded via the UTEST Reset Escalation Configuration DCF record, then the escalation counter expires after a number of resets equal the the RET value rather than the value of RET+1.

Workaround: When configuring the SIU_RCR[RET] count via the UTEST Reset Escalation Configuration DCF client, be aware that the number of resets before expiration will be equal to RET rather than RET+1. Set the value of RET in the DCF record according to the number of reset cycles desired before reset escalation action is to be taken.

e10578: MPC577xC: The Temperature Sensor may cause extended reset times

Description: This device enables the Temperature Sensor Reset Enable bits (REE) by default during reset, and during reset, the Temperature Sensor analog block is powered down for a portion of the reset time.

This could lead to a condition where supply transitions while the Temperature Sensor analog block powers up may cause the Temperature Sensor flags to set, which would start the reset sequence over.

This could result in a longer reset cycle.

Workaround: The UTEST Power Management Controller (PMC) Temperature Sensor Reset Event Enable Control (REE TMPSNS) DCF client should be updated twice. The first value should be 0x00000040 (to load zeros into the REE bits), and the second value should be 0x00000000 (to allow software writes to make updates again).

After that, the UTEST PMC TS REE DCF should never be written to enable the Temp Sensor resets. This should be done via software updates to the PMC_REE_TD register.

In addition, when exiting from reset, check PMC_ESR_TD. If any of these bits are set to 1, the temperature should be checked by using the Temperature Sensor ADC measurement. If the temperature is within the valid range, then the PMC_ESR_TD bits should be cleared (by writing a 1 to their locations). After this, the PMC_REE_TD bits can be enabled to allow subsequent temperature event resets.

e10610: NPC: Core cannot be halted if NPC is not enabled when entering debug mode

Description: When attempting to halt a core via the corresponding COREn bit of the SIU_HLTn register while in debug mode and if the Nexus interface is enabled, the Nexus Port Controller (NPC) tracing must also be enabled or the core will not halt. 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), even if trace (program, data, ownership, watchpoint, data acquisition) is not enabled.

Workaround: Enable NPC tracing by enabling the Message Clock Output (MCKO) in the NPC Port Configuration Register (NPC_PCR) when a debugger is connected to allow a core to halt when requested.

e8340: 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.

e6726: NPC: MCKO clock may be gated one clock period early when MCKO frequency is programmed as SYS_CLK/8.and gating is enabled

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

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

e7120: NZxC3: DQTAG implemented as variable length field in DQM message

Description: The e200zx core implements the Data Tag (DQTAG) field of the Nexus Data Acquisition Message (DQM) as a variable length packet instead of an 8-bit fixed length packet. This may result in an extra clock (“beat”) in the DQM trace message depending on the Nexus port width selected for the device.

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

e9087: PADRING: Input High Voltage (max) and Hysteresis (min) specs not met

Description: The datasheet specifications for Input High Voltage, Vih (max), and Hysteresis (min) are not met in this device.

The original specifications are:

Vih max = (0.55 * VDDE).

Vih max with hysteresis enabled = (0.65 * VDDE).

Hysteresis min = (0.10 * VDDE).

The actual values met by this device are:

Vih max = (0.57 * VDDE).

Vih max with hysteresis enabled = (0.67 * VDDE).

Hysteresis min = (0.09 * VDDE).

Workaround: Ensure that external devices connected as inputs are compatible with these actual input level specifications. A future revision of the datasheet will be updated to reflect these new values.

e8369: PAD_RING: Reset output (RSTOUT) pin is not driven during Power-On Reset (POR) or Low-Voltage Detect (LVD) assertion

Description: RSTOUT, as an output pin, is expected to always be driven either high or low. However, in low-voltage conditions during POR and/or LVD reset (core VDD or Input/Output VDDEx supply domains), the RSTOUT pin will be in a high-impedance (hi-z) state until voltages reach the operational ranges specified in the datasheet.

Once out of POR/LVD state, RSTOUT pin is driven low until the device completes start-up and leaves reset condition..

Workaround: Add an external pull-down resistor if the application requires RSTOUT to always be driven low during any reset condition.

e9250: PASS: JTAG password not working during reset

Description: The Debug Interface Access cannot be enabled by supplying the JTAG password during reset.

Workaround: To enable the Debug Interface Access, supply the JTAG password after reset.

e10396: PASS: Password challenge to PASS fails while program erase ongoing in any block in memory partition 0

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

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

Workaround: Users should ensure that no program or erase operations are occuring on any memory partitions shared with the UTEST block before initiating a password challenge. This can be monitored through the flash module configuration register program and erase status bits (C55FMC_MCR[PGM], C55FMC_MCR[ERS]).

e7904: PASS: Programming Group Lock bit (PGL) can be de-asserted by multiple masters writing the correct password sections to the CINn registers.

