SPC574K72E5, SPC574K72E7 Errata sheet

SPC574K72E5, SPC574K72E7 device errata JTAG_ID = 0x0AF06041

Introduction

This errata sheet describes all the functional and electrical problems known in the revision 1.0 of the SPC574Kxx devices, identified with the JTAG_ID = 0x0AF06041.

All the topics covered in this document refer to SPC574Kxx reference manual (*RM0334 rev 2*) and *SPC574Kxx datasheet rev 3* (see *A.1: Reference document*).

Device identification:

- JTAG_ID = 0x0AF06041
- MIDR register:
  - MAJOR_MASK[]: 0
  - MINOR_MASK[]: 0
- Package device marking mask identifier: AA

This errata sheet applies to SPC574Kxx device in accordance with *Table 1*.

Table 1. Device summary

Part number	Package
SPC574K72E5	eTQFP144
SPC574K72E7	eLQFP176

Contents

List of tables

1 Functional problems

1.1 ERR002775: Core_0: Safety core I-MEM and D-MEM arrays do not implement additional measures against address and control faults

Description:

Core_0: The additional dedicated measures against address and control faults are not implemented for the Core_0 (safety core) memories (I-MEM and D-MEM). Protections is still provided by Error Correction Code (ECC) which is computed for data and address signals that will detect data and addressing faults.

Workaround:

Rely on ECC coverage on Core_0 (safety core) I-MEM and D-MEM arrays for protection against address and control errors. This version is not intended to be used for series production of safety-relevant applications.

1.2 ERR003632: DCI: Cannot disable watchdog via EVTI assertion

Description:

Assertion (low) of the Event In 0 pin (EVTI0) while negating the JTAG Compliance pin (JCOMP) will not result in disabling of system watchdog timers (SWT).

Workaround:

If the SWT needs to be disabled by a tool, it should be disabled by clearing the Watchdog Enable bit in the SWT Control register (SWT_CR[WEN]=0b0).

1.3 ERR003660: MEMU: MBIST error address reported for the Fast Ethernet Controller (FEC) Receive Interface FIFO (RIF) and Message Information Block (MIB) RAM in the Memory Error Management Unit (MEMU) is not correct

Description:

When an error in the Fast Ethernet Controller (FEC) RAM Interface FIFO (RIF) or Message Information Block (MIB) RAM is identified during Memory Built In Self Test (MBIST), the error address reported for the FEC by the Memory Error Management Unit (MEMU) is not correct.

Workaround:

When a FEC channel MBIST error is reported, if the base address is 0x4F0A0000, the error is reported from FEC RIF channel; if the base address is 0xFC0B0000, the error is reported from FEC MIB channel.

Workaround for FEC RIF RAM reported MBIST error:

If the MIBST error address reported from RIF channel is err_addr_reported[31:0], the real error address could be recalculated by right shifting the err_addr_reported[9:3] for one bit, and insert 1'b1 at err_addr_reported[9]. The correct err_addr equation is:

err_addr_new [31:0] = {err_addr_reported[31:10], '0b1' err_addr_reported[9:3], '0b00'}

An example,

Given an error address reported by MEMU:

err_addr_reported[31:0]=0x4F0A0008=0100 1111 0000 1010 0000 0000 0000 1000

the actual error address should be:

err_addr_new[31:0]= 0x4F0A0204=0100 1111 0000 1010 0000 0010 0000 0100

Workaround for FEC MIB RAM reported MBIST error:

When a FEC MIB channel MBIST error is reported, if the base_address is received as 0xFC0B0000, the correct value is considered as 0xFC0B0200. The correct err_addr 'err_addr_new[31:0]' can be derived from the reported 'err_addr_reported[31:0]' by using this equation:

err_addr_new[31:0] = {err_addr_reported[31:10], '0b10', err_addr_reported[8:3], '0b00'}

An example,

Given an error address reported by MEMU:

err_addr_reported[31:0]=0xFC0B0008=1111 1100 0000 1011 0000 0000 0000 1000

the actual error address should be:

err_addr_new[31:0]= 0xFC0B0204=1111 110 0000 1011 0000 0010 0000 0100

1.4 ERR003877: MC_ME: ME_IMTS[S_DMA] gets set instead of ME_IMTS[S_NMA] on a mode change request to non existing mode

Description:

If software attempts a mode change request to a non-existing mode, i.e. reserved configurations in the Mode Entry Mode Control register (ME_MCTL[TARGET_MODE]), the Mode Entry Invalid Mode Transition Status register's Disabled Mode Access status bit (ME_IMTS[S_DMA]) may be incorrectly set, instead of the Non-existing Mode Access status bit (ME_IMTS[S_NMA].

Workaround:

Software should either not select invalid modes or handle the setting of the disabled mode status and the non-existing mode status as the same condition.

1.5 ERR003878: MC_ME: Flash low-power modes cannot be activated during MC_ME mode transitions.

Description:

The flash does not support low-power modes. Therefore, if software configures a device mode in the Mode Entry module (MC_ME) to put the flash into its low-power or power-down mode by setting the FLAON field in the corresponding Mode Configuration register (ME_"mode"_MC) to 0b01 or 0b10, the mode transition to that mode will fail, and the device will hang.

Workaround:

Software must always configure the flash to be in its normal mode by setting ME_"mode"_MC[FLAON] = 0b11 for all device modes.

Note that this must be actively done for HALT0 and STOP0 modes, since the default values of ME_HALT0_MC[FLAON] and ME_STOP0_MC[FLAON] after reset are 0b10 and 0b01, respectively. For all other modes, the default value after reset is already 0b11.

1.6 ERR003879: MC_ME: Wakeup from HALT0/STOP0 modes to RUNx may get stalled

Description:

When a mode transition request from RUN[0:3] to HALT0/STOP0 modes is attempted which would result in a different system clock and different clock source configurations in the destination mode than are currently used, it is possible that the Mode Entry Module (MC_ME) will stop the clock sources while clock switching by the Clock Generation Module (MC_CGM) is in progress.

Workaround:

This mode transition issue can be avoided by using the following workaround:

Step 1. Change the RUNx system clock to match the HALT0/STOP0 system clock. Software requests for a mode change from RUNx to RUNx with system clock configuration ME_RUNx_MC[SYSCLK] same as ME_<HALT0/STOP0>_MC[SYSCLK].

Step 2. Transition from RUNx to HALT0/STOP0

Software requests for a mode transition from RUNx to HALT0/STOP0 with target clock source configuration (ME_<HALT0/STOP0>_MC[PLL1ON, PLL0ON, XOSCON and IRCON]) for HALT/STOP0 modes keeping SYSCLK the same.

Step 3. After wake up, return to RUNx mode with the same system clock that was used in HALT0/STOP0 mode. Software requests a mode change from RUNx to RUNx with the original target clock source configuration and system clock in a single mode transition attempt.

1.7 ERR003881: MC_CGM: Core 2 must be used to change the clock ratio between the cores and the Cross Bar (XBAR) interfaces

Description:

The Peripheral I/O Processor (IOP/Core 2) is the only core that can successfully change the clock ratio between the cores and the Cross Bar (XBAR) interfaces (including the peripheral bridge frequencies). Therefore only core 2 should be used to modify the System Clock Divider Configuration Registers (CGM_SC_DC 0, 1, & 2). These registers should not be changed while Core 0, core 1 or the HSM (Hardware Security Module) is performing a bus transfer through the interface that is changing frequency.

Workaround:

If only Core2 is enabled by the Boot Assist Flash (BAF), then the CGM_SC_DC 0, 1, & 2 registers can be changed by core 2 prior to enabling the other cores, including the HSM, FlexRay, FEC (Fast Ethernet Controller), DMA (Direct Memory Access), and the SIPI (Serial Interprocessor Interface).

However, if the HSM, Core 0 or Core 1 are enabled in the BAF , then the following should be taken into account:

1. Switch clock ratios using only Core 2 and ensure Core 0 and Core 1 are not accessing the bus during the clock ratio switch.
2. If the HSM is enabled but idle, clock ratios between cores and rest of the system can be changed using core 2 with core0 and core 1 disabled or idle.
3. The HSM must never run faster than the Slow XBAR. Therefore, one should change its clock ratio prior to change the Slow XBAR clock ratio. The HSM must be running from its local SRAM when the MCG register is being changed by core 2.
4. While writing the CGM_SC_DC_0, no access can be done on the Fast XBAR by any core (other than Core 2).
5. While writing the CGM_SC_DC_1, no access can be done on the Slow XBAR by any core (other than the Core 2).

1.8 ERR003903: SIUL2: The Output Drive Control bits have no effect on the output drive when DSPI output functionality is selected for a pin

Description:

When the Deserial Serial Peripheral Interface (DSPI) output functions are selected for a pin, the output buffer enable for the pin is controlled by the DSPI module.The Output Drive Control bits in the Multiplexed Signal Configuration register (SIUL2_MSCRx[ODC]), therefore, have no effect on the output drive.

Workaround:

For future compatibility, the DSPI pin should be configured in the MCSR as required for the application. However, it is the user's responsibility to insure that this configuration agrees with the mode forced by the DSPI module.

1.9 ERR003921: FCCU: Recoverable Fault Status Register 1, bit 19 (FCCU_RF_S1[19]) returns invalid data when checker core0 is disabled

Description:

The Fault Collection and Control Unit (FCCU) Recoverable Fault Status Register 1, bit 19 (FCCU_RF_S1[19]) returns invalid data when checker core (lock-step core with core 0) is disabled and therefore that register bit may not contain its specified reset value of 0x0.

Workaround:

Ignore the data read from FCCU_RF_S1[19] when checker core0 is disabled, or enable checker core0 prior to reading FCCU_RF_S1[19].

1.10 ERR003922: SSCM: Peripheral bus accesses to disabled DSPI, PIT or I2C modules causes device to hang if the SSCM EEROR [RAE] bit is 0

Description:

If the Register Bus Abort Enable bit (SSCM ERROR[RAE]) in the System Status and Control Module Error register is set to 0, and any of the DSPI (Deserial Serial Peripheral Interface), PIT (Periodic Interrupt Timer) or I2C (Inter-Integrated Circuit) modules are disabled, any internal bus accesses to a disabled DSPI, PIT or I2C module will cause the MCU to hang.

Workaround:

Do not access the DSPI, PIT or I2C modules when the corresponding module is disabled and SSCM ERROR[RAE] = 0.

1.11 ERR003963: MC_RGM: CAN RAM controller reset does not reset other CAN modules

Description:

The Reset Generation Module (RGM) peripheral reset for the CAN RAM controller (RGM_PRST2[CAN_RAM_CTRL0_RST]) is independent of the other CAN module peripheral resets (RGM_PRST2[TTCAN_A0_RST, MCAN_*_RST]) and thus the CAN RAM controller may be reset without forcing a reset of the other CAN modules. The required function will be to have hardware also force a peripheral reset of all the CAN modules whenever the CAN RAM controller is reset via the RGM peripheral reset.

Workaround:

Whenever the CAN RAM controller is reset, software should also set the peripheral resets for the other CAN modules at the same time. These are TTCAN_RST, MCAN_1_RST, MCAN_2_RST and MCAN_3_RST. All the CAN resets are located together in the RGM_PRST2 register.

1.12 ERR003970: NAR: Trace messages include a 6-bit Source Identification field instead of 4- bits

Description:

The source field (SRC) of trace messages from the Nexus Aurora Router are 6-bits in length. All other clients implement a 4-bit SRC field. Per the IEEE-ISTO 5001 Standard (Nexus) the SRC field of all clients on a device should be the same length. The two most significant bits of the SRC are 0b00.

Workaround:

Tools should treat the SRC field as a 4-bit field for all Nexus clients. In addition, tools should ignore the extra 2-bits as an extra field with no meaning. In the case of the NAR Error Message (TCODE=8), these two bits are between the 4-bit SRC field and the 4-bit Error Type (ETYPE) field. For the NAR Watchpoint Message, these bits are between the 4-bit SRC and the 6-bit Watchpoint Hit (WPHIT) field.

1.13 ERR004012: GTM: Bus interface blocked by two consecutive write commands to the BRIDGE_MODE register

Description:

(GTM-IP-62)

Two consecutive write commands to the Generic Timer Module (GTM) Top Level Configuration register BRIDGE_MODE can result in a missing transfer acknowledge or complete signal on the AEI bus interface. This causes the bus interface to be blocked until the GTM Integration Module soft reset signal is applied by setting the AEI interface soft-reset control bit in the GTM AEI Control Register [GTM_AEI_CR(AEI_SRST)].

Workaround:

Workaround 1:

Do not execute 2 consecutive write commands to the BRIDGE_MODE register. Any combinations of the 2 write commands can be collapsed in to 1 write command.

Workaround 2:

Wait for a minimum of 12 peripheral bus clock cycles to pass before the second write is issued. This will guarantee that the first write command is fully executed and the AEI transaction buffer is empty before the second write is accepted.

1.14 ERR004048: PAD_RING: Pin drive type (CMOS/OD/LVDS) ignores the ILS setting in MSCR.

Description:

The Input Level Select (ILS) and Output Drive Control (ODC) bit fields in the Multiplexed Signal Configuration Register (MSCR) are not used to select LVDS operation. Instead, the user must write to the Source Signal Select (SSS) bit field to enable LVDS operation on both the positive (P) and negative (N) ports of the LVDS function. Once the LVDS function is selected, the port input or output levels are forced to LVDS and all ILS and ODC settings are ignored.

When the LVDS Fast Asynchronous Serial Transmission (LFAST) signals are enabled, the output buffer can be disabled by writing to the pin control register in the LFAST module.

Workaround:

Set port input or output levels to LVDS by writing to the SSS. Do not expect the ILS or ODC setting to affect the LVDS input or ouput levels.

1.15 ERR004122: FLASH: Read Reference Monitor not implemented

Description:

The Flash Read Reference Error (RRE) Monitor, which is a common mode safety item, is not implemented in the flash. Therefore, in the unlikely event of a common mode issue corrupting reads, the RRE bit in the flash Module Configuration Register will never be set (1).

Workaround:

Other safety methods may be used to ensure safety component of reads. This includes doing system or software CRC (Cyclic Redundancy Check) reads, to ensure data integrity. Do not expect the MCR[RRE] bit to be set.

1.16 ERR004124: FLASH: Read Pump Voltage monitor not implemented

Description:

The Read Pump Voltage Monitor, which is a common mode safety item, is not implemented in the Flash. Therefore, in the unlikely event of a common mode issue corrupting reads, the Read Voltage Error (RVE) bit in the flash Module Configuration Register will never be set (1).

Workaround:

Other safety methods may be used to ensure safety component of reads. This includes doing system or software CRC (Cyclic Redundancy Check) reads, to ensure data integrity. Do not expect MCR[RVE] to be set.

1.17 ERR004136: XOSC and IRCOSC: Bus access errors are generated in only half of non- implemented address space of XOSC and IRCOSC, and the other half of address space is mirrored

Description:

Bus access errors are generated in only half of the non-implemented address space of Oscillator External Interface (40MHz XOSC) and IRCOSC Digital Interface (16MHz Internal RC oscillator [IRC]). In both cases, the other half of the address space is a mirrored version of the 1st half. Thus reads/writes to the 2nd half of address space will actually read/write the registers of corresponding offset in the 1st half of address space.

Workaround:

Do not access unimplemented address space for XOSC and IRCOSC register areas OR write software that is not dependent on receiving an error when access to unimplemented XOSC and IRCOSC space occurs.

1.18 ERR004171: SIUL2: MSCR register byte and halfword accesses are not allowed

Description:

The System Integration Unit Multiplexed Signal Configuration Registers (SIUL2_MSCR) are only accessible as 32-bit registers. Byte and half-word write accesses are not supported.

Workaround:

Always access the SIUL2_MSCR registers using word (32-bit) writes.

1.19 ERR004182: SARADC: Trigger signals on analog watchdog threshold crossover events not implemented

Description:

The signals indicating a threshold crossover event on a Successive Approximation Register Analog to Digital Converter (SARADC) are not implemented.

These signals are meant to be configured as trigger inputs for the Generic Timer Module (GTM) though the System Integration Unit Multiplexed Signal Configuration registers (SIUL2_MSCRx).

Workaround:

Do not use the threshold crossover event triggers.

1.20 ERR004183: SDADC: Wraparound mechanism for input multiplexer not implemented

Description:

The hardware wraparound mechanism for the Analog channel Multiplexer for the Sigma Delta Analog to Digital Converter (SDADC) is not implemented.

As the consequence the following features are not available:

- Control of the hardware sequence of the channel conversions, automatic switching of input multiplexer on each valid trigger
- Selection of the trigger sources (both Software and Hardware) during wraparound mode
- Configuration of the initial and wraparound values of the wraparound counter.

Workaround:

Do not use the SDADC hardware wraparound mechanism features.

The application software must not write:

- The wraparound mode enable bit (WRMODE) in the SDADC Module Configuration Register (SDADCx_MCR)
- The Analog Channel Selection Wraparound Value field (ANCHSEL_WRAP) in the Channel Selection Register (SDADCx_CSR)

In addition, the value to the Analog Channel Selection Counter field (ANCHSEL_CNT) of the Status Flag Register (SDADC_SFR) should be ignored.

1.21 ERR004184: SDADC: Range watchdog not implemented

Description:

The Sigma Delta Analog to Digital Converter (SDADC) range watchdog functionality is not implemented on the device.

Workaround:

Do not use the SDADC watchdog range feature.

The application software must not write the Watchdog Enable bit in the Module Configuration Register (SDx_MCR[WDGEN]) or write the watchdog threshold registers (SDx_WTHHLR). In addition, the Watchdog Upper/Lower Threshold Cross Over Event status bits (WTHH, WTHL) in the Status Flag register (SDx_SFR) should be ignored.

1.22 ERR004188: NXMC: Limited Trace Configuration During Reset

Description:

During reset, the clocks to the Nexus Crossbar Multi-Master Client (NXMC) modules are disabled. Therefore, configuration of the NXMC registers must be done after reset is negated.

Workaround:

Perform NXCM configuration for trace of the Crossbar clients after reset is negated. Note that the cores can be configured to exit the reset state in a stopped mode. This would allow the NXMC registers to be configured prior to any code execution. The cores can then be allowed to begin code execution.

1.23 ERR004207: MEMU: Error buffer overflow flags of the Memory Error Management Unit are never asserted

Description:

The Memory Error Memory Management Unit's (MEMU) error buffer overflow flags MEMU_ERR_FLAG[PR_EBO], MEMU_ERR_FLAG[F_EBO] and MEMU_ERR_FLAG[SR_EBO] are

not connected to the overflow signals at the input of the MEMU block. Therefore, any overflow that occurs if the MEMU is running at a lower frequency clock than the peripherals will not be reported by these flags.

These flags are intended to indicate when the internal error processing buffer of peripheral RAM (PR), Flash (F) or System RAM (SR) MEMU reporting block have overflowed.

Workaround:

Run the MEMU at a frequency equal or higher than the error reporting peripherals. Do not rely on error reporting from peripherals that run at higher frequencies than the MEMU.

1.24 ERR004211: MEMU: Memory Error Management Unit (MEMU) cannot handle sources running at higher frequencies than MEMU

Description:

Memory Error Management Unit (MEMU) cannot handle sources running at higher frequencies than the MEMU. Errors reported from peripherals that run at higher frequencies than the MEMU might be missed and not stored to the MEMU's reporting tables.

Workaround:

Run the MEMU at equal or higher frequencies than the error reporting peripherals. Do not rely on error reporting from peripherals that run at higher frequencies than the MEMU.

The peripheral clocks are set via the Clock Generation Module (CGM) Auxiliary Clock Select Control Registers (CGM_ACx_SC) and the Divider Configuration Registers (CGM_ACx_Dx).

The MEMU runs at the same frequency as the cores. The clock for the cores is set in the Mode Entry Mode Configuration Register System Clock Select field (MC_ME_xxxx_MC[SYSCLK] where xxxx is the operating mode: RESET, TEST, SAFE, DRUN, RUN0, RUN1, RUN2, RUN3, HALT0, or STOP0) and Clock Divider Configuration 0 (CGM_SC_DC0).

1.25 ERR004223: IMA: IMA is not implemented

Description:

The Indirect Memory Access module (IMA) has not been implemented.

Workaround:

Software should not access or attempt to use any of the features of the IMA. The IMA is located at address 0xFFF5_4000 through 0xFFF5_7FFF.

1.26 ERR004227: PMC: Temp Sensor User Adjust register size is 5 bits and should be 4 bits.

Description:

The Temperature Sensor Configuration Register Trim Adjust Under Trim CTL_TD[TRIM_ADJ_UNDER[ ]] and Over Trim CTL_TD[TRIM_ADJ_OVER[ ]] bit fields only support 4 bits.

Workaround:

Values written by the user to CTL_TD[TRIM_ADJ_UNDER[4:0]] and to CTL_TD[TRIM_ADJ_OVER[4:0]] must not be lower than -7 nor greater than +7. CTL_TD[TRIM_ADJ_UNDER[4]] and CTL_TD[TRIM_ADJ_OVER[4]] must both be programmed to 0.

1.27 ERR004234: STCU2: Start-up self test failure does not prevent application from starting

Description:

If an unrecoverable fault is detected during the start-up self test by the Fault Collection and Control Unit (FCCU) that is controlled by the Self-Test Unit (STCU2), the MCU will exit reset and attempt to execute application code instead of rerunning the self test and stay in reset if the self-test fails a second time. This could result in unpredictable behavior or even in the device hanging.

Workaround:

Application code should check for unrecoverable faults and be prepared to handle exceptions caused by the fault. These may include illegal access, illegal instructions, or bus errors.

1.28 ERR004238: MC_RGM: Peripheral Reset of FCCU not Implemented

Description:

The Fault Collection and Control Unit (FCCU) reset bit in the Reset Generation Module Peripheral Reset register 5 (RGM_PRST5[FCCU_RST], bit 22) and the register protection mechanism is not implemented. Therefore, the FCCU module cannot be reset except through a full MCU reset.

Workaround:

If the FCCU needs to be reset, reset the whole MCU.

1.29 ERR004241: PASS: Pseudo Random Clock Divider not Implemented on CAN and FlexRay Clocks

Description:

The Pseudo Random Clock Divider (PRCD) clock jitter injector on the CAN and FlexRay clocks is not implemented. Also, the on-chip oscillator (XOSC) can always be selected to source the FlexRay protocol clock. This means that off-chip communication via these modules cannot be prevented when the device is in "production disable" mode.

Workaround:

Do not expect the CAN and FlexRay clocks to be random or that the XOSC selection will be disabled, if the device is set to "production disable" mode with a Device Configuration Format setting in the UTEST Flash.

1.30 ERR004242: MC_CGM: System Clock Divider Configuration Update cannot be Aligned with Software Trigger

Description:

The Divider Update Type register (CGM_DIV_UPD_TYPE) and Divider Update Trigger register (CGM_DIV_UPD_TRIG) required for enabling the clock divider update alignment via a software trigger as well as the Divider Update Status register (CGM_DIV_UPD_STAT) are missing in the Clock Generation Module (MC_CGM). Accessing these registers results in an access error.

This also means that each system clock divider's configuration is always updated immediately when the corresponding System Clock Divider Configuration register (CGM_SC_DCn for divider n) in the MC_CGM is written. The result is that the dividers do not change their configurations together but rather one at a time.

Workaround:

Software should not access the CGM_DIV_UPD_TYPE, CGM_DIV_UPD_TRIG, and CGM_DIV_UPD_STAT register locations.

In addition, software must take extra care to ensure not only that the final system clock divider values are compatible but also that the intermediate values are compatible. In order to simplify this, it is recommended that only simple division factor relationships are used (e.g., powers of 2 if possible) and that the CGM_DC_DCn registers are written by software in the order of increasing resultant frequency.

For example, based on a 400 MHz system clock frequency:

1. Write CGM_DC_DC2 0x80070000 - AIPS clock frequency = 50 MHz
2. Write CGM_DC_DC1 0x80030000 - slow XBAR clock frequency = 100 MHz
3. Write CGM_DC_DC0 0x80010000 - fast XBAR clock frequency = 200 MHz

1.31 ERR004245: SPU: Divide by 4 operating mode is not implemented

Description:

The ability to change the frequency divider (DIV) for the Sequence Processing Unit (SPU) state processing operation is not implemented in the SPU Enable Register (SPU_SE). Updating the SE register DIV bit will have no effect. SPU operation is fixed in “Div1" mode.

Workaround:

Be aware that SPU counter operations will not be available for state evaluation immediately in the next SPU clock cycle, and special handling must be done in state evaluation during this delayed effect to assure that it does not create undesirable results. To workaround this hardware limitation, a delay may require to be inserted between the update to a counter and the evaluation of this counter.

Solution1: A dummy state can be used between the counter update (State n) and the counter evaluation (State n+1). A dummy state is a state with any input, that on both then/else conditions, the action proceeds to the next state (State n+1) where the counter evaluation occurs. The insertion of this dummy state provides sufficient time for the update of the counter to be applied, in time for the counter evaluation.

Solution2: In a complex sequence with multiple states, it is also possible to achieve the same outcome by scheduling states in a manner such that the counter update and counter evaluation are not performed on consecutive states in a sequence.

1.32 ERR004248: MC_RGM: Illegal Register Access will not generate access error

Description:

Any read or write access to unused registers or any write access to read-only registers in the Reset Generation Module (MC_RGM) module will not generate an access error.

Workaround:

Do not expect illegal access to unused or read-only registers of the MC_RGM to cause an exception.

1.33 ERR004249: MC_RGM: 'Destructive' Reset Escalator not Implemented

Description:

Since the 'destructive' reset escalator is not implemented, access to the Destructive Reset Escalation Threshold register (RGM_DRET) in the Reset Generation module (MC_RGM) will cause an access error. In addition, one of the following may occur:

- Reset cycling due to recurring 'functional' reset events that cannot be corrected by a 'destructive' reset
- The chip exits reset in an unknown state after a start-up self test failure, potentially resulting in the chip hanging

Workaround:

To avoid an access error, do not access the RGM_DRET register.

There is no workaround for the reset cycling or start-up self-test failure.

1.34 ERR004250: MC_RGM: ESR0 Deassertion Cannot be Controlled by Software

Description:

The Reset Generation Module External Reset Output Extension control register (RGM_EROEC) is not implemented and will cause a bus access error if accessed. Therefore, software control of the external reset pin (ESR0) to deassert ESR0 is not possible.

Workaround:

Do not use the ESR0 deassertion control feature, and do not access the RGM_EROEC register.

1.35 ERR004258: LINFLEXD: Maximum UART Baud Rate supported is 1 Mb/s

Description:

The sample rate is a fixed 16 times the baud rate for the LINFLEX module. This limits the baud rate of the serial protocol to a maximum of 1M bits/second (b/s).

Workaround:

Limit baud rates to less than 1M b/s.

1.36 ERR004260: LINFLEXD: LINIBRR & LINFBRR registers cannot be read back immediately after being written

Description:

The LINFlexD Integer and Fractional Baud Rate registers (LINFLEXD_x[LINIBRR] & LINFLEXD_x[LINFBRR]) cannot be read back until 4 system clocks after being written.

Workaround:

Wait 4 system clocks prior to reading the LINFLEXD_x[LINIBRR] and LINFLEXD_x[LINFBRR] registers after writing the registers or avoid reading the registers after they have been written.

1.37 ERR004275: PMC: The Reset Enable bits default values cannot be set in the DCF records

Description:

The Device Configuration Format records that control the default values of the Power Management Controller Reset Event Enable Registers (REE_VDn) cannot be programmed into the flash by customers.

Workaround:

The PMC_REE_VDx default values are not loaded from the flash during reset. Therefore, they will remain in the power-on reset configuration (resets enabled). Software will always be able to clear the REE bits to mask the reset from occurring based on the low voltage detect circuit (instead of the normal operation of the flash loaded enable bits being locked and unable to clear). To prevent accidental writes masking the resets, software can write the bits in these registers to a 1 to prevent accidental writes of 0 that would mask resets from the LVD.

1.38 ERR004370: SIPI: The register field SIPIMCR[PRSCLR] can be written with only a 32 bit access to the register

Description:

When writing the Prescaler field (PRSCLR) of the Serial Inter-processor Interface (SIPI) Module Control register (SIPI_SIPIMCR), the whole 32 bit register must be written. Half word writes to this register to change the PRSCLR field may lead to unexpected values being written. Other fields within the register are not effected.

Workaround:

Use only 32 bit register wide access to SIPIMCR to write the PRSCLR field. This can be done with a read modify write to ensure only the contents of the PRSCLR field are changed.

1.39 ERR004391: SSCM: Spurious SSCM activation not detected by the FCCU

Description:

A spurious activation of the SSCM (System Status and Configuration Module) will not be detected by the FCCU (Fault Collection and Control Unit).

Workaround:

Usage of the current version of the MCU in series production for safety-relevant applications is not recommended.

1.40 ERR004394: MCAN: Debug over MCAN is not supported

Description:

The 'debug over CAN' feature that allows sending and receiving of debug message over the MCAN module is not implemented.

The MCAN module does not support the debug over CAN feature, and cannot generate Direct Memory Access (DMA) request triggers. The Debug Messages State-machine (DMS) field of the RxFIFO1 Status Register is not implemented, and reads as 0.

Workaround:

The debug over CAN feature is not implemented and cannot be used. Avoid configuration of the Standard Filter Element Configuration (SFEC) and Extended Filter Element Configuration (EFEC) field of the message ID filter elements to '111'.

Debug support via the CAN bus can be implemented by CPU assistance, via debug monitor software.

1.41 ERR004399: JTAGM: Register Protection is not implemented for the JTAG Master Module Control Register (JTAGM_MCR)

Description:

Register Protection is not implemented for the JTAG Master Module Control Register (JTAGM_MCR).

Workaround:

Do not use this version of the device in Safety-critical applications or do not rely on the register protection to protect the JTAGM MCR if using the JTAGM for internal software debug mode (debug features implemented in the user application).

1.42 ERR004415: PSI5 : When a DMA request from the PSI5 is active and the PSI5 enters Debug Mode, then the data read by the DMA will be corrupted

Description:

The Peripheral Sensor Interface (PSI5) provides specific registers to be read by a Direct Memory Access (DMA) module. These registers are:

- Channel x DMA PSI5 Message Register (PSI5_CHx_DPMR)
- Channel x DMA Diagnostic Status Register (PSI5_CHx_DDSR)
- Channel x DMA SMC Frame Register (PSI5_CHx_DSFR)

The DMA reads these registers when a specific watermark is reached and in response to which, the module generates the DMA request.