Description: The eight Challenge Input Registers (CINn) in the Password and Device Security Module (PASS) where the 256-bit unlock lock password (8 × 32-bit registers) is provided, can be written by multiple masters. If the written password is correct even though it has been provided from different masters, the password Group Lock (PASS PGL) in the Password Group n Lock 3 Status register (PASS_LOCK3_PGn) is de-asserted and UnLockMaster (MSTR) is set to 0xF.

Therefore, internal registers would not be writable by any of the master other than master whose ID is 0xF if the Master Only (MO) bit is set PASS_LOCK3_PGn.

If a Master wants to update internal registers, it needs to unlock the PASS by writing into all the 8 Password registers.

Workaround: Set the master only bit inside the PASS (LOCK3_PGn.MSTR) to block other master accesses to the unlocked registers. If the written password has been provided from different masters, a single master should perform the unlock operation again by writing into all the 8 password registers.

e10226: PMC: In SMPS mode, during input supply power down, core voltage can exceed specification limits

Description: When the Switch Mode Power Supply (SMPS) of the Power Management Controller (PMC) is used for internal regulation and the PMC input supply (VDDPMC) ramps down, the output of the regulator (core voltage supply) may briefly rise above specification limits for a duration which is dependent on the ramp down rate. This core over-voltage condition may, over time, cause reliability issues with the device.

The issue is more likely to occur when:

1) VDDPMC and VDDPWR (PMC SMPS driver supply) supply inputs are driven from different sources and VDDPMC ramps down before VDDPWR.Note that on some devices these two supplies are required to be tied together.Refer to the device data sheet.

2) In the case where VDDPMC/VDDPWR are tied together, the problem is most likely to occur when the ramp down time is long, that is, for slow ramp down rates.

The issue occurs when the PMC stops driving the Regulator Control (REGCTL) output while it is in the low (on) position.This leaves the external p-MOSFET device on until the pull-up resistor can shut the device off.For large values of the resistor, this can require enough time that the core voltage output of the regulator briefly drives above specification limits.

Workaround: An external pull-up resistor in the range of 2K to 4.7K ohms must be used between REGCTL and VDDPWR.

VDDPMC and VDDPWR supplies must ramp down through the voltage range from 2.5v to 1.5v in less than 1 second. Slower ramp down times may result in reduced lifetime reliability of the device

e9804: PMC: LVD/HVD Event Status Register does not consistently indicate the reset source

Description: The Power Management Controller (PMC) LVD/HVD Event Status Register (PMC_LVD_HVD_EVENT_STATUS) does not always indicate the reset source for a High-Voltage Detect (HVD) or Low-Voltage Detect (LVD) event when reset is enabled through the PMC Reset Event Enable register (PMC_REE), although the reset itself occurs.

Workaround: Disable LVD/HVD resets in PMC_REE by clearing all bits to zero. Instead, configure the Fault Collection Control Unit (FCCU) reaction to reset the part upon LVD or HVD fault events. These are Non-Critical Fault (NCF) channels 1 and 2. After any reset event caused by the FCCU LVD/HVD faults, reading the PMC_LVD_HVD_EVENT_STATUS register will correctly identify the reset source.

Example of configuration and reset source check sequence:

1) Configure PMC_REE = 0x0000 to disable all LVD/HVDs reset

2) Configure FCCU registers to generate reset

FCCU.LVD_ERROR(NCF=1) and/or

FCCU.HVD_ERROR(NCF=2)

3) Over/under operating voltages can generate HVD/LVD reset

4) Check if reset was generated through FCCU

5) Check FCCU registers indicate source of reset

6) Check PMC LVD HVD Status occurrence

Note that this FCCU workaround has a limited selection resolution regarding the LVD/HVD source that causes a reset event when compared to PMC_REE configuration. The FCCU cannot individually enable/distinguish which LVD source caused the FCCU LVD request. The same applies to HVD.

e10135: PMC: Resets that occur during the PMC Self Test process can cause corrupted results.

Description: Some system resets, if they occur while the Power Management Controller (PMC) self test is running, can result in the PMC self test logic being left in a corrupted state. The affected reset sources are: System software reset, Fault Collection and Control Unit (FCCU) reset fault reactions, and debug (JTAG) reset.

Workaround: After a reset caused by one of the affected reset sources, check the status of the self test by reading the PMC Voltage Detect User Mode Test Register (PMC_SELF_TEST_UM_VD_REG). If a PMC self test failure is indicated, run the PMC self test again to confirm the failure.

e5887: PSI5: Detection of a received bit causes an electrical error in specific conditions

Description: During the Peripheral Sensor Interface 5 (PSI5) bit extraction from the Manchester encoded receive data, if the transition of the data occurs in the first two clock periods of the detection window then the transition is not recognized and the electrical error bit (PSI5_PMRRH[E]) will be set in the PSI5 Message Receive Register High.

Workaround: PSI5 sensors are specified to have a 45%/55% duty cycle output worst case. If the external PSI5 bus line circuit is not balanced then this can potentially alter the duty cycle of the sensor data received by the PSI5 module. The user must ensure that the sensor bus circuit is designed such that the Manchester Coded signal received by the PSI5 module has a worst case 30%/70% duty cycle

e9863: PSI5: Enabling interrupts in the General Interrupt Control Register (PSI5_CH0_GICR) has no effect

Description: Interrupt sources described in the Peripheral Sensor Interface 5 (PSI5) General Interrupt Control Register (PSI5_CH0_GICR) cannot cause an interrupt. Enabling interrupt control bits in this register has no effect. The pending interrupt status for these events, however, will be correctly indicated in the General Interrupt Status Register (PSI5_CH0_GISR).

Workaround: In order to respond to these events, the system must poll the status bits in the PSI5_CH0_GISR register.

e6992: PSI5: IS_DEBUG_FREEZE bit is not documented

Description: Bit 0 (the most significant bit of the register, MSB) of the Peripheral Sensor Interface 5 General Interrupt Status Register (PSI5_GISR) is not documented. This bit is the IS_DEBUG_FREEZE bit and is set when the PSI5 module is stopped in debug freeze mode. To enable the debug freeze mode, both PSI5 Debug mode Enable and the Debug Freeze Control bits must be set in the PSI5 Channel Control register (PSI5_PCCR[DEBUG_EN] = 0b1 and PSI5_PCCR[DEBUG_FREEZE_CTRL] = 0b1). When thePSI5 module receives the request to enter the debug mode, it finishes the current processing and is stopped coherently. At this stage, the IS_DEBUG_FREEZE bit is set to “1” . This bit automatically gets cleared by the hardware when the debug mode is exited.

Workaround: Expect bit 0 (MSB) of the PSI5_GISR register to be set when the PSI5 module is stopped in debug freeze mode if both DEBUG_EN and DEBUG_FREEZE_CTRL are set. The documentation will be updated.

e7234: PSI5: No transfer error generated for accesses within the unused range of the PSI5 peripheral window

Description: The Peripheral Sensor Interface (PSI5) uses 4 Kbytes of the 16 Kbytes range of the peripheral bridge slot assigned to it.

Accesses after the 4 Kbytes (from offset 0x1000 to offset 0xFFFF) will not generate a transfer error.

Note: accesses to unimplemented locations within the 4 Kbyte window will correctly generate a transfer error.

Workaround: Take into account that no transfer error will be generated outside the 4 Kbyte region used by the PSI5 module.

In case such accesses must be detected, use the memory protection unit (MPU) to limit accesses.

e5073: PSI5: Possible message reception errors due to incorrect data latency reference point

Description: The Peripheral Sensor Interface (PSI5) module incorrectly defines the end of the message to coincide with the detection of idle, which occurs one T-Bit (bit time in counts of the sample clock) duration after the midpoint transition of the CRC0 (Cyclic Redundancy Check) or Parity bit of the frame. However the PSI5 specification defines the end of the frame as the midpoint transition of the CRC/Parity bit. This difference means that a correctly timed PSI5 frame could be interpreted as arriving before the previous frame had completed.

Workaround: The error can cause at least one of the following status bits to be set:

PSI5 Message receive register low “C-bit” (PSI5_PMRRL[C]) - Indicates CRC recalculation error in message

PSI5 Message receive register high “E-bit” (PSI5_PMRRH[E]) -Indicates Electrical error

PSI5 Message receive register high “T-bit” (PSI5_PMRRH[T]) -Indicates Timing error

Software can monitor these bits, and if these errors are present on several concurrent messages then the software can

re-initialize the system

e6553: PSI5: T bit error Ambiguity is noticed in Synchronous mode

Description: When the Peripheral Sensor Interface (PSI5) module is configured in synchronous mode and two frames are consecutive, the T bit error flag, in the PSI5 Message Receive Register High (PSI5_PMRRH), may incorrectly be set in the PSI5 Message Receive Register High (PSI5_PMRRH) of the second frame. This happens when the duration of the gap between the frames is less than the duration of one idle T bit.

Workaround: Ensure a delay of one idle T bit period between two consecutive data frames in synchronous mode.

One Idle T bit duration is 32 cycles of the PSI5 sampling clock.

e8368: REACM2: ETPU_C clock does not halt if REACM2 Module Disable (MDIS) is set to 1

Description: In this device, the Enhanced Time Processor Unit C (ETPU_C) and Reaction Module (REACM2) share the same clock domain. A clock halt request performed by setting System Integration Unit Halt 1 ETPU_C bit (SIU_HLT1[ETPUC]) requires an acknowledge response from both modules to correctly complete. If REACM2 Module Configuration Register Module Disable bit (REACM2_MCR[MDIS]) is set to 1, REACM2 will not return the required acknowledge signal. As such, the ETPU_C clock will not halt.