Each of these registers are single 32-bit wide registers and each read of these registers from the DMA automatically refreshes the contents of the register, and makes it ready for the next read by the DMA.

The PSI5 module immediately enters Debug Mode when receiving the request. The module immediately stops refreshing these registers on each read of the DMA.

At the end of the DMA, when the request for entering debug mode is given to the DMA, then the DMA is not halted immediately but finishes its current minor loop and then goes into the Debug Mode.

Due to this difference in the behavior of the DMA and the PSI5 module, the DMA keeps on reading the contents from the listed registers even while the module has stopped refreshing them.

Moreover, when leaving the Debug Mode, the PSI5 module will maintain its DMA request (thinking the DMA has halted the read during debug mode). This would cause the DMA to start reading again the PSI5 module contents.

This anomaly between the DMA and the IP behavior in the Debug Mode would cause the corruption in the DMA read sequence and the data.

Workaround:

If there is some DMA transaction ongoing in the PSI5 module, wait for it's completion before entering the debug mode. To selectively prevent the MSI5 module from entering the Debug Mode, keep the DEBUG_EN bit as "0".

1.43 ERR004441: NZxC3: Nexus timestamps are not implemented

Description:

The Nexus clients in the e200zx cores do not implement timestamp packets on the Nexus trace messages.

Workaround:

Do not expect timestamps on the Nexus messages from the core, even though timestamps are enabled in the device.

1.44 ERR004518: GTM: reading the DPLL_RAM_INI register triggers RAM initialization

Description:

(GTM-IP-72)

Reading the Generic Timer Module (GTM) Digital PLL (DPLL) RAM Initialization register (DPLL_RAM_INI) may set bit 4 (Most Significant bit = 0) of the register, which is specified as Reserved. This read may start an unintentional DPLL RAM initialization, depending on the preceding read/write access data.

Workaround:

Before reading the DPLL_RAM_INI register, reset the Automotive Electronics Interface (AEI) bridge by setting the AEI Reset bit in the GTM AEI Control Register [GTMAEICR(AEIRST)] and read the bridge mode register [GTM_BRIDGE_MODE]. Make sure that no other read/write access occurs between reading the GTM_BRIDGE_MODE register and reading the DPLL_RAM_INI register.

1.45 ERR004520: GTM: After a GTM Bridge Reset, a read or write command can be performed twice

Description:

(GTM-IP-74)

A write to bit 16 (BRG_RST) of the Generic Timer Module (GTM) GTM_BRIDGE_MODE register will reset the GTM Bus Bridge. The following transfer (read / write) can be executed twice causing an unexpected behavior e.g. if the transfer after a GTM bridge reset is a FIFO access this will result in an incorrect FIFO queue (Write: data was issued twice to FIFO; Read: one piece of data will be lost). This failure can be activated dependent on the protocol, GTM_BRIDGE_MODE in use, buffer depth of the bridge and clock frequencies.

Workaround:

After issuing a bridge reset by writing the BRG_RST Bit to '1' the next transfer must be to a register which is not sensitive to duplicate read or write accesses. It is recommended to use a READ of the GTM_BRIDGE_MODE register.

1.46 ERR004522: GTM: TIM input signal incorrect during reset

Description:

(GTM-IP-78)

While the Generic Timer Module (GTM) Timer Input Module (TIM) is in reset, the input to the channel (TIM_CH), Timeout Detection Unit (TDU) and the TIM debug out port are all "0" regardless of the actual TIM input signal value.

Workaround:

Ensure that the device CPU, the GTM Debug Interface (GTM-DI) and the TIM0 Mapping Module (MAP) do not read any TIM registers while the TIM module is being reset. The Advanced Routing Unit (ARU) will not read the TIM registers while it is in reset as the data will not be marked as valid. There is no signal in the TIM which indicates reset activity.

1.47 ERR004523: GTM: TIM input signal incorrect while TIM channel is disabled

Description:

(GTM-IP-80)

While the Generic Timer Module (GTM) Timer Input Module (TIM) is disabled in the channel control register (TIM[i]_CH[x]_CTRL[TIM_EN] = 0), bit 0 of the edge counter field of the channel General Purpose Register (TIM[i]_CH[x]_GPRn[ECNT]) is 0 and does not reflect the actual signal level at the filter output. This means that the signal value in the general purpose registers (TIM[i]_CH[x]_GPRn) and the Shadow Counter Registers (TIM[i]_CH[x]_CNTS) will not reflect the filtered output signal value. As soon as the TIM channel is enabled the signal value of bit 0 of the ECNT bit field is correct.

Workaround:

Ensure any affected software expects bit 0 of ECNT to be zero when the TIM is disabled.

1.48 ERR004524: GTM: TIM counter value incorrect when in TPWM one shot mode

Description:

(GTM-IP-85)

After the Generic Timer Module (GTM) Timer Input Module (TIM) channel has stopped in TIM PWM Measurement (TPWM) one shot mode, the counter register, TIM[i]_CH[x]_CNT, has the value 1. The correct expected value of the counter should be 0.

Workaround:

Reset the TIM Channel by setting the Software Reset Channel bit (RST_CHn) in the TIM Reset Register (TIM[i]_RST) register and then reconfigure the TIM channel after a completed compare event when TPWM one shot mode measure is used.

1.49 ERR004525: GTM: When the TIM timeout event occurs, the ARU signal level is not updated

Description:

(GTM-IP-90)

If the Generic Timer Module (GTM) Advanced Routing Unit (ARU) is enabled in the Timer Input Module Control Register (TIM[i]_CH[x]_CTRL[ARU_EN=1) and a new ARU transfer is initiated due to a Timeout Detection Unit (TDU) event, the ARU input signal level bit is not updated to the actual input signal level.

Workaround:

Ensure that the destination for the TIM data via the ARU does not depend upon the signal level for correct operation.

1.50 ERR004526: GTM: TIM GPRz_OFL bit set when ARU is enabled and the GPR capture event and ARU are serviced at the same time

Description:

(GTM-IP-91)

If the Generic Timer Module (GTM) Advanced Routing unit (ARU) routing is enabled by setting ARU_EN=1 in the Timer Input Module (TIM) Control Register (TIM[i]_CH[x]_CTRL), and an ARU transfer is serviced during the same system clock cycle as a new capture event takes place, the General Purpose Register Overflow flag in the TIM Channel IRQ Notify register [TIM_CH_IRQ_NOTIFY(GPRz_OFL)] is set, no update of the ARU input signal level bit to the actual input value is done.

Workaround:

Do not use ARU routing if the input signal change time is less than ARU round trip time.

1.51 ERR004568: MC_CGM: DE of the CGM_SC_DCn are writable to 0

Description:

The Divider Enable bits (DE) of the System Clock Divider Configuration registers (CGM_SC_DCn) are not read-only but, rather, writable. Therefore, writing a 0b0 to these bits will cause the corresponding system clock divider to be disabled and, subsequently, the corresponding divided system clock to be stopped. This will cause the cores and buses connected to these clocks to stop functioning.

Workaround:

Always write 0b1 to the DE bit when writing to any of the CGM_SC_DCn registers.

1.52 ERR004582: SIPI: Module must be in INIT mode to modify Channel Control Register

Description:

Each of the four Serial Inter-processor Interface (SIPI) channels has a Channel Control Register (CCRn) for selecting read commands, write commands, or trigger commands and the size of the transfer. The current implementation requires the module to be in INIT mode (SIPIMCR[INIT] = 1) to program these registers.

Workaround:

Prior to changing the Channel Control Register values, check the channel busy status of all the four channels and then place the SIPI module into INIT mode.

1.53 ERR004583: SIPI: Channel 2 priority is too high

Description:

Channel 2 of the Serial Inter-processor Interface (SIPI) has the highest priority of all of the SIPI channels. Channel 0 and channel 1 should both have a higher priority than channel 2. Channel 2 supports streaming. and it is possible that the streaming feature could consume most of the bandwidth of Inter-processor communication Interface (over the LVDS Fast Asynchronous Serial Transmission [LFAST] module). Since channel 2 implements the highest priority, it is possible that all other channels will be blocked.

Workaround:

Expect that Channel 2 of the SIPI has the highest priority and limit the use of the streaming mode to insure that the other channels receive bandwidth of the inter-processor interface.

1.54 ERR004600: GTM: TIM TGPS mode Edge Counter reset not functional

Description:

(GTM-IP-81)

When the Generic Timer Module (GTM) Timer Input Module (TIM) channel is in Gated Periodic Sampling Mode (TGPS) the edge counter reset (ECNT_RESET) bit in the TIM[i]_CH[x]_CTRL register has no function. Setting ECNT_RESET does not reset the TIM edge counter.

Workaround:

Do not reset the TIM Edge Counter (ECNT) using the ENT_RESET bit. The whole channel can be reset by setting RST_CHx in the TIM[i]_RST register.

1.55 ERR004601: GTM: TIM TGPS mode delayed start of operation

Description:

(GTM-IP-82)

When the Generic Timer Module (GTM) Timer Input Module (TIM) channel is configured in Gated Periodic Sampling (TGPS) mode with ISL=0 (Ignore Signal Level not set), the counting of clock events in the counter register TIM[i]_CH[x]_CNT starts with the first input signal edge defined by the signal level bit (DSL) in TIM[i]_CH[x]_CTRL register and not immediately after enabling the channel when the input signal has the value defined by DSL.

Workaround:

Only enable the TIM Channel in TGPS mode with ISL=0 when the input signal TIM_IN(x) is the inverse state of the DSL setting. Alternatively, ignore the first resultant count of edges indicated by the first TIM[i]_NEWVAL[x]_IRQ as this value could be inaccurate.

1.56 ERR004602: GTM: TIM TGPS mode input signal delayed by one system clock

Description:

(GTM-IP-83)

When the Generic Timer Module (GTM) Timer Input Module (TIM) channel is in Gated Periodic Sample (TGPS) mode, the input signal to the channel is delayed by one additional system clock before it is used for gating the TIM[i]_CH[x]_CNT counter. This will result in a deviation of 1 if Time Base Unit (TBU) values running from the system clock are used for capturing in the TIM[i]_CH[x]_GPRn registers.

Workaround:

Take this deviation of 1 in to account in the GTM system. It can be added back on to the captured values by the Mutli Channel Sequencers (MCS) but that delays the data availability.

1.57 ERR004603: GTM: TIM TGPS one shot mode counter value incorrect

Description:

(GTM-IP-84)

After the Generic Timer Module (GTM) Timer Input Module (TIM) channel is stopped in Gated Periodic Sampling (TGPS) one shot mode in the TIM Channel Control register (OSM=1 in TIM[i]_CH[x]_CTRL) the counter register TIM[i]_CH[x]_CNT has the value 1. The correct expected value is 0.

Workaround:

Ensure that software does not read TIM[i]_CH[x]_CNT when the channel is in TGPS OSM mode. The channel is enabled again by setting TIM_EN in TIM[i]_CH[x]_CTRL to 1 which correctly resets the counter to 0.

1.58 ERR004604: GTM: TIM TPWM mode with external capture and ARU enabled data incorrect

Description:

(GTM-IP-86)

If the Generic Timer Module (GTM) Timer Input Module (TIM) is configured in Pulse Width Modulation (PWM) Measurement Mode (TPWM) with Advanced Router Unit (ARU) connection enabled (ARU_EN=1) and External Captures Enabled (EXT_CAP_EN=1) in TIM[i]_CH[x]_CTRL, the value in the ARU data word is incorrect.

Workaround:

Do not use the TIM channel in this mode combination if the system depends on the ARU words accuracy.

1.59 ERR004605: GTM: TIM TBCM mode with external capture edge counter not functional

Description:

(GTM-IP-87)

If the Generic Timer Module (GTM) Timer Input Module (TIM) is configured in Bit Compression Mode (TBCM) with Advanced Router Unit (ARU) connection enabled (ARU_EN=1) and External Captures Enabled (EXT_CAP_EN=1) in TIM[i]_CH[x]_CTRL, the edge counter (ECNT) is not functional. Only bit 0 toggles on an input vector change, all other bits stay 0. The embedded ECNT values in TIM[i]_CH[x]_GPRn and TIM[i]_CH[x]_CNTS show the same erroneous behavior.

Workaround:

Do not use TBCM mode with ARU_EN and EXT_CAP_EN set.

1.60 ERR004607: GTM: TIM TBCM mode with external capture not functional

Description:

(GTM-IP-88)

If the Generic Timer Module (GTM) Timer Input Module (TIM) is configured in Bit Compression Mode (TBCM) with External Captures Enabled (EXT_CAP_EN=1) in TIM[i]_CH[x]_CTRL, an external capture event does not capture data in the registers (TIM[i]_CH[x]_GPRn) and does not issue a NEWVAL interrupt. If ARU_EN=1 is set, no ARU transfer is initiated on an external capture event.

Workaround:

Do not use the TIM channel in external capture TBCM mode.

1.61 ERR004608: GTM: TIM TGPS mode with external capture not functional

Description:

(GTM-IP-89)

When the Generic Timer Module (GTM) Timer Input Module (TIM) channel is in Gated Periodic Sample (TGPS) mode with External Captures Enabled (EXT_CAP_EN=1) in TIM[i]_CH[x]_CTRL, the edge counting functionality is not operational.

Workaround:

Do not use the TIM channel in this configuration.

1.62 ERR004609: GTM: MCFG RAM Borrow Mode may read incorrect data

Description:

(GTM-IP-97)

If the Memory Configuration (MCFG) submodule of the Generic Timer Module (GTM) configures an Multi-Channel Sequencer (MCS) in borrow mode and the corresponding MCS then tries to read data from that borrowed RAM area, the wrong data is read.

Workaround:

Do not use the MCFG borrow mode. The use of swap mode is not affected.

1.63 ERR004610: GTM: DPLL makes incorrect direction decision when the 24-bit TBU wrap around occurs between an active slope and the following inactive slope

Description:

(GTM-IP-104)

If a wrap around (transitions from 0xFF_FFFF to 0x00_0000) of the Generic Timer Module (GTM) 24- bit Time Base Unit (TBU) time base value which is assigned to the Digital PLL (DPLL) occurs, and this event happens after an active slope and before the following inactive slope, the DPLL incorrectly indicates and reacts to the false direction change.

Workaround:

Do not use the TBU time base in the DPLL. The Clock Management Unit clock CMU_CLK0 is unaffected by this erratum.

1.64 ERR004611: GTM: DPLL calculates an action which was to occur in the past and delivers the wrong timestamp.

Description:

(GTM-IP-105)

When the Generic Timer Module (GTM) Digital Phase Lock Loop (DPLL) calculates an action which was to occur in the past, the wrong time stamp will be given by the DPLL for the Time Stamp of Action Calculated (TSAC) value. This happens if the calculated action runs into the past during a direction change condition. The correct output data in this case would be the time value of the last active event on the DPLL input. When the action is calculated to occur in the past, the output value will be not the latest timestamp but an older value. This is not different from the correct output data because when the timestamp of the DPLL output is in the past an action will be executed immediately. This is independent of the wrong time value.

Workaround:

No workaround identified.