Workaround: Before submitting a halt request to ETPU_C, ensure that REACM2 is in its active mode (REACM2_MCR[MDIS]=0).

e8367: REACM: Register is unexpectedly written after exiting halted state

Description: If the Reaction Module (REACM) is set to low-power mode by asserting the associated Enhanced Time Processor Unit C (ETPU_C) halt bit in the System Integration Unit Halt 1 register (SIU_HLT1) and then writes are performed to REACM registers during the halted state, the first such write access will actually take effect when the REACM module later exits the halted state. Read and write accesses during the halted state will still generate the appropriate bus error indications.

Workaround: Workaround 1: Detect improper accesses during the halted state by using the bus error exception and then either re-initialize REACM to a known-good state after exiting the halted state or restore its register contents from a backup area in RAM created prior to entering the halted state.

Workaround 2: Prevent access to the REACM during the halted state by using a Memory Protection Unit (MPU) region.

e10415: SDADC: Additional DMA request generated with slow SDADC SD_CLK clock

Description: The Sigma-Delta Analog-to-Digital Converter (SDADC) may issue an extra Direct Memory Access (DMA) transfer request when the ratio between SDADC conversion clock (SD_CLK) period and the SDADC bus interface clock (FM_PER_CLK) is more than 8:

(PERIOD SD_CLK)/PERIOD FM_PER_CLK)>8.

The DMA transfer may complete before the SDADC data comparison cycle is completed, actually triggering a new DMA transfer request.

Workaround: When using Direct Memory Access transfer request, including transfers using watchdog threshold, ensure that the relation between SD_CLK and FM_PER_CLK is: (PERIOD SD_CLK)/PERIOD FM_PER_CLK)<=8

e8711: SDADC: Digital filter and FIFO not disabled when MCR[EN] is cleared

Description: When the Enable bit (EN) of the Sigma-Delta Analog to Digital Converter (SDADC) Module Configuration Register (MCR) is cleared (MCR[EN]=0), the digital part of the SDADC continues operating and does not go to low power mode if the module is disabled while a valid conversion is already in process and the application software continues to initiate conversions. As a consequence, the digital block of the SDADC still produces new conversion results in the Channel Data Register (CDR) and dummy data are transferred to the result First-In, First-Out (FIFO) buffers. In addition, interrupt and/or Direct Memory Access (DMA) events are still generated.

Note: the analog part does enter the power-down mode, reducing the consumption on the ADC high voltage supply domain (VDDA_SD).

Workaround: Do not initiate a conversion prior to enabling the SDADC (MCR[EN]=1). In addition, once the SDADC has been enabled (MCR[EN]=1), if the SDADC needs to be disabled (MCR[EN]=0), prior to clearing the EN bit, either turn off the clock to the SDADC module in the System Integration Unit (SIU) or Select the External Modulator Mode (EMSEL) by setting the MCR[EMSEL] bit along with the clearing the MCR[EN].

e8225: SDADC: FIFO Flush Reset command requires clearing the Data FIFO Full Flag

Description: When the Sigma-Delta Analog-to-Digital Converter (SDADC) FIFO is flushed by writing ‘1’ to the FIFO Control Register FIFO Flush Reset bit (SDADC_FCR[FRST]), the FIFO is correctly flushed, but the Status Flag Register Data FIFO Full Flag (SDADC_SFR[DFFF]) may be incorrectly asserted, indicating the FIFO is full when it is empty..

Workaround: Clear SDADC_SFR[DFFF] by writing a ‘1’ to this field after performing a FIFO Flush Reset command or after the FIFO is disabled.

e10378: SDADC: Incorrect data provided when FIFO is disabled and FIFO overwrite is enabled

Description: The Sigma-Delta Analog-to-Digital Converter (SDADC) allows continuous data acquisition. When the overwrite functionality is enabled (FCR[FOWEN] = 1) in the FIFO Control Register (FCR), previously converted data will eventually be overwritten if it is not read before new data is available. In case the overwrite functionality is enabled (FCR[FOWEN] = 1) in the FIFO Control Register (FCR) together with the disabling of the associated FIFO buffer (FCR[FE] = 0), the Data FIFO Empty flag in the Status Flag register (SFR[DFEF]) will toggle high (Data FIFO is empty) and low (Data FIFO is not empty). If the Converted Data Register (CDR) is read while in the “Data FIFO is empty" state, then the previous converted data is provided rather than newest converted data.

Workaround: Always disable the FIFO overwrite functionality in the FIFO Control Register (FCR[FOWEN]=0) if the FIFO buffer is disabled (FCR[FE]=0).

e6906: SDADC: Invalid conversion data when output settling delay value is less than 23

Description: In the Sigma Delta Analog to Digital Converter (SDADC), if the Output Settling Delay field of the Output Settling Delay register (OSDR[OSD]) is programmed to a value less than 23 then the initial converted data from SDADC block is “0000” instead of the correct conversion result.

Workaround: Program the OSDR[OSR] value equal to or greater than 23.

e8631: SDADC: low threshold watchdog cannot be used with signed data

Description: Each Sigma Delta Analog to Digital Converter (SDADC) provides a watchdog (WDG) to monitor the converted data range. This watchdog should trigger when a converted value is either higher than the value configured in the WDG Threshold Register Upper Threshold Value bit-field (SDADC_WTHHLR[THRH]), or lower than the value configured in the Lower Threshold Value bitfield (SDADC_WTHHLR[THRL]). Instead, the low WDG threshold acts as a high WDG threshold, triggering when a converted value is greater than the value configured in SDADC_WTHHLR[THRL].

Workaround: There are two workarounds available:

1) Do not use the WDG function by clearing the SDADC Module Control Register Watchdog Enable Bit (SDADC_MCR[WDGEN]).

2) Configure the WDG low threshold SDADC_WTHHLR[THRL] to the value 0x7FFF. This guarantees that a low threshold trigger will not be generated. The WDG high threshold (SDADC_WTHHLR[THRH]) can be used without restriction.

e7356: SDADC: The SDADC FIFO does not function correctly when FIFO overwrite option is used

Description: In the Sigma-Delta Analog-to-Digital Converter (SDADC), when the FIFO Over Write Enable bit (FOWEN) of the FIFO Control Register (FCR) is set (FCR[FOWEN]=1), the following flags of the Status Flag Register (SFR) may not reflect the correct status:

- Data FIFO Full Flag (DFFF)
- Data FIFO Empty Flag (DFEF)

When the number of entries received by the FIFO reaches 2x the FIFO size (field FSIZE of FIFO Control Register (FCR)):

- SFR[DFFF] is cleared, incorrectly indicating the FIFO is not full
- SFR[DFEF] is set, incorrectly indicating the FIFO is empty

The expected behavior is that:

- SFR[DFFF] remains set until data is read out of the FIFO
- SFR[DFEF] remains clear until all data is read out of the FIFO

Workaround: Do not use the FIFO Overwrite option to overwrite FIFO contents. Software shall clear the FIFO overrun condition (if necessary) and flush the FIFO contents before expecting valid data in the FIFO.

e8710: SDADC: Watchdog Crossover event missed if FM Peripheral Clock frequency is less than or equal to the sigma-delta ADC clock frequency

Description: In the Sigma-Delta Analog-to-Digital Converter (SDADC), the watchdog monitor Lower and Higher threshold crossover events may get missed if the FM peripheral clock (FM_PER_CLK) has a frequency less than or equal to the SDADC clock (SD_CLK). Therefore, the Watchdog Upper Threshold Cross Over Event (WTHH) and Watchdog Lower Threshold Cross Over Event (WTHL) bits of the Status Flag Register (SDADC.SFR) may not be set and the corresponding Direct Memory Access (DMA) or interrupts are not triggered.

Workaround: When setting the different clocks in the System Integration Unit (SIU), ensure that FM_PER_CLK frequency is always greater than SD_CLK. This includes the variation in FM_PER_CLK due to frequency modulation, if enabled.

e7204: SENT: Number of Expected Edges Error status flag spuriously set when operating with Option 1 of the Successive Calibration Check method

Description: When configuring the Single Edge Nibble Transmission (SENT) Receiver (SRX) to receive message with the Option 1 of the successive calibration pulse check method (CHn_CONFIG[SUCC_CAL_CHK] = 1), the number of expected edges error (CHn_STATUS[NUM[EDGES_ERR]) gets randomly asserted. Option 2 is not affected as the number of expected edges are not checked in this mode.

The error occurs randomly when the channel input (on the MCU pin) goes from idle to toggling of the calibration pulse.

Note: The Successive Calibration Pulse Check Method Option 1 and Option 2 are defined as follows:

Option 2 : Low Latency Option per SAE specification

Option 1 : Preferred but High Latency Option per SAE specification

Workaround: To avoid getting the error, the sensor should be enabled first (by the MCU software) and when it starts sending messages, the SENT module should be enabled in the SENT Global Control register (by making GBL_CTRL[SENT_EN] = 1). The delay in start of the two can be controlled by counting a fixed delay in software between enabling the sensor and enabling the SENT module. The first message will not be received but subsequent messages will get received and there will be no false assertions of the number of expected edges error status bit (CHn_STATUS[NUM[EDGES_ERR]).

Alternatively, software can count the period from SENT enable (GBL_CTRL[SENT_EN] = 1) to the first expected calibration pulse. If the number of expected edges error status bit (CHn_STATUS[NUM[EDGES_ERR]) is asserted, software can simply clear it as there have no messages which have been completely received.