1.65 ERR004612: GTM: DPLL Wrong evaluation of action which occurs past

Description:

(GTM-IP-106)

The Generic Timer Module (GTM) Digital PLL (DPLL) wrongly calculates an action to occur in the past when the Time Base Unit (TBU) wraps around its 24-bits. This means that an action calculation is completed as "past" although the action calculation should still be active. This behavior occurs when the Time Stamp Field of Trigger Events TSF_T(p+m) and TSF_T(p) of Action Calculation of Trigger Forwards/Backwards equations 16.11a2, 16.11a4, 16.13a2 and 16.13a4, or the Time Stamp Field of State Events TSF_T(p+m-n) and TSF_T(p-n) of Action Calculation of State Forwards/Backwards equations 16.11a1, 16.11a3, 16.13a1, 16.13a3 span the 24 bit wrap around of the TBU_CH0 time stamp. This causes early and incorrect action output data.

Workaround:

The timestamp field in DPLL RAM memories RAM1c, RAM2 can be monitored and the action calculation can be disabled by clearing the action enable bits when the conditions are such that the timestamps are used for calculations where there is a 24-bit overrun between both relevant values.

1.66 ERR004613: GTM: DPLL Wrong PVT check evaluation

Description:

(GTM-IP-107)

In the Generic Timer Module (GTM) Digital PLL (DPLL) submodule the wrong Plausibility Value of next Trigger (PVT) check evaluation is indicated through the Plausibility Window Violation Interrupt (PWI) because the PVT evaluation is not skipped for two events (one increment) when an unexpected missing trigger occurred before. The direction decision and then change occurs too early or incorrectly when an unexpected missing trigger occurred before.

Workaround:

Do not use the PVT value or the PWI feature of the DPLL. Disable the interrupt source (PWI_IRQ_EN = 0) in the DPLL_IRQ_EN register.

1.67 ERR004697: TDM: Diary base address must be 16KB aligned

Description:

In the Tamper Detection module (TDM), the Diary base address must be 16KByte aligned. Assigning the Diary Base Address to a non 16KByte will cause the diary to behave as if the base address is 16KB aligned with respect to the different diary region and the diary select.

Workaround:

Set the base address of the Tamper Detect Module Diary to a 16KB aligned region.

1.68 ERR004764: SSCM: Spurious reset protection missing

Description:

Spurious reset protection for the System Status and Configuration Module (SSCM) is not implemented.

Spurious resets may corrupt the SSCM and could impede Flash operations or interfere with the device configuration.

Workaround:

Do not rely on the spurious reset protection and ensure a high enough safety level by other means.

1.69 ERR004800: PSI5-S: PSI5-S module is not implemented

Description:

The Peripheral Sensor Interface Support module (PSI5-S) is used to interface directly to a UART compatible PSI5 transceiver over a high speed UART link. The PSI5-S module is not available in this revision of the device.

As a consequence a read or write access to the module registers will generate a bus termination error, and the associated interrupt and DMA sources are not available. System Integration Unit Lite2 (SIUL2) Multiplexed Signal Configuration Register (MSCR) Source Signal Select (SSS) bitfield values supporting PSI5-S signal multiplexing are not available. In addition, accessing the Auxiliary Clock 5 Cascaded Divider Configuration registers (CGM_AC5_CDC0, CGM_AC5_CDC1, and CGM_AC5_CDC2) in the Clock Generation Module (MC_CGM) results in an access error since these registers are not available.

Workaround:

Don't use the PSI5-S interface.

Do not access the memory range 0xFBF7_4000 - 0xFBF7_7FFF.

Do not configure the interrupt vectors 946 to 961 via their corresponding Interrupt Controller Priority Select registers (INTC_PSR946-947, INTC_PSR948_949, ... INTC_PSR960_961).

Do not enable the PSI5-S source 49 on the DMA multiplexer 2 and source 50 on the DMA multiplexer 3.

Do not program SIUL2 MSCR_SSS values for PSI5-S signal selection.

Do not access the CGM_AC5_CDC0, CGM_AC5_CDC1, and CGM_AC5_CDC2 register locations.

1.70 ERR004906: GTM: DPLL requires software reset after being disabled

Description:

(GTM-IP-99)

When the Generic Timer Module (GTM) Digital PLL (DPLL) is disabled the direction of the SUB_INC signals is not reset internally in the DPLL submodule, and hence, the DPLL_STATUS register Backwards Drive of SUB_INCx bits (BWD1 and BWD2) is not cleared when the module is re enabled. After enabling the DPLL (DEN = 1 in register DPLL_CTRL_1) the DPLL should start in forward mode but due to the missing reset of the direction state logic (as reflected in the BWD bits) the DPLL starts in backward mode again.

Workaround:

Any time the DPLL is disabled, perform a software reset (SWR = 1 in register DPLL_CTRL_1).

1.71 ERR004907: GTM: TIM external capture in TPWM and TPIM not correctly disabled

Description:

(GTM-IP-101)

In the Generic Timer Module (GTM), Timer Input Module (TIM) when external capture is enabled (EXT_CAP_EN = 1), the channel is disabled (TIM_EN = 0) and the channel is configured for PWM Measurement or Pulse Integration Mode (TIM_MODE = '000' or '001') in register TIM[i]_CH[x]_CTRL the TIM channel will capture on an external capture event. The TIM channel captures data and places it in the General Purpose Registers (GPRx) and issues an interrupt when an external capture event occurs. This functionality is incorrect because the channel is disabled (TIM_EN = 0).

Workaround:

Ensure EXT_CAP_EN is only ever set if TIM_EN= 1. This can be done by setting the bit fields EXT_CAP_EN and TIM_EN bits, in register TIM[i]_CH[x]_CTRL, atomically throughout all software.

1.72 ERR004908: GTM: MCS NARD(I) instruction not working properly

Description:

(GTM-IP-110)

The Generic Timer Module (GTM) Multi-Channel Sequencer (MCS) Suspend Channel instruction 'NARD(I)' may terminate early, with the status SAT = 0 (unsuccessful Advanced Routing Unit [ARU] transfer) even though the data source is ready to send data. In this case, the data source does not receive the request to send and thus no data is lost. This scenario happens over two cycles within an ARU round trip cycle. When NARD(I) terminates with SAT=0, the conclusion that the data source has 'no data' is unreliable.

Workaround:

Duplicate the NARD(I) instruction as follows:

```
NARD R0, R1, addr
JBS STA, SAT, go_on NARD R0, R1, addr
go_on:
...
```

and ensure, that the sequence above is executed as a whole, in other words that it is not interrupted by software (for example, enabling/disabling of MCS-channel).

1.73 ERR004909: GTM: DPLL RDT_T/S overrun in case of TBU_CH0 wrap around

Description:

(GTM-IP-111)

The Generic Timer Module (GTM) Digital PLL (DPLL) submodule does not operate correctly when the calculated value of the nominal increments DT_S_actual and DT_T_actual are larger than 23-bits (0x7FFFFF). The calculation of the reciprocal values, RDT_T and RDT_S, is performed incorrectly because of a sign overrun of the 24-bit serial divider. The result is the calculation of an incorrect negative value for RDT_S and RDT_T, which then leads to too large difference (EDT_T) and weighted difference (MEDT_T) values of the predicted values. These erroneous values result in faulty action calculation results and wrong calculation of subincrement prediction. This errata case also applies when DT_S/T_actual are smaller than 0x7FFFFF but the resultant MEDT_T/S, QDT_T/S, SYN_T, or CDT_TX value results in an overrun (more than 23 bits) of any downstream calculation. In these cases, DPLL actions are ended as they are indicated as actions in the past instead of the DPLL calculating new action output data.

This situation can be forced by setting the TBU_CH0_BASE and could happen in a real application when the car is starting out of stop in start/stop mode.

Workaround:

If DT_S or DT_T are found to be larger than 0x7FFFFF disable the DPLL module by writing DEN = 0 in DPLL_CTRL_1.

1.74 ERR004910: GTM: ATOM SOMC mode CPU controlled, change of compare strategy is ignored

Description:

(GTM-IP-112)

When the Generic Timer Module (GTM) ARU Connected Timer Output Module (ATOM) channel is configured as Signal Output Compare Mode (SOMC) and the ARU is disabled (ARU_EN=0 in ATOM[i]_CH[x]_CTRL register), and an update of the SOMC compare strategy, by the CPU, takes place before a compare match, the compare strategy change is handled incorrectly. The active compare (e.g. CCU0) will be canceled, but the new compare on the other compare unit (e.g. CCU1) may not be started. As an example: when the CPU updates the compare strategy from Compare In CCU0 Only (ACB[4:2]=010) to Compare In CCU1 Only (ACB[4:2]=011), the compare on CCU0 is cancelled, but the compare on CCU1 is not started.

Workaround:

Cancel the pending compare by using the Compare Strategy ACB[4:2]=111 (change ARU address, which has no effect when ARU_EN=0). Set up the new compare strategy by writing the required value to ACB[4:2] and then update Compare Registers ATOM[i]_CH[x]_CM0 and ATOM[i]_CH[x]_CM1.

1.75 ERR004986: FCCU: FCCU_RF_S1 bit 21 in the Recoverable Fault Status Register 1 returns invalid data out of reset

Description:

Bit 21, representing the Pad Compensation source fault, of the Fault Collection and Control Unit (FCCU) Recoverable Fault Status Register 1 (FCCU_RF_S1), returns invalid data after reset and therefore that register bit may not contain its specified reset value of 0b0.

Workaround:

Ignore the data read from FCCU_RF_S1[21] after reset since the Pad Compensation source fault is not implemented.

1.76 ERR005006: INTC: Non-safety cores interfere with each other or the safety core via manipulation of INTC operational registers

Description:

Non-safety cores can modify the interrupt configurations for other cores, including the safety core via manipulation of the Interrupt Controller (INTC) operational registers. Intentional or unintentional manipulation of other cores' INTC registers should not be allowed. Exception: cores can interrupt each other through INTC Software Set Interrupt Registers.

Workaround:

Ensure that software running on cores do not access the interrupts being used by other cores, but especially no interrupts used by the safety core.

This version is not intended to be used for series production of safety-relevant applications.

1.77 ERR005021: IAHBG: Intelligent AHB Gasket safety mechanisms missing

Description:

There are no safety measures to cover faults in the intelligent AHB gaskets (IAHBGs) which connect the computational shell and the peripheral shell to crossbars (XBARs).

Workaround:

This device revision is not intended to be used in safety-relevant series production.

1.78 ERR005023: SMPU: Missing Safety Measures

Description:

Safety measures are not provided against faults in the System Memory Protection Unit (SMPU). As a result, SMPU failures might cause valid transactions to be aborted without notifying the associated bus master. In case of a write operation, this might lead to a lost update.

Workaround:

Do not rely on the SMPU for safety critical operation.

1.79 ERR005037: CRC: CRC-32 (Ethernet) and CRC-16 (CCITT) operation do not match industry standards.

Description:

CRC-32 (Ethernet) and CRC-16 (CCITT) do not calculate CRCs according to industry standards. For CRC-32, the CRC engine expects the CRC Input to be byte swapped. For CRC-16, the CRC engine expects the CRC Input to be bit reversed.

Workaround:

The input data has to be bit reversed and entered into the CRC Input Register (CRC_INP) for CRC- 16 calculations and byte reversed for CRC-32 calculations to get the correct CRC signature.

For CRC-32, given input data of 0xABCDEF98, the user software must enter 0x98EFCDAB in CRC_INP register to get the correct result for CRC-32. Also, the Inversion (INV) and Swap selection (SWAP) bits of the CRC Configuration Register (CRC_CFG) are to be set to 1 for CRC-32 polynomial calculations along with the byte swaps.

For CRC-16, given input data of 0xABDEF98, the user software must enter 0x19F7B3D5 into the CRC_INP register. INV and SWAP bits are programmed to zero for CRC-16 operation.

1.80 ERR005038: FCCU: Entering safe mode may unexpectedly disable transmission pin of communication peripherals

Description:

If the Safe Mode Control (SMC) bit of Multiplexed Signal Configuration Register (MSCRn) in the System Integration Unit Lite2 (SIUL2) for a communication module's transnmit (TX) pin is set to "disable" (SIUL2_MSCR[SMC]=0) then it is expected that the pin will be disabled whenever the Fault Collection and Control Unit (FCCU) moves into the FAULT state and drives a fault condition on the FCCU_F[0] pin. However, the pin will be disabled when the FCCU moves into the FAULT state even if the FCCU_F[0] pin is not driven.

Workaround:

Do not use this feature (SIUL2_MSCR[SMC]) if you require the transmitting pin to only be disabled upon the FCCU moving into FAULT state and the fault condition being driven on FCCU_F[0] pin.

1.81 ERR005039: JTAGM: JTAGM_MCR[evti1_assert] flag does not correctly reflect the state of the system in software debug mode

Description:

When the JTAG Master Module (JTAGM) is in software mode (when JTAGM_MCR[DTM] == 1) then a write to "0" to the Module Configuration Register EVTI1 Assert bit (JTAGM_MCR[evti1_assert]) should not have any effect on the system and the system should NOT enter into debug mode through software. The system does NOT enter into the debug mode as expected, but the bit JTAGM_MCR[evti1_assert] does get written to a value "0". Reading this bit would not correctly reflect the state of the system.

Workaround:

In software debug mode (when JTAGM_MCR[DTM] == 1), do not clear the JTAGM_MCR[evti1_assert] bit and do not rely on its value to know the status of the system.

1.82 ERR005049: PAD_RING: Nexus Event Output B0 may not be observable on PF[15]

Description:

The PF[15] pin is configured for the slowest edge rate when the Nexus Event Output B0 function is enabled. Due to this frequency limitation, Nexus Event Output B0 may not be observable on pin PF[15].

Workaround:

Use one of the other pins (PF[14], PM[4], or PM[5]) for Nexus Event Output B0 to insure that the EVTOB0 signal is visible.

1.83 ERR005073: 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.

1.84 ERR005083: MC_ME: PLL0 and PLL1 cannot be enabled with a single mode transition

Description:

In order for each PLL to provide a clean clock, it must be turned on only when its reference clock is stable. It is intended for the Mode Entry module (MC_ME) to ensure that the reference clock source is turned on and its output has stabilized before turning the PLL on. In the case where the PLL's reference clock is either the Internal RC Oscillator (IRCOSC) or the Crystal Oscillator (XOSC), this sequencing is performed correctly. However, in the case where PLL1's reference clock is the PHI1 output of PLL0, PLL1 is turned on together with PLL0 without waiting for PLL0 to lock. The result is that PLL1 could lock to an incorrect and potentially too high frequency.

Workaround:

When transitioning from a non-low-power mode that has both PLLs off to a non-low-power mode that has both PLLs on, turn on PLL1, PLL0, and PLL0's reference clock source in two mode transition steps:

1. Set up the current mode's Mode Configuration Register (ME_<current mode>_MC) so that PLL1 is off (PLL1ON = 0b0), PLL0 is on (PLL0ON = 0b1), and PLL0's reference clock source is on (XOSCON or IRCON = 0b1) and then request a mode change to the current mode via the Mode Control Register (ME_MCTL).
2. Set up the target mode's Mode Configuration Register (ME_<target mode>_MC) register so that PLL1 is on (PLL1ON = 0b1), PLL0 is on (PLL0ON = 0b1), and PLL0's reference clock source is on (XOSCON or IRCON = 0b1) and then request another mode change to the target mode via the ME_MCTL register.