Alternatively, the software can clear this bit at the start and move ahead. When pause pulse is enabled, then NUM_EDGES will not assert spuriously for subsequent messages which do not have errors in them or cause overflows.

e8082: SENT: A message overflow can lead to a loss of frames combined with NUM_EDGES_ERR being set

Description: In the case of a Single Edge Nibble Transfer (SENT) receiver (Rx) message overflow (CHn_STATUS[FMSG_OFLW] = 1) and if the following registers are continuously being read without clearing the FMSG_RDY[F_RDYn] bit, there is a possibility that one message will be lost. Additionally, if the pause pulse feature is enabled, the module assert up to two NUM_EDGES_ERR in the status register (CHn_STATUS). In this case up to two frames can be lost.

Note that some debuggers perform a continuous read of memory which can cause this issue to occur.

Register	Register Name
CHn_FMSG_DATA	Channel Fast Message Data Read Register
CHn_FMSG_CRC	Channel Cyclic Redundancy Check Register
CHn_FMSG_TS	Channel Fast Message Time-stamp Register

Workaround: 1. Software should ensure that SENT message overflow does not occur.

If interrupts are used (when the Enable FDMA (FDMA_EN) bit of Fast Message DMA Control Register (SRX_FDMA_CTRL) is set to 0) to read the SENT messages, the interrupt for data reception should be enabled by setting the Enable for Fast Message Ready Interrupt (FRDY_IE[n]) bit of Fast Message Ready Interrupt Control Register (SRX_FRDY_IE) for every channel n and the interrupt priority should be such that the software is able to read the message before the next message arrives.

When using Direct Memory Accesses (eDMA) to access the SENT (when the Enable FDMA (FDMA_EN) bit of Fast Message DMA Control Register (SRX_FDMA_CTRL) is set to 1), the DMA request from the SENT module should be serviced before the next message arrives.

The minimum duration between the reception of two consecutive messages in one channel is 92 times the utick length (time).

2. Ensure that the following registers are not read continuously either in the software code or as a result of a debugger being connected. The following registers should be read once per message and the FMSG_RDY[F_RDYn] bit should be cleared after the reads.

Register	Register Name
CHn_FMSG_DATA	Channel Fast Message Data Read Register
CHn_FMSG_CRC	Channel Cyclic Redundancy Check Register
CHn_FMSG_TS	Channel Fast Message Time-stamp Register

e7425: SENT: Unexpected NUM_EDGES_ERR error in certain conditions when message has a pause pulse

Description: When the Single Edge Nibble Transmission (SENT) Receiver (SRX) is configured to receive a pause pulse (Channel ‘n' Configuration Register – CHn_CONFIG[PAUSE_EN] = 1) the NUM_EDGES error can get asserted spuriously (Channel ‘n' Status Register – CHn_STATUS(NUM_EDGES_ERR] = 1) when there is any diagnostic error (other than number of expected edges error) or overflow in the incoming messages from the sensor.

Workaround: Software can distinguish a spurious NUM_EDGES_ERR error from a real one by monitoring other error bits. The following tables will help distinguish between a false and real assertion of NUM_EDGES_ERR error and other errors. Software should handle the first error detected as per application needs and other bits can be evaluated based on these tables. The additional error may appear in the very next SENT frame. Table 1 contains information due to erratum behavior. Table 2 contains clarification of normal NUM_EDGES_ERR behavior.

Table 1. Erratum behavior of NUM_EDGES_ERR

First Error Detected	Other error bits asserted	Cause for extra error bits getting asserted	Action
NIB_VAL_ERR	NUM_EDGES_ERR asserted twice	Upon detection of the first error, the state machine goes into a state where it waits for a calibration pulse, the first NUM_EDGES_ERR error is for the current message as the state machine does not detect an end of message. The second error comes when both the Pause pulse and the Calibration pulse are seen as back to back calibration pulses and no edges in between.	Ignore both NUM_EDGES_ERR error
FMSG_CRC_ERR	NUM_EDGES_ERR asserted twice	Same as NIB_VAL_ERR.	Ignore both NUM_EDGES_ERR errors
CAL_LEN_ERR	NUM_EDGES_ERR asserted once	Since the calibration pulse is not detected as a valid calibration pulse, the internal edges counter does not detect the end of one message and start of bad message (which has CAL_LEN_ERR); hence the NUM_EDGES_ERR gets asserted.	Ignore NUM_EDGES_ERR error
FMSG_OFLW	NUM_EDGES_ERR asserted once (random occurrence)	A message buffer overflow may lead the state machine to enter a state where it waits for a calibration pulse	Ignore NUM_EDGES_ERR error

		(behavior also seen in ERR007404). When in this state, the state machine can detect both a Pause pulse and a Calibration pulse as back to back calibration pulses and no edges in between. Then, the NUM_EDGES_ERR can get asserted. Since entry into this state is random, the error can be seen occasionally.	

Table 2. Expected behavior, clarification of NUM_EDGES_ERR cases

First Error Detected	Other error bits asserted	Cause for extra error bits getting asserted	Action
NUM_EDGES_ERR (when edges are less than expected)	NIB_VAL_ERR is asserted	When the actual number of edges in the message are less than expected, then a pause pulse gets detected as a nibble since the state machine expects nibbles when actually there is a pause pulse present. This generates NIB_VAL_ERR.	Ignore the NIB_VAL_ERR
NUM_EDGES_ERR (when edges are more than expected)	NIB_VAL_ERR and PP_DIAG_ERR are asserted	When the actual number of edges in a message are more than expected, then after receiving the programmed number of data nibbles, the state machine expects a pause pulse. However, the pause pulse comes later and gets detected as a nibble and hence NIB_VAL_ERR is asserted. Since the message length is not correct, PP_DIAG_ERR is also asserted.	Ignore NIB_VAL_ERR and PP_DIAG_ERR

e10645: SIU: WKPCFG is not applied until after initial reset negation

Description: After power-up of the device, Weak-pull Configuration (WKPCFG) is not applied until after the negation of the initial reset input (RESET pin).

If Self-Test Control Unit (STCU) offline self-test is disabled, then WKPCFG is applied once, just before Reset Output (RSTOUT) is negated.

If STCU offline self-test is enabled, then WKPCFG is applied twice: first just before STCU self-test begins, and second just before RSTOUT is negated (re-applying WKPCFG value from DCF or pin each time),

Therefore, the pad state during reset for initial reset assertion for pins that use WKPCFG is determined by the default value of the System Integration Unit Pad Configuration Register Weak Pull State bit (SIU_PCR[WPS]) for each pin, not by WKPCFG.

The pad state during reset for subsequent resets is based on the previously applied WKPCFG value (from prior RSTOUT negation). Then at subsequent reset negation, the present value of WKPCFG is applied.

The SIU_PCR[WPS] reset value for all pins that use WKPCFG is 0. These pins are PCR numbers: 114-145, 147-204, 432-437, 441-472, and 475-491.

Workaround: System must not rely on internal weak pull up or down for WKPCFG-controlled pins during reset assertion. External circuits monitoring such pins should mask out the value while reset is asserted, or add external pull resistors if needed.

e9658: 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.

e7431: STCU2: LBIST does not accept programmable seed value

Description: The Logic Built-In Self Test (LBIST) feature is specified to include a Pseudo-Random Pattern Generator (PRPG) load value that would allow a user to experiment with different seed values for running the self-test. This feature is non-functional. A default seed value will be used instead and programming values in registers STCU2_LB_PRPGL/Hn will have no effect.

Workaround: The programmable seed value feature cannot be used. This does not affect the usability of the self-test feature as the default seed value is sufficient. Appropriate values for input test cycle length and expected MISR (Multiple-Input Signature Register) result should be found in user documentation for this device.

e7339: STCU2: STCU2 fault injected by FCCU is self clearing

Description: In the Self-Test Control Unit (STCU2), a fault can be injected by the Fault Collection and Control Unit (FCCU) in order to verify the correct behavior of the interface (fake fault).

The STCU_LMBIST_USR_ERR signal, which is connected to the FCCU input #8, generates only a pulse when an error is injected to this signal by the FCCU.

This is different to other signals from STCU2, where injected faults remain asserted until explicitly cleared.

Workaround: Use a software recoverable fault (select-able with FCCU_RF_CFG) for FCCU input #8, when a fault is injected into the STCU2.

e10636: STCU: Improper behavior in some cases at hot temperatures during offline LBIST and MBIST operation.

Description: At hot temperatures, the JTAG Compliance (JCOMP) input pin is not effectively pulled down internally and may prevent the Self-Test Control Unit (STCU) from performing a configured off-line self-test operation (MBIST and/or LBIST).

Workaround: After normal operation resumes after a reset, the device will report the cause of reset as a reset in the System Integration Unit Reset Status Register (SIU_RSR). STCU status registers may indicate that no self-test has executed. At this point, the application can re-run self-test in online mode.

e10443: STCU: Improper behavior in some cases when external reset is asserted during LBIST execution

Description: If the external reset pin (RESET) is asserted during Self-Test Control Unit (STCU) Logic Built-In Self Test (LBIST) execution, incorrect operation may be observed in some cases. This incorrect operation will have one of two possible effects:

(1) Following external reset assertion, the device will be unable to read Device Configuration Format (DCF) records and will experience the STCU initialization timeout of ~8.6ms, after which DCF records will load correctly and normal operation will resume.