When transitioning from a non-low-power mode that has both PLLs on to a low-power mode that has both PLLs off, turn off PLL1 before entering the low-power mode and then turn it back on after low- power mode exit using the following steps:

1. Set up the current mode's Mode Configuration Register (ME_<current mode>_MC) so that PLL1 is off (PLL1ON = 0b0), PLL0 is on (PLL0ON = 0b1), and PLL0's reference clock source is on (XOSCON or IRCON = 0b1) and then request a mode change to the current mode via the Mode Control Register (ME_MCTL).
2. Set up ME_<target mode>_MC register so that PLL1 is off (PLL1ON = 0b0) and PLL0 is off (PLL0ON = 0b0) and then request another mode change to the target mode via the ME_MCTL register.
3. After low-power mode exit, Set up the current mode's Mode Configuration Register (ME_<current mode>_MC) so that PLL1 is on (PLL1ON = 0b1), PLL0 is on (PLL0ON = 0b1), and PLL0's reference clock source is on (XOSCON or IRCON = 0b1) and then request a mode change to the current mode via the Mode Control Register (ME_MCTL).

1.85 ERR005085: MC_ME: Access error not generated on writes to read-only ME_CCTL0 register

Description:

An access error does not occur when software attempts to write to the CORE0 Control register (ME_CCTL0) in the Mode Entry module (MC_ME) even though this register is not writable.

Workaround:

Software should not attempt to write to the ME_CCTL0 register.

1.86 ERR005087: Flash: Short functional reset causes flash error

Description:

During a short 'functional' reset, the flash configuration in the Platform Flash Memory Controller (PFLASH) is reset while the flash memory array is not reset. If such a reset occurs while the flash is being read, the first read of the flash on reset exit may be corrupted and thus cause a machine check.

Workaround:

Configure all reset events to be long by setting all programmable bits in the 'Functional' Event Short Sequence register (RGM_FESS) of the Reset Generation Module (MC_RGM) to 0b0, and by configuring all faults to generate any reaction other than a "soft reaction" in the Fault Collection and Correction Unit (FCCU).

1.87 ERR005089: MC_ME:Unexpected ICONF_CU interrupt generated on correct mode transition

Description:

The invalid mode configuration interrupt for clock usage, as indicated by the setting of the I_ICONF_CU bit of the Interrupt Status register (ME_IS) in the Mode Entry module (MC_ME), is generated during a mode transition if a clock source is configured to be turned off in the target mode when it is actually used by a peripheral that is on in the target mode. However, in some cases, this interrupt is also generated when no peripheral enabled in the target mode uses any clock source that is configured to be turned off in the target mode.

Workaround:

Set the M_ICONF_CU bit of the Interrupt Mask register (ME_IM) to 0b0. This completely disables the invalid mode configuration interrupt for clock usage generation.

1.88 ERR005100: GTM: Wrong calculation of TSAC in DPLL for actions due to the erroneous internal address pointer

Description:

(GTM-IP-113)

The Generic Timer Module (GTM) Digital PLL (DPLL) internal address pointer to RAM 2 or RAM1c can have an unintended overflow and therefore point to an unpredictable address inside the RAM region. As a result the Calculated Timestamp for Action (TSAC) may be wrong. The likelihood of getting the wrong pointer depends on the Trigger and State Number (TNU and SNU) values. The probability of this happening can be estimated as 1/(2*TNU+2) or 1/(2*SNU+2) respectively. The probability is independent of the chosen Number of Recent TRIGGER/STATE events (NUTE/NUSE) value in the DPLL_NUTC and DPLL_NUSC registers.

The following calculation, at the next input event for the same action request, does not result in a correct address pointer. The observed variation of TSAC depends on the current and last Time Base Unit Channel 0 Base (TBU_CH0_BASE) value.

In the event of the erroneous action calculation, TSAC is an unpredictable value. The sub increment generation is not effected.

Workaround:

Use Calculated Position Value for Action (PSAC) instead of TSAC.

When the error occurs varies for different configurations, in different situations, and is not predictable by SW by observing DPLL status information. The defect is caused by the value of internal registers and pointers which are not accessible via AEI interface, therefore, there is no way to predict the misbehavior.

Error detection is only possible when the action is calculated redundantly with the angle difference of >= 360/(TNU+SYN_NT+1) or >= 360/(SNU+SYN_NS+1). In these cases at least one of the two calculations is correct and the difference must be evaluated by software.

It is also possible to predict the TSAC value by software using the PSAC value and compare it with the provided one.

1.89 ERR005101: GTM: Unintended pulse correction by MPVAL1 in DPLL with first event after enabling the DPLL if PCM1 is not set

Description:

(GTM-IP-114)

In the Generic Timer Module (GTM) When the Missing Pulses to be Added or Subtracted Directly to SUB_INC1 and INC_CNT1 (MPVAL1) value is written to a non zero value, and when the first event after enabling the GTM Digital Phase Lock Loop (DPLL) is detected, the pulse correction is done even when the Pulse Correction Mode for SUB_INC1 generation is disabled (PCM1=0 in DPLL_CTRL1). Unintended pulse correction leads to wrong Time Base Unit (TBU) counter values.

Workaround:

Do not write the MPVAL1 value before the DPLL is enabled. Write the value when the first event is detected.

1.90 ERR005107: LFAST: PLL has excessive long term jitter increasing the BER

Description:

The Phase Lock Loop (PLL) of the Low Voltage Differential Signalling (LVDS) Fast Asynchronous Serial Transmission (LFAST) has an excessive long term jitter leading to a higher Bit Error Rate (BER) on LFAST transactions.

The impact on the BER increases with the size of the transmit frame.

Workaround:

The BER can be improved by adjusting the LFAST Phase Lock Loop (PLL) Optimization to the 2 times multiplier for the charge pump current (LFAST PLLCR[LPCFG]=0b11), by reducing the data rate to 160 mega-bits per second (instead of 320 MBPS, and by using shorter frames (32 bit frame instead of 96 bits). These are listed in order of impact to the overall system, but the largest gain in performance is reducing the data rate.

1.91 ERR005116: JDC: Operation of the JTAG Input Data Register Ready bit MSR[JIN_RDY] requires JTAG Clock (TCK) to continue to run after exit from the Update-DR state.

Description:

The JTAG Input Data Register Ready (JIN_RDY) bit of the JDC Module Status Register (MSR) register is cleared upon peripheral bus (PBRIDGE) read of the register. However, the bit will not be cleared unless TCK is allowed to run at least two clock periods after exit from the Update-DR state (e.g. the state when the controller decodes and selects register) for the Data Register (DR) shift that was used to perform the write to the JIN register.

Workaround:

Allow TCK to continue to run at least two clocks after exit from the Update-DR state following DR scans.

1.92 ERR005118: JDC: MSR[JIN_RDY] and MSR[JIN_INT] will not be cleared if IPS access occurs while JTAG state machine is still in Update-DR state.

Description:

Following a JTAG write to the JTAG Input Data (JIN) register, the JTAG Input IPS (JIN_IPS) register is updated with the JIN contents and the JDC Module Status Register JIN Ready MSR[JIN_RDY] and JIN Interrupt MSR[JIN_INT] bits are set, indicating new data has been written to JIN_IPS. Since the system clock frequency is typically much faster than the JTAG clock frequency, the JTAG state machine may still be in the Update-DR state when the system receives the JIN_INT interrupt request. If the interrupt request is serviced before the JTAG state machine transitions out of the Update-DR state, the system read of the JIN_IPS register will not clear the MSR[JIN_RDY] bit, and the system write to the MSR[JIN_INT] bit will not clear the MSR[JIN_INT] bit.

The JTAG state machine must first transition out of the Update-DR state for the MSR[JIN_RDY] and MSR[JIN_INT] bits to update correctly in response to system reads and writes to the JDC MSR register.

Workaround:

Delay service of JDC interrupt (including read of JIN_IPS register and write to MSR[JIN_INT] to clear the interrupt) for a number of clocks equal to four times the ratio of system clock frequency to JTAG TCK clock frequency after receiving the JDC interrupt request. This will allow the JTAG state machine to fully transition out of the Update-DR state.

1.93 ERR005133: LINFLEXD : PBRIDGEx_CLK frequency must be less than or equal to two- thirds of LIN_CLK and greater than or equal to half of LIN_CLK for proper functionality.

Description:

The bus interface unit clock of the LINFlex module (from the Peripheral Bridge, PBRIDGEx_CLK) must be less than or equal to two-thirds of the module clock to the LINFlex (LIN_CLK). In addition, the PBRIDGEx_CLK must be greater than or equal to one-half of the LIN_CLK frequency:

(1/2)*LIN_CLK =< PBRIDGEx_CLK =< (2/3)*LIN_CLK

Communication errors can occur if these requirements are not met.

Workaround:

Ensure that the bus clock to the LINFLEX is less than or equal to two-thirds of the module clock and is also greater than or equal to one-half of the module clock.

1.94 ERR005137: JDC: JDC MSR[JOUT_RDY] bit may be cleared even though data from JOUT_IPS has not been read.

Description:

In normal operation of the JTAG Data Communication module (JDC), the Module Status Register JTAG Out ready (MSR[JOUT_RDY]) bit is set following a system write to the JTAG Output Data Register (JOUT_IPS), and cleared upon exit of the Update-DR state of a JTAG read of the JOUT register contents.

If the system write to the JOUT_IPS register occurs during a JTAG poll of the JOUT register, then the MSR[JOUT_RDY] will be cleared following exit of the Update-DR state, which is before the data has actually been read via the JTAG port. The system will have no way of determining of the MSR[JOUT_RDY] bit was cleared as a result of a successful read of the JOUT_IPS register value or due to the improper clear of the bit as described above.

Workaround:

Since the JOUT_RDY value cannot be reliably read when polling JOUT, consecutive JOUT_IPS values from software must be unique. This allows the tool to determine when new data is available even if JOUT_RDY is not asserted.

Software restrictions: Due to a case where the JOUT_RDY value may be cleared prior to being read by the tool, consecutive JOUT_IPS values must be unique.

Tool restrictions: If the JOUT_IPS field of the JOUT register changes value but the JOUT_RDY bit of the JOUT register is cleared, assume the system cleared JOUT_RDY prior to it being read via JOUT. In this case, respond to the new JOUT_IPS value as if JOUT_RDY had been asserted.

1.95 ERR005299: MC_RGM: ESR0 can be configured to generate non-reset reaction

Description:

Writing a 0b1 to the read-only Disable ESR0 External Reset (D_ESR0) bit of the ‘Functional' Event Reset Disable register (RGM_FERD) of the Reset Generation Module (MC_RGM) will change the value of this bit to 0b1 and will result in the ESR0 external reset event triggering a non-reset reaction. Also, read-only bit 31 of the ‘Functional' Event Alternate Request register (RGM_FEAR) will retain its written value, and if the D_ESR0 bit is set to 0b1, it will determine whether a SAFE mode (RGM_FEAR[31] = 0b0) or an interrupt request (RGM_FEAR[31] = 0b1) will be generated on an ESR0 event.

Workaround:

Ensure that the D_ESR0 bit of the RGM_FERD register is always written with 0b0.

1.96 ERR005300: GTM: TIM TGPS mode count starts early

Description:

(GTM-IP-115)

When The Generic Timer Module (GTM) Timer Input Module (TIM) is in Gated Periodic Sampling Mode (TGPS) mode (TIM_MODE = 101) with the settings ISL=0, DSL=0 in register TIM[i]_CH[x]_CTRL, the counting of clock events in TIM[i]_CH[x]_CNT starts as soon as the channel is enabled independent of the present input signal level.

Workaround:

Setup the TIM channel in a way that it is ensured that counting only starts if input signal=0. This can be achieved by the following configuration sequence:

- Activate TGPS with DSL=0
- Set CNTS to 0 to prevent TIM[i]_CNT from counting
- Set TIM input via TIM[i]_IN_SRC to 0
- Enable the channel
- Set TIM input via TIM[i]_IN_SRC to 1
- Set CNTS to sampling frequency
- Clear notify bit
- Switch back to TIM[i]_CH input

1.97 ERR005302: GTM: DPLL may not trigger the requested action when the ARU delivers data to the DPLL

Description:

(GTM-IP-118)

The Generic Timer Module (GTM) Digital PLL (DPLL) internal processing unit may not trigger an action when the Advanced Routing Unit (ARU) delivers data to the DPLL correctly. The condition can occur over a 4 system clocks period when an internal observation time frame for new action data ends, and during that same interval, new action

data is transferred by the ARU. The action data is transferred correctly to the RAM1a memory, however the action calculation is not started by the DPLLs internal processing unit. If further requests for action calculation are placed within the same event cycle they are not calculated either.

Workaround:

Do not use the GTM DPLL. If the DPLL is used, the probability of this conditions occurrence can be calculated by dividing the 4 system clocks by the ARU round trip time.

1.98 ERR005303: GTM: TIM timeout and overflow ACB bits may not be correct

Description:

(GTM-IP-119)

When the timeout detection unit (TDU) of a Generic Timer Module (GTM) Timer Input Module (TIM) channel is enabled (TIM[i]_CH[n]_CTRL[TOCTRL] != '00') and the ARU routing for this channel is enabled (TIM[i]_CH[n]_CTRL[ARU_EN] = '1') the ARU Control Bits (ACB) presented to the Advanced Routing Unit (ARU) may not indicate the correct status when the Timeout Detection Unit (TDU) timeout event occurs 1 GTM system clock before a General Purpose Register (GPR0/1) update.

Workaround:

Configure the TIM filter and the TIM timeout unit to operate on the same Clock Management Unit Clock (CMU_CLK) which must have a frequency lower than or equal to the half of the GTM system clock.

1.99 ERR005305: SARADC: Conversion data can be lost when exiting from STOP Mode

Description:

After exiting Mode Entry (MC_ME) STOP mode, the Successive Approximation Register Analog-to- Digital Converter (SARADC) digital portion can re-start a conversion using a slower clock while the SARADC analog portion is using the correct clock. As a result, the SARADC can lose conversion data.

Workaround:

Before entering STOP mode, disable SARADC conversions. After STOP mode exits, the user can re-start conversions.

1.100 ERR005350: PAD_RING: The value at the pad output is incorrect when Output Drive Control is Open-drain / Open-source and the MSCR Invert bit is set to 0b1.

Description:

The value of the System Integration Unit Lite2 (SIUL2) I/O Pin Multiplexed Signal Configuration Register Invert bit (MSCR[INV]) is not taken into account when the MSCR Output Drive Control (ODC) configures ports to Open-drain or Open-source drive. The result is that the value seen on the device pin is not properly inverted.

Workaround:

Do not expect the value seen at the pin to be inverted when MSCR[INV] is asserted during Open- drain or Open-source operation.

1.101 ERR005435: GTM: TOM/ATOM SOMP mode, center aligned PWM with 0% duty cycle outputs 100% PWM for one PWM period

Description:

(GTM-IP-120)

If a Generic Timer Module (GTM) Advanced Router Unit Connected Timer Output Module (ATOM) or Timer Output Module (TOM) channel is configured to reset Counter 0 (CN0) on a trigger event by a preceding channel, the period of this channel is defined by the value of this trigger while the edges inside this period are defined by Compare 0 (CM0) and Compare 1 (CM1). For this configuration (usually used to create center aligned Pulse Width Modulation (PWM)) one would expect to get 0% duty cycle by setting CM0=MAX and CM1=0 where MAX is the expected value of CN0 when trigger of preceding channel resets CN0.

In this case, an initial PWM period of 100% duty cycle is applied before the correct 0% duty cycle is applied.

Also, when the (A)TOM configuration is with CM1=1 and CM1<CM0<MAX, 0% duty is expected. If 0% duty is requested this way, a small pulse of only one clock period (depending on selected Clock Management Unit clock (CMU_CLKx/FXCLKx)) is generated.

Workaround:

Do not use the configuration where CN0 is reset by another channel (center aligned PWM) and the setting of CM1=1or 0. Instead set up 0% duty cycle by setting CM1=2 and CM0>MAX.

1.102 ERR005436: GTM: DPLL misses a trigger edge from the ARU when an action calculation is requested by the DPLL at the same time

Description:

(GTM-IP-121)

When a new input event (Trigger or State) occurs in a short time frame before an action request is scheduled (Position Minus Time, PMT) by the Generic Timer Module (GTM) Digital PLL (DPLL), the input event may not be recognized by the DPLL internal data processing unit. As a result the sub_inc generation for that input event is not calculated, nor is the synchronization executed. Sub increment pulses will be lost due to the lost input event therefore the sub_inc generation is incorrect and loss of synchronization is probable. The resultant PMT action request calculation based on the last observed edge may be inaccurate.

Another effect of this missing input event is that the internal unit which checks for missing triggers and raises interrupts for them, does recognize the new trigger edge but it is not updated with the new time stamp. As a result the missing event is not indicated by the missing trigger interrupt (DPLL_MT_IRQ).

Workaround:

Do not use the GTM DPLL.

1.103 ERR005581: PMC: FlexRay power supply required before or with VDD_HV_ADV

Description:

The power supply for the FlexRay pads (VDD_HV_IO_FLEX) must be powered before or at the same time as the supply for the analog to digital converter modules (VDD_HV_ADV).

If VDD_HV_IO_FLEX is not available, extra current consumption can be seen during rising of VDD_HV_IO_FLEX, and PC[8] may show unexpected leakage.

Workaround:

Ensure that VDD_HV_IO_FLEX supply is provided before or with VDD_HV_ADV.

1.104 ERR005583: SARADC: ADC0 should always be enabled to use other SAR ADCn modules.

Description:

The Successive Approximation Analog to Digital Converter (SARADC) bias generator is controlled by the clock of the SAR ADC0. It is switched-off when SAR0 AD enters any stop mode (ADC stop mode). In order to ensure correct conversion of other SARn ADC modules, SAR0 ADC should remain on. It can be switched off if no SAR module are needed

Workaround:

Always keep the clock to SAR0 ADC enabled. Use the Mode Entry module (ME) peripheral run control registers to enable SARADC0 for all run modes. There are 8 Run Peripheral

configuration registers (ME_RUN_PC[7..0]) available for run modes at the device level. Reserve one of them with all bits set (e.g. ME_RUN_PC[7] = 0x0000_00FE), which will enable the peripheral associated with ME_RUN_PC[7] to be on for all modes (TEST, DRUN, SAFE, RUNx). Select the ME_RUN_PC7 configuration in the SARADC0 Peripheral configuration select register (ME_PCTL[127].RUN_CFG) by setting the register / bitfield to 0x7. This will ensure that SARADC0 is always on, and consequently the SARADC Bias will always be on.

1.105 ERR005584: MEMU: Unexpected non-correctable ECC errors are reported to the MEMU on system bus reads initiated by the DMA, HSM, SIPI, FEC or FlexRAY

Description:

Unexpected non-correctable Error Correction Coding (ECC) errors are reported to the Memory Error Management (MEMU) on system bus reads initiated by the Direct Memory Access (DMA) module, the Hardware Security Module (HSM), the Serial Interprocessor Interface (SIPI), the Fast Ethernet Controller (FEC) or the FlexRAY module.

Workaround:

When servicing a data storage exception due to a system bus error on a transaction initiated by the DMA, HSM, SIPI, FEC or Flexray; read the Valid (VLD) bit in the MEMU Peripheral RAM Uncorrectable Status register (PERIPH_RAM_UNCERR_STS[VLD]).

If PERIPH_RAM_UNCERR_STS[VLD] is cleared, take no further action.

If PERIPH_RAM_UNCERR_STS[VLD] is set, then:

1. Clear VLD bit in the PERIPH_RAM_UNCERR_STS register
2. Clear the Peripheral RAM Uncorrectable Error bit (PR_UCE) in the MEMU Error Flag register (ERR_FLAG)
3. Clear the Recoverable Fault Status 20 bit (RFS20) in the Fault Collection Control Unit Recoverable Fault Status register (FCCU_RF_S0)

Use Memory Built-in Self Test (MBIST) to verify the integrity of the target memory location.

1.106 ERR005595: DCI: Nexus Event Output B0/B1 filtered by the device

Description:

Pins PF[14], PM[4], and PM[5] are configured for the slowest edge rate when the Nexus Event Output B0 function is enabled.

Pins PH[11], PF[15], PK[14], and PM[6] are configured for the slowest edge rate when the Nexus Event Output B1 function is enabled.

Due to this frequency limitation, the Nexus Event Output B0/B1 may not be observable on these pads.

Workaround:

Minimize the load capacitance when using PF[14], PM[4], and PM[5] for EVTO_B0 or PH[11], PF[15], PK[14], and PM[6] for EVTO_B1.

1.107 ERR005604: Flash: Program/Erase controller fault not provided to FCCU module

Description:

The internal monitoring for the correct execution of the Program/Erase controller of the flash is not provided to the Fault Collection and Control Unit (FCCU) channel 5. As a consequence, failure during Program/Erase controller will not be reported through the FCCU.

Failure is very unlikely. However, if a failure occurs, the flash program/erase operation may not complete. A Flash reset may be needed to recover after failure.

Workaround:

Implement software monitoring of flash program/erase operation.

- Ensure program/erase operation is done in expected time. A PIT timer can be used, for example, to monitor actual execution time. As an alternative, a watch-dog can be programmed to generate a reset if operation does not complete within the expected time.
- Monitor the Flash Module Configuration Register (MCR) register Program/Erase Goog (PEG), bit 22, at end of program and erase operations to identify any failure (PEG = 0 indicates failure).

1.108 ERR005605: PMC: The status of the TESTMODE pin cannot be read.

Description:

The TEST_PIN bit (bit location 24, Most significant bit = bit 0)) in the register Module Control Register (MCR), at address PMC_BASE + 0x0350, does not reflect the status of the TESTMODE pin. This prevents the safety monitoring of the status of the TESTMODE pin while the application is running.

Workaround:

The TESTMODE pin status cannot be monitored.

1.109 ERR005607: SARADC: External trigger on SARADC module are not synchronized

Description:

The external trigger on the Successive Approximation Register Analog-to-Digital Converter (SARADC) modules are not re-synchronized with the SARADC clock. Behavior of the SARADC on such event is unpredictable and may prevent correct functionality of SARADC module after they occur.

The affected signals are the triggers associated with the pins PA[0], PE[10], PF[2], PF[5], PF[14], PG[15], PH[1], PH[12] and PH[13].

Workaround:

Do not use external pins as SARADC trigger sources.

1.110 ERR005608: RGM: A power on reset event is falsely reported following a destructive reset

Description:

When a destructive reset occurs, the device generates a falling edge on the Power On Reset (PORST) pin. The associated event detection in the Reset Generation Module (RGM) should be temporarily masked.

However the masking is incomplete: the external power on reset flag in the Destructive Event Status register (RGM_DES[F_PORST]) is also set in the case of an internally generated destructive reset.

Workaround:

Always check all bits in the RGM_DES register to determine the actual reset source.

1.111 ERR005609: FCCU: The pull-up associated to an I/O pin is not activated in Safe Mode

Description:

Each I/O pin can be configured to move automatically into a known safe state when a fault is detected. This forces the pin into high impedance mode with a weak pull-up. This is configured by clearing the MSCR[SMC] bits in the system Integration Unit (SIUL). The pin is correctly forced into high-impedance (main pin output buffer is disabled), but the associated pull-up is not activated.

Workaround:

Implement an external pull-up where pin is safety critical.

1.112 ERR005610: GTM: Input pin PH[8] and PH[9] not mapped on TIM2_1, TIM2_2, TIM2_3

Description:

Some of the device input pins cannot be associated to the Generic Timer Module (GTM) Timer Input Module 2 (GTM-TIM2) channel inputs if selected by the Source Signal Select (SSS) field the System Integration Unit Lite Multiplexed Signal Configuration Registers. All three signals are on Port H (PH). In all cases, if the TIM2 channel is selected, the channel input will be connected to ground (internally).

Pin Name	MSCR Number	SSS Value	TIM Input Function
PH(7)	530	0x0000_1010	TIM2_2
PH(8)	529	0x0000_1010	TIM2_1
PH(9)	531	0x0000_1010	TIM2_3

Workaround:

Use an alternate pin to trigger the TIM2_1, TIM2_2, and TIM2_3 inputs. Each of these TIM2 channelscan be triggered by different input pins as shown in this table.

TIM Channel	Alternate pins
TIM2_1	PC[1], PC[0], PD[8], PE[8], PE[12]
TIM2_2	PE[0], PF[6], PE[9], PH[9], PC[15]
TIM2_3	PE[1], PF[7], PH[11], PA[2],PH[8], PC[14]

1.113 ERR005611: MEMU: ECC bits not included in MEMU error log

Description:

The Error Correction Code (ECC) syndrome is not reported by the Memory Error and Management (MEMU) for single-bit errors that are corrected. Only the address information is reported by the MEMU.

Workaround:

Do not try to make use of the syndrome. If considered safety-relevant, the user must execute a software test to detect ECC error changes periodically within the Fault Tolerant Time Interval (FTTI).

1.114 ERR005622: GTM: SPE may treat an input pattern as invalid after detection of a bouncing input pattern

Description:

(GTM-IP-122)

When the Generic Timer Module (GTM) Sensor Pattern Evaluation (SPE) module detects a sequence of input patterns that leads to a signal of bouncing (Bouncing Input Signal flag SPE_BIS in the SPE[i]_IRQ_NOFITY register), this last input pattern is not treated to be valid and therefore the registers Actual Input Pattern (AIP) and Previous Input Pattern (PIP) are not updated. No new input pattern detected flag is raised (NIPD) and the pattern pointer SPE_PAT_PTR retains its previous value. Due to the missing update of the AIP and PIP registers, each succeeding input pattern is treated to be invalid even though it is a valid input pattern.

As a result, with the input patterns following the bouncing pattern, a pattern error is signaled (flag PERR is set).

With the detection of bouncing (setting of BIS) and erroneously ignoring the new input pattern (not updating AIP and PIP, not signalling NIPD) the SPE module does not update the SPE_PAT_PTR and in turn does not update the SPE output control value (SPE_OUT_CTRL).

For motor control applications, a stuck pattern pointer SPE_PAT_PTR may lead to critical stuck of commutation.

Workaround:

To re-enable the automatic update of SPE_PAT_PTR on a valid new input pattern sequence after detection of bouncing (indicated by flag/interrupt BIS in register SPE_IRQ_NOTIFY), the software has to react immediately (triggered by BIS interrupt) and re-initialize the bit fields AIP and PIP in SPE[i]_CTRL_STAT register.

1.115 ERR005623: GTM: Wrong AEI_STATUS when DPLL RAM1b is accessed during the initialization phase

Description:

(GTM-IP-123)

When the Generic Timer Module (GTM) Digital PLL (DPLL) RAM1b is accessed via the GTM Wrapper during the initialization phase (between end of the initialization of RAM1a and the end of initialization of RAM1b), the bus error signal (AEI_STATUS) from the GTM Wrapper is set to "No Error" instead of "Error".

Workaround:

Poll the RAM Region 1b Initialization In Progress (INIT_1b) bit of the Initialization Control and Status for RAMs register (DPLL_RAM_INI) before accessing RAM1b the first time after power up to ensure that the initialization is complete.

1.116 ERR005625: GTM: DPLL RAM read access while GTM is in DEBUG/HALT state leads to erroneous behavior of DPLL

Description:

(GTM-IP-125)

If the Generic Timer Module (GTM) Digital PLL (DPLL) is performing a read access to its RAM regions (two system clock cycles per read) while the GTM is entering DEBUG/HALT mode (via the GTM Debug Interface Module) and the read access is interrupted by the DEBUG/HALT state immediately after the address phase, the interrupted read access is not re-done correctly when leaving the DEBUG or HALT state. Depending on the interrupted read address, the sub increment generation and/or the position minus time calculation may be wrong.

Workaround:

Do not access DPLL RAMs while GTM is in DEBUG/HALT state.

1.117 ERR005626: GTM: Reading of ATOM SR0/SR1 by the device cores invalidates SR0/SR1 data for ARU interface

Description:

(GTM-IP-126)

When the Generic Timer Module (GTM) Advanced Routing Channel Unit (ARU) Connected Timer Output Module (ATOM) Shadow Registers (SR0 and SR1) are read by a Nexus Master through the GTM Integration Module (GTMINT) it invalidates the SR0/SR1 data for the ARU interface and therefore the ATOM channel is blocked.

If the ATOM is under the conditions:

- Signal Output Mode Compare (SOMC) mode
- The Advanced Routing Unit (ARU) is enabled
- The channel has a compare match results stored in its shadow counter registers (SR0/SR1) and waits for that result to be read via the ARU
- A Nexus read access to register SR0 or SR1 register is performed
- The ATOM channel invalidates its write request to ARU (i.e. signal of ARU interface indicating valid data) but remains in an internal state waiting for the SR0/SR1 data to be read.

Workaround:

Do not ask the Nexus masters to read the ATOM shadow registers SR0 and SR1 of any ATOM channel if the channel is configured in SOMC mode and the ARU is enabled. If this event does occur, the ATOM channel can be unblocked by a (software) reset of the channel or by a read access from the devices cores.

1.118 ERR005627: GTM: AEI bus transactions before and after a write to BRIDGE_MODE register might be serviced incorrectly

Description:

(GTM-IP-129)

An Automotive Electronics interface (AEI) bus transaction to the GTM by the CPU, before and after a write transaction to the Generic Timer Module (GTM) BRIDGE_MODE register with BRG_RST=1 (Bridge Soft Reset), may be serviced incorrectly in one of three ways

1. Read access to the GTM responds with unexpected data.
2. Write access to the GTM is not performed.
3. The GTM bus interface does not issue AEI_RESPONSE_READY which could lead to a bus timeout of the serving bus master.

Workaround:

Ensure that the write command to the BRIDGE_MODE register is issued while the transaction buffer is empty.

This condition can not be controlled/observed by the hardware, which means it must be guaranteed by software e.g. a loop, either side of the BRIDGE_MODE register write, which ensures that a certain amount of clocks will pass before the next command is issued, dependant on the aei_clk/sys_clk ratio.

1.119 ERR005630: PMC: LVD/HVD EPR registers may not show the source of a destructive reset.

Description:

When a destructive reset is caused by a Low Voltage Detect (LVD) or High Voltage Detect (HVD), it is possible the corresponding Event Pending Register (EPR) flag indication bit will not get set at the same time.

Workaround:

If it is not necessary to determine the source of the LVD / HVD event, use a functional reset instead of a destructive reset.