(2) Following external reset assertion, the device will reload DCF records and attempt to restart self test procedure. It will, however, be unable to restart self test and will experience the STCU watchdog timeout programmed in the STCU2 Watchdog Register Granularity (STCU2_WDG). After this timeout, the device will reset and not attempt self test execution again. Normal operation will resume.

Effect #1 will be observed in either online or offline (or both) execution of one or more LBIST partitions and the external reset is asserted during LBIST execution of partition 0, 1, 2, 3, or 4. This case occurs regardless of whether MBIST is configured or not.

Effect #2 will be observed if offline execution of one or more MBIST partitions is configured and offline or online execution (or both) of one or more LBIST partitions is performed and the external reset pulse is asserted during test of LBIST partition 5.

All other combinations of self test execution online or offline (or both) are not affected by this issue.

Workaround: For effect #1: After normal operation resumes, the device will report the cause of reset as external reset in System Integration Unit Reset Status Register (SIU_RSR). STCU status registers will indicate no self-test has executed. At this point, the application can either re-run self test online or, if possible, signal an external device to apply external reset in order to cause offline self-test to run again.

For effect #2:

Ensure that the BIST watchdog timer (STCU2_WDG[WDGEOC]) is properly initialized to limit the amount of time that the device will remain in reset due to this issue. The maximum value for the watchdog (0xffff_ffff) causes a very long timeout and the device may appear to be stuck in reset indefinitely. For example, if the system clock is configured to 50MHz for the test execution and STCU clock configuration (STCU2_CFG[CLK_CFG]) is set to divide-by-2, the watchdog timeout period will be (0xffff_ffff * 16 STCU clocks) / 12.5MHz or about 1.5 hours.

e10025: SWT: System clock source must be set to IRC prior to changing SWT clock selection

Description: The Software Watchdog Timer (SWT) Control Register Clock Select bit (SWT_CR[CSL]) must only be changed from 0 to 1 or from 1 to 0 while the system clock source is set to Internal RC Oscillator (IRC). Otherwise, unpredictable behavior may result.

Workaround: Follow this procedure to change the SWT clock selection:

1. Set the system clock source to IRC.

2. Write SWT_CR[CSL] to change its value from 0 to 1 or from 1 to 0.

3. If desired, the system clock source may now be changed without affecting proper operation of the device.

e9336: TDM: Erase of TDR flash block may be blocked by the TDM - CSE systems

Description: Erase of a flash block associated with a Tamper Detect Region (TDR) may be improperly blocked by the Tamper Detect Module (TDM) in a system with an active CSE (Cryptographic Services Engine) . When this occurs, the erase operation will be shown as complete with Program/Erase Good (c5FMC_MCR.DONE=1 and C55FMC_MCR.PEG=1), but the block will not be erased

Workaround: Avoid CSE program/erase operations when erasing a flash block covered by a TDR. Alternatively, when erase of a flash block is attempted, but it completes with C55FMC_MCR.PEG=1 and is not erased, a new diary entry should be written before attempting to erase the flash block again.

e7236: XBIC: XBIC may trigger false FCCU alarm

Description: The Crossbar Integrity Checker (XBIC) will incorrectly signal a fault alarm when a system bus request results in a bus error termination from a crossbar client. The Fault Correction and Collection Unit (FCCU) alarm number corresponding to the XBIC will be signaled.

Workaround: Software should handle faults on FCCU alarm corresponding to the XBIC in case of a system bus error.

e8310: XBIC: Crossbar Integrity Checker may miss recording information from an initial fault event in the case of back-to-back faults

Description: When the Crossbar Integrity Checker (XBIC) detects back-to-back faults on a system bus path through the crossbar switch (AXBS), the fault information captured in the XBIC Error Status Register (XBIC_ESR) and the XBIC Error Address Register (XBIC_EAR) does not correspond to the initial fault event, but rather the subsequent fault event. While the fault event is properly detected, diagnostic status information in the XBIC_ESR and XBIC_EAR registers describing the initial fault event is lost. This defect can only occur in the event of a series of bus transactions targeting the same crossbar slave target, where the series of bus transactions are not separated by idle or stall cycles.

Workaround: Expect that the XBIC_EAR and XBIC_ESR registers may not contain the initial fault information, but will contain the latest fault information.

e8730: XBIC: XBIC may store incorrect fault information when a fault occurs

Description: The Crossbar Integrity Checker (XBIC) may incorrectly identify a fault's diagnostic information in the case when the slave response signals encounter an unexpected fault when crossing the crossbar switch (XBAR) during the data phase. While the fault event is detected, the diagnostic status information stored in the XBIC's Error Status Register (XBIC_ESR) and Error Address Register (XBIC_EAR) does not reflect the proper master and slave involved in the fault. Instead, the preceding master or slave ID may be recorded.

Workaround: Expect that when a fault is reported in the XBIC_EAR and XBIC_ESR registers the actual fault information may be from the preceding transition.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

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

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.