If a destructive reset is necessary--and the software needs to determine which of the LVD/HVDs asserted, then the following solution is possible:

Software disables all the LVD/HVD reset events via the Reset Event Enable (REE) bits. Software also turns on the Interrupt Enables for these bits. In the Interrupt Service Routine, the software would need to store the event in another location. This might be RAM (if it is not reset), in Flash, or in the Standby RAM (if not reset).

Software should then turn on the REE bits, this would cause the destructive reset to occur

1.120 ERR005632: LINFlexD: FEF not set in UART Mode Status Register (UARTSR) when a second or third invalid stop bit is received

Description:

The Framing Error Flag in the UART (Universal Asynchronous Receiver/Transmitter) Status Register (UARTSR[FEF]) does not set if the second or third stop bits are invalid. It will only set if the first stop bit received is invalid.

Workaround:

The UART should be set to only use one stop bit. This is done by programming the Stop Bits in UART reception mode field of the UART Mode Control Register (UARTCR[SBUR]) to 0x00.

1.121 ERR005633: FLASH: Sleep and disable features not implemented

Description:

The flash does not support Sleep and Disable functionality. Therefore these modes should not be enabled with the Mode Entry (ME) Module.

Workaround:

Do not set the Flash Power Down Control (FLAON) field in either the ME STOP0 Mode Configuration register or the ME HALT0 Mode Configuration register to either 0b01 or 0b10. Always set FLAON to 0b11.

1.122 ERR005639: SSCM: PAE/RAE may not block bus error generation

Description:

The following configuration of the SSCM ERROR register are not functional:

- SSCM.ERROR[PAE] = 0 and SSCM.ERROR[RAE] = 1
- SSCM.ERROR[PAE] = 1 and SSCM.ERROR[RAE] = 0

If the System Status and Configuration Module (SSCM) is configured to enable any of the following function:

- Register Bus Abort on accesses to unimplemented registers (SSCM_ERROR[RAE=0b1),
- Peripheral Bus Abort on accesses to unused off-platform peripheral slot (SSCM.ERROR[PAE] = 1)

Then exceptions will also occur for the not enabled feature.

Workaround:

Configure the SSCM.ERROR register with one of the following configuration:

- SSCM.ERROR[PAE] = 0 and SSCM.ERROR[RAE] = 0, if no exception is to be generated on peripheral erroneous accesses (default value after reset)
- SSCM.ERROR[PAE] = 1 and SSCM.ERROR[RAE] = 1, when exception must be generated on peripheral erroneous accesses

1.123 ERR005649: MPC5744K/SPC574K70: Not all spatial separation measures against CCFs implemented

Description:

Not all spatial separation measures against Common Cause Failures (CCF) of replicated channels have been implemented. Redundant I/O blocks have not been physically separated. Logic in the safety core shares a common well with logic in the checker core. Routing from the safety core is not physically separated from routing in the checker core.

Workaround:

Take these items into account in the safety analysis. This revision is not intended for series production of a safety-relevant application.

1.124 ERR005650: MPC5744K/SPC574K72: Device power consumption

Description:

The specified power consumption target for the MPC5744K/SPC574K72 has been exceeded. Total current, static current, dynamic current, and delta current targets are all above the target.

Workaround:

The power supply should allow for up to 400mA on the 5V supply as described in the DC electrical specifications of the latest Datasheet, pending further device characterization. Additional capacitance should be provided on the application board to handle current variation. Current can be reduced by running at a lower temperature to reduce static leakage, running at a lower frequency, and by writing software which avoids large instantaneous changes in silicon usage.

1.125 ERR005651: LINFlexD: DRF/RFE of the UART Status Register is always set after the first stop bit is received

Description:

When the LINFlexD is configured in UART (Universal Asynchronous Receiver/Transceiver) mode then the Data Reception Completed Flag in the UART Status Register (UARTSR[DRF/RFE]) is always set after the first stop bit is received. This occurs even if the number of stop bits is configured to 2 or 3 using the Stop Bits in UART reception mode field of the UART Control Register (UARTCR[SBUR]).

Workaround:

The UART must be configured to use buffered received mode rather than DMA receive mode. Software is then used to poll the Data Reception Completed Flag in the UART Status Register (UARTSR[DRF/RFE]), and depending on whether the number of Stop Bits in the UART reception mode field of the UART Control Register (UARTCR[SBUR]) is 2 or 3, the software waits an additional 1 or 2 bits.

1.126 ERR005652: MBIST: MBIST on-line may corrupt RAM content

Description:

Running the Memory Built In Self-Test (MBIST) in on-line mode may result in the corruption of any RAM memory embedded within the device.

The corruption occurs at the end of the on-line MBIST. MBIST may generate a partial access on the data and address bus on the last cycle of its execution . When the RAM is accessed during this cycle, the contents of the RAM may be corrupted.

Workaround:

Ensure no RAM accessing is done during the MBIST execution. In particular:

- If any processors are running an active loop, ensure execution is done directly from Flash array, and any access to the local RAM, cache and system RAM is prevented. Alternatively ensure all processors are in HALT mode during MBIST execution.
- Ensure any masters potentially accessing the RAM are disabled: processor, DMA, Ethernet module
- Disable peripherals with embedded RAM: CAN module, Flexray module, Generic Timer Module (GTM)

Software can detect that MBIST is complete by polling RUNSW bit of Self Test Control Unit (STCU) of the Run Software Register (STCU_RUNSW). Such polling must be performed in the simplest form with code executing directly from flash (without accessing any RAM).

1.127 ERR005655: FEC: Ethernet REF Clock Selector and Divider Registers Associated with Auxiliary Clock 11 Instead of 10

Description:

The source for the Ethernet transmit clock (TXCLK_REF) is selected in the System Integration Unit Lite (SIUL2) instead of via the auxiliary clock 10 selector in the Clock Generation Module (MC_CGM).

In addition, the Ethernet REF clock selector and divider are implemented on auxiliary clock 11 instead of auxiliary clock 10.

Workaround:

Select the Ethernet transmit clock as a source for AUX_10, via the Multiplexed Signal Configuration Register 42 (SIUL2_MSCR42).

Configure the Ethernet transmit clock (TXCLK_REF) via the Auxiliary Clock 10 Select Control Register (CGM_AC10_SC)

Configure the Ethernet REF clock via the Auxiliary Clock 11 Select Control Register (CGM_AC11_SC) and Auxiliary Clock 11 Divider 0 Configuration Register (CGM_AC11_DC0).

1.128 ERR005660: MC_CGM: System Clock Divider registers MC_CGM_SC_DC0/1/2 cannot be changed by I/O Processor (IOP)/Core 2

Description:

The System Clock Divider registers (MC_CGM_SC_DC0/1/2) in the Clock Generation Module (CGM) cannot be changed while the I/O Processor (IOP) is executing code. If the System Clock Divider is updated while I/O processor is executing code, the behavior of the I/O processor becomes undefined.

Workaround:

In order to change the System Clock Divider, the I/O processor should be put into reset and the change of the division factors should be done using safety core - Core 0. Once the dividers are changed, the I/O Processor can be restarted and proceed with the normal instruction flow.

1.129 ERR005662: STCU2: the Logic BIST partitions 3, 5, 6, and 7 are not implemented

Description:

The Logic Built-in Self-Test (LBIST) functionality of the partitions 3, 5, 6, and 7 are not implemented and no LBIST can be run these partitions.

Partition 3: main core and platform

Partition 5: application peripherals on AIPS0 Partition 6: GTM module

Partition 7: checker core and flash module

Workaround:

LBIST partitions 3, 5, 6, and 7 should not be enabled in the Self Test Unit (STCU2) Configuration, LBIST Control, and MBIST Control registers (STCU_CFG[PTR], STCU_LB_CTRL_X[PTR] and STCU_MB_CTRLX.[PTR]).

1.130 ERR005670: SIUL2: Missing alternate functions on some pins

Description:

Some pins do not implement all associated alternate functions. The alternate function which are not implemented, as well as their associated Multiplexed Signal Configuration Register (MSCR) and Source Signal Select bit values, the are shown in the following list.

PA[3]:
- DSPI_2.SOUT on MSCR3[SSS] = 0000_0100
- DSPI_2.SIN on MSCR886[SSS] = 0000_1001

PA[8]:
- DCI.EVTO_B0 on MSCR8[SSS] = 0000_0011
- DCI.EVTI_B01 on MSCR8[SSS] = 0000_0100

PA[9]:
- DCI.EVTO_B1 on MSCR8[SSS] = 0000_0011
- DCI.EVTI_B13 on MSCR8[SSS] = 0000_0100

PA[13]:
- DSPI_4.CS1on MSCR13[SSS] = 0000_0011

PB[8]:
- DSPI_2.SCK on MSCR24[SSS] = 0000_0100
- DSPI_2.SCK on MSCR887[SSS] = 0000_0100

PB[9]:
- DSPI_2.CS0 on MSCR25[SSS] = 0000_0110
- DSPI_2.CS0 on MSCR888[SSS] = 0000_0100

PB[10]:
- DSPI_2.CS1 on MSCR26[SSS] = 0000_0110

PB[11]:
- DSPI_2.CS2 on MSCR27[SSS] = 0000_0110

PC[9]:
- PSI5_0.SDOUT on MSCR41[SSS] = 0000_0101

PC[10]:
- DSPI_1.CS3 on MSCR42[SSS] = 0000_0111

PD[7]:
- ETHERNET.MDIO on MSCR55[SSS] = 0000_0001
- ETHERNET.MDIO on MSCR924[SSS] = 0000_0101

PD[8]:
- DSPI_2.SOUT on MSCR56[SSS] = 0000_0100
- DSPI_2.SIN on MSCR886[SSS] = 0000_0110

PD[10]:
- DSPI_2.CS7 on MSCR58[SSS] = 0000_0110

PD[11]:
- SAR_B.DECODE_EXTCH_0 on MSCR60[SSS] = 0000_0100

PD[12]:
- SAR_B.DECODE_EXTCH_1 on MSCR61[SSS] = 0000_0100

PD[14]:
- LINFlexD_2.RXD on MSCR850[SSS] = 0000_0111

PE[3]:
- SAR_B.DECODE_EXTCH_2 on MSCR67[SSS] = 0000_0100

PE[10]:
- DSPI_5.SIN on MSCR895[SSS] = 0000_0011

PF[13]:
- ETHERNET.MDC on MSCR93[SSS] = 0000_0100

PF[14]:
- LINFlexD_15.TXD on MSCR94[SSS] = 0000_0011

PF[15]:
- LINFlexD_15.RXD on MSCR862[SSS] = 0000_0111

PG[15]:
- DSPI_4.CS0 on MSCR111[SSS] = 0000_0011

PH[7]:
- GTM.TIM2_2 on MSCR530[SSS] = 0000_1111

PH[8]:
- DSPI_4.CS1 on MSCR120[SSS] = 0000_0010
- GTM.TIM2_1 on MSCR529[SSS] = 0000_1111

PH[9]:
- GTM.TIM2_3 on MSCR531[SSS] = 0000_1111

PJ[0]:
- LINFlexD_15.RXD on MSCR863[SSS] = 0000_1001

PJ[1]:
- LINFlexD_15.TXD on MSCR144[SSS] = 0000_0011

Workaround:

Map the required functionality on alternative pins.

1.131 ERR005671: XBAR: Slow Crossbar is implemented as 64-bit instead of 32-bit crossbar

Description:

The slow crossbar (XBAR) is implemented as a 64-bit bus instead of a 32-bit bus. 64 bit transactions across the slow crossbar will have different timing than a 32-bit implementation. This allows 64-bit transactions to be completed in 1 clock cycle. On a 32-bit bus, 64-bit transactions will be split into two 32-bit accesses, reducing throughput slightly.

Workaround:

Either: use only 32 bit or smaller transactions on the slow crossbar; or use 64-bit transactions only where their timing is not crucial.

1.132 ERR005689: GTM: DPLL RAM Region 1 b+c initialization beyond implemented address range

Description:

(GTM-IP-133)

The Generic Timer Module (GTM) Digital PLL (DPLL) RAM Regions 1b and 1c are 384 words with 24 bit width. During initialization of RAM, only 384 words need to be accessed from offset addresses at 0x000 to 0x17F. The initialization actually initializes a RAM area of 512 words. This means that when addresses over 0x17F are access and a memory protection units is enabled to check for illegal RAM accesses the DPLL RAM initialization would cause an illegal access error.

The device is configured to initialize the RAMs out of reset.

Workaround:

Ensure by either hardware or software that before initialization of DPLL RAM regions is started that any illegal RAM address accesses reporting is disabled.

1.133 ERR005717: FCCU: Breakpoint can cause FCCU to exit CONFIG state

Description:

A watchdog timer monitors the duration for which the Fault Collection and Control Unit (FCCU) stays in CONFIG state. However, due to debug events such as a breakpoint, the FCCU may stay in the CONFIG state longer than the programmed duration. This will cause the watchdog timeout to occur which forces the FCCU to exit from the CONFIG to the NORMAL state. This also resets the CONFIG registers to the default values. CONFIG registers are those which are writeable only in CONFIG mode. A safe duration to stay in the CONFIG state, is the lower of CONFIG timeout (which can be programmed to a maximum value of 8.14ms) and FCCU Output Supervision Unit (FOSU) Timeout period.

Workaround:

Ensure that the FCCU is kept in the CONFIG state for less than the FOSU timeout period during any debug event. Otherwise do not enable breakpoints while the MCU is in the CONFIG state.

1.134 ERR005718: SIPI: Channel coding must be the same for the SIPI partner

Description:

The Serial Inter-Processor Interface (SIPI) channel encoding select bit (CHNSB, bit 27, least significant bit = 31) is not implemented in the SIPI Module Configuration Register (SIPIMCR). Therefore only one set of channel numbers that are selected by CHNSB=0 are available to communicate to a remote SIPI client. The channel encoding are listed below.

Channel	CHNSB=0	CHNSB=1
Channel 0	0b100	0b000
Channel 1	0b101	0b001
Channel 2	0b1110	0b010
Channel 3	0b111	0b011

Workaround:

Always use SIPI channels that are selected by setting CHNSB to a 0 (Channel 0 = 0b100, Channel 1 = 0b101, Channel 2 = 0b110, and Channel 3 = 0b111) in the SIPIMCR.

1.135 ERR005719: SIUL: MSCR936-967 are not protected through the register protection mechanism

Description:

The Multiplexed Signal Configuration Registers (MSCR) 936-967 are not protected through the register protection mechanism. These MSCR are associated to the micro second bus 0 module. Therefore, the configuration for these register cannot be locked by software.

Workaround:

Expect that MSCR[936] - MSCR[967] are not protected by the register locking mechanism and application software should insure that the settings are not changed and, for safety critical applications, that the values are rewritten to their required values periodically.

1.136 ERR005748: SARADCB: Programming TCCAPR0-6 will result in an electrical short between the test channel and another internal analog channel

Description:

In the Successive Approximation Register Analog-to-Digital Converter instance B (SARADC_B) it is possible to connect any test channel to any other internal analog channel. The connection mapping is controlled by the Test Channel Connection with Analog Pin Registers (TCCAPR0-7) programming. For test channels 124-127 controlled by TCCAPR7, the connections are intentional and the short is through a serial 20Kohm resistor.

However, for test channels 96-123 controlled by TCCAPR0-6, the connections are not intended and programming of these registers may create an electrical short circuit without serial resistance and may damage the channels.

Workaround:

The registers TCCAPR0-6 must not be programmed and must be kept at their default values always.

1.137 ERR005749: SDADC: New conversion data is discarded if the overflow (DFORF) status bit is set

Description:

The Sigma-Delta Analog-to-Digital Converter (SDADC) stops filling the Converted Data Register (CDR) and FIFO (if enabled) when the FIFO overrun bit in the Status Flag Register (SFR[DFORF]) becomes set.

The SFR[DFORF] bit becomes set when a FIFO or CDR overflow condition occurs, and once this happens the SDADC stops filling the CDR and FIFO, causing new converted data results to be discarded until the software clears the overflow bit. After clearing the overflow, normal operation resumes.

Workaround:

If the Global DMA/Interrupt gating feature is not being used then the problem can be avoided by either of two methods.

1. If DMA is being used then continuously transferring the the CDR or FIFO data via DMA will prevent overflows.
2. If DMA is not used then use an interrupt service routine (ISR) to clear the overflow bit (SFR[DFORF]) and empty the FIFO/CDR

If the Global DMA/Interrupt gating feature is being used then use an interrupt service routine at the start of the DMA gating window to clear the DFORF bit and empty the FIFO and CDR. This will resume normal filling of the CDR. or FIFO which should then be serviced using DMA or ISR for the duration of the gating window.

1.138 ERR005824: PMC: Test Enable bits are not implemented for User Test, ADC Channel Select, and test mode

Description:

Power Management Controller (PMC) Module Control Register (MCR) enable bits are not supported for User Test (UTEST), Analog-to-Digital Converter Channel Select (ADC CH SEL), and Test Mode (TESTMODE). Without dedicated enable bits, when the Power Management Controller (PMC) ADC mode digital control bus changes from one value to another, it may transition to an unintended value for a short time due to uneven delays on different bits. This may cause the PMC analog block to start an unintended test without enough time to complete the test.

Workaround:

Do not access MCR ADC Channel Select Enable (ADC_CHSEL_EN), User Self Test Enable (USER_SELF_TEST_EN), and TEST_CTRL enable (test only) bits.

1.139 ERR005845: PMC: Temperature Sensor Adjust Registers do not handle negative numbers

Description:

The Temperature Sensor User Adjust Registers do not correctly process negative numbers entered in the Temperature Sensor Configuration Register (PMC.CTL_TD) fields TRIM_ADJ_OVER and TRIM_ADJ_UNDER for respectively over and under temperature trim.

Negative numbers (indicated by the most significant bit in the Adjust Register bus) are added to the Temperature Sensor TRIM values, not subtracted.

Workaround:

Negative numbers cannot be used to adjust the temperature sensor trim values.

1.140 ERR005856: SMPU: FlexRay writes not possible when SMPU is enabled

Description:

If the System Memory Protection Unit (SMPU) is enabled by setting Control/Error Status Register Global Valid bit (SMPUx_CESR0[GVLD] == 1), writes from FlexRay will be terminated with an Error.

Workaround:

If writes from FlexRay are expected, set SMPUx_CESR0[GVLD] to 0.

1.141 ERR005857: PMC: The control and status registers (EPR, REE, RES, and FEE) do not exist for HVD145_F.

Description:

The control and status registers (Event Pending Register (EPR), Reset Event Enable (REE), Reset Event Selet (RES), and FCCU Event Enable (FEE)) do not exist for High Voltage Detect for Flash Module (HVD145_F).

Workaround:

Do not access EPR, REE, RES, and FEE control and status regsiters for HVD145_F. The HVD145_C can be used to detect any over voltage events on the Flash module.

1.142 ERR005859: CCCU: Clock Calibration on CAN Unit not implemented

Description:

The Clock Calibration on CAN Unit (CCCU) is not available on this device.

As a consequence, the CAN protocol clock is derived from either the PLL0 or the external oscillator.

Workaround:

Do not access to CCCU registers (address range 0xFFF04000 - 0xFFF07FFF).

Use the Auxiliary Clock Selector 8 Source Control field (CGM_AC8_SC[SELCTL]) to select between PLL0 and external oscillator as reference source.

Use the Auxiliary Clock 8 Divider 0 Configuration Register (CGM_AC8_DC0) to configure the prescaling factor.

1.143 ERR005860: DSPI: Timing does not match specification

Description:

The Deserial Serial Peripheral Interface (DSPI) AC Specifications are not met by the device.

Workaround:

Do not use DSPI communication line at maximum baud rate.

1.144 ERR005883: PSI5: Incorrect detection of PSI5 message length for message spreading across a configured and an unconfigured slot

Description:

In the Peripheral Sensor Interface (PSI5), the detection of the length of the associated PSI5 message can be incorrect. The "expected number of bits" PSI5_SnFCR[5:1] value is taken from the configuration register PSI5_SnFCR (Slot n Frame Configuration Register) belonging to slot[n+1] rather than the configuration register PSI5_SnFCR belonging to slot[n-1].

This incorrect detection happens when the following conditions are met:

1. Three consecutive Time Division Multiplexing (TDM) slots (slot[n-1], slotn, slot[n+1] where n=2 to 5) are programmed such that:
   - slot[n-1] and slot[n+1] are configured
   - slot[n] is unconfigured
2. The subsequently received PSI5 message contains no frame in slot[n-1] which remains empty
3. The received PSI5 frame for slot [n+1] is such that start bit S0 is detected in slot[n] and start bit S1 is detected in slot[n+1] (early frame).

Workaround:

Software can detect this fault scenario as:

- An electrical error in the Tbit will be set (not matching expected number of bits)
- The frame will be stored in slot[n-1] with slot[n] as frame number

The software can then set the expected number of bits corresponding to PSI5_SnFCR of slot[n+1] and not of slot[n-1].

1.145 ERR005884: PSI5: In synchronous mode if the first frame arrives early then subsequent frames may be disrupted

Description:

If the PSI5 (Pehiperal Sensor Interface) is configured in synchronous mode and the first message arrives so early that it overlaps with the Manchester decoder disable period then the first message and subsequent messages may be received incorrectly.

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.

1.146 ERR005887: 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.

1.147 ERR005906: GTM: TOM and ATOM inter module triggers do not work when there is a register in the trigger path between one submodule instance and the next.

Description:

(GTM-IP-134)

The trigger signal between the Generic Timer Module (GTM) Timer Output Module (TOM) or ARU Connected TOM (ATOM) submodules (e.g. signal TOM_TRIG_[i]) can be stored in a register at the module output to break long combinational paths. When this store register in place, it results in a delay of one system clock period of the trigger signal.

Between module instances TOM[i] / ATOM[i] and TOM[i+1] / ATOM[i+1], when there is a store register in the the trigger path, this trigger is only recognized by the channel of TOM[i+1] / ATOM[i+1] if the channel is running from a source identical to the system clock (i.e. the selected Clock Management Unit Fixed Frequency Clock (CMU_FXCLKx) or Clock Management Unit Clock (CMU_CLKx) period is the system clock (SYS_CLK) ÷ 1). If a lower frequency is chosen to clock the TOM[i+1] / ATOM[i+1] channel, the trigger is not recognized.

Workaround:

TOM Workaround 1:

To use the inter module triggers, the channel of TOM[i+1] that is to be triggered from TOM[i] must use a clock period identical to the SYS_CLK period.

TOM Workaround 2:

On TOM[i+1] configure a redundant channel to trigger another channel of TOM[i+1] as it was configured on TOM[i] to trigger the other channel. Then start TOM[i] and TOM[i+1] synchronously by using the Time Base Unit (TBU) comparator of the TOM Global Control

(TGCx) unit (TOM[i]_TGC[y]_ACT_TB register).

ATOM Workaround 1:

When there is a register in the trigger path between ATOM[i] and ATOM[i+1], the channel of ATOM[i+1] that should be triggered has to use a clock of period identical to SYS_CLK period. The configuration of the ATOM outputs differs between devices, in some cases each ATOM has the save trigger register, in some devices every second ATOM module has the register. Check the GTM specification for the configuration applicable to the device in use.

ATOM Workaround 2:

On ATOM[i+1] configure a redundant channel to trigger another channel of ATOM[i+1] as it was configured on ATOM[i] to trigger the other channel. Then start ATOM[i] and ATOM[i+1] synchronously by using the Time Base Unit (TBU) comparator of the ATOM Global Control (AGC) unit (ATOM[i]_AGC_ATC_TB register).

1.148 ERR005907: GTM: TIM ACB word is incorrect in the case of timeout detection

Description:

(GTM-IP-135)

When the timeout detection unit (TDU) of a Generic Timer Module (GTM) Timer Input Module (TIM) channel is enabled (TIM[i]_CH[n]_CTRL[TOCTRL] != '00') and the ARU routing for this channel is enabled (TIM[i]_CH[n]_CTRL[ARU_EN] = '1') the ARU Control Bits (ACB) presented to the Advanced Routing Unit (ARU) may indicate a timeout with an overflow of the General Purpose Registers (GPRz=0/1), ACB2=1 and ACB1=1 respectively, although no measurement cycle has finished (e.g. no signal change at the input port).

This situation occurs when two or more successive timeout events occur without any ARU read access from the TIM data's destination. The ARU connected data destination of the TIM channel will detect a valid measurement cycle after a timeout event although only a timeout event occurred.

Workaround:

To reliably detect that a measurement cycle finished after a timeout event, the GTM should additionally route the edge counter via the ARU. Whenever the ARU destination detects a transfer with ACB2=1 and ACB1=1 but the edge counter value did not change since the last ARU transfer, the received ARU word should be treated as normal timeout event (ACB2=1 and ACB1=0).

This workaround can be applied directly in the MCS if the MCS is used as data destination. If FIFO is used as data destination, the workaround has to be implemented by SW.

1.149 ERR005912: PMC: Monitoring of supply is implementing reduced hysteresis during power- up sequence

Description:

In order to exit power-up sequence, both power supplies for the IOs (VDD_HV_IO_MAIN) and the analog to digital converter modules (VDD_HV_ADV) are monitored. The VDD_HV_ADV supply is used in order to implement an hysteresis of minimum 100mV on the high voltage domain when releasing the reset, thus avoiding multiple reset sequence.

Since VDD_HV_ADC and VDD_HV_IO may be separated on the application, it is possible that VDD_HV_ADC is rising faster than VDD_HV_IO. This would potentially reduce hysteresis to be below 50mV.

Workaround:

Ensure that VDD_HV_ADV is ramping such that it stays at same level or below VDD_HV_IO_MAIN during the power-up sequence.

1.150 ERR005922: FCCU: Non Maskable Interrupt may be asserted for one safe clock cycle when FCCU change state from ALARM to FAULT

Description:

Depending on the setting of the Fault Collection and Control Unit (FCCU) Non Maskable Interrupt (NMI) Enable registers (FCCU_NMI_ENx), NMI may assert for one safe clock when the FCCU changes from the ALARM to the Fault state. If all of the bits in the FCCU_NMI_ENx registers are programmed to zero, then the NMI should not be generated when the FCCU moves to the FAULT state, however, it may assert for at least one safe clock cycle.

Workaround:

Safety Core NMI Interrupt Service Routine (ISR) should resolve the NMI source. In case there is no identified source, it should finally verify that FCCU moved from ALARM to FAULT state and NMI is disabled for all fault sources. Such case should be resolved as spurious NMI event from FCCU and ignored without any further action.

1.151 ERR005946: SARADC: AWOR_CH[x] bits not usable to isolate range WDG crossover per channel base

Description:

In the Successive Approximation Register Analog-to-Digital Converter (SARADC), the Analog Watchdog Out of Range status bit (AWOR_CH[x]) in the Internal, Test, and External channel Analog Watchdog Out of Range Registers (ICAWORRx, TCAWORRx, ECAWORRx) may not be set in the case of a crossover event (violation of minimum or maximum conversion range).

The AWOR_CH[x] bits are active for one SARADC clock cycle but are not latched into the I/T/ECAWORRx registers unless a write access is done to the SARADC interface.

Workaround:

The sofware should not rely on the AWOR_CH[x] bits to identify which analog channel caused a crossover.

Instead, the software could identify which threshold generated an event, using the Watchdog Threshold Interrupt Status Register (WTISR). The software should then read the Internal, Test, External Channel Data Registers (ICDRx, TCDRx, ECDRx) of the analog channels using the crossed threshold to identify the channel which triggered the interrupt.

1.152 ERR005947: SARADC: ADC may miss a GTM trigger pulse if width of pulse is less than 1 AD Clk cycle

Description:

The Successive Approximation Register Analog to Digital Converter (SARADC) may miss a trigger (and no conversions will be started) from the Generic Timer Module (GTM) if the pulse width from the GTM Timer Output Module (TOM) or Advanced Routing Unit (ARU) Connected TOM (ATOM) is less than one ADC clock. The GTM Counter Compare Unit registers (CM0/CM1) set the pulse width.

Workaround:

The GTM registers Counter Compare Unit registers (CM0 and CM1) should be appropriately programmed such that pulse width of trigger pulses is always greater than one (1) ADC clock cycle.

1.153 ERR005962: MC_ME: Incorrect setting of ME_IMTS[S_MRIG] bit on illegal mode requests

Description:

The S_MRIG bit of the Invalid Mode Transition Status register (ME_IMTS) in the Mode Entry module (MC_ME) is supposed to be set only when a mode change request is ignored while a SAFE mode transition is in progress. However, incorrectly, it is also set when a mode change request is made to a mode other than DRUN while the device is in SAFE mode and the transition to SAFE mode has completed. The correct behavior in this case would be that the S_MRI bit gets set. The mode change request itself is correctly ignored, and the device stays in SAFE mode.

Workaround:

Software should treat the setting of the ME_IMTS[S_MRIG] bit after a mode change request when already in SAFE mode the same as the ME_IMTS[S_MRI] bit being set. Whether the transition to SAFE mode has fully completed can be determined by comparing the MC_ME's Global Status register (ME_GS) content with that of its SAFE Mode Configuration register (ME_SAFE_MC).

1.154 ERR005976: DMA: Unexpected Non-correctable ECC error is reported to MEMU on DMA transfers of less than 64-bit destination size

Description:

Direct Memory Access (DMA) transfers, where the transfer destination is mapped to Peripheral Bridge (PBRIDGE) space or e200z4 core local memory, and the destination transfer size is less than 64-bits, may result in the reporting of a non-correctable Error Correction Code (ECC) event to the Memory Error Management Unit (MEMU).

The errata has three elements to it:

1. The DMA controller will hold stale data in its write buffer between transfers
2. The stale data will be propagated to the slaves during write transfers
3. The e200z4 Core local memories and PBRIDGE memories are sensitive to the stale data

System RAM is not sensitive to the stale data.

The condition occurs on DMA writes to Core local memories and/or PBRIDGE memories when a smaller transfer size (e.g. 32-bit, 16-bit, 8-bit) follows a larger size transfer.

Workaround:

For e200z4 CPU core local memories and PBRIDGE memories, use 64-bit DMA transfers only. The errata will be avoided since the user is not presenting stale data to the memories.

For System RAM, mixed size transfers (8-bit, 16-bit, 32-bit, 64-bit) can be used since System RAM is not sensitive to the stale data.

Note: The TCD reload function does a fixed, max-size DMA transfer. So if the TCD reload function is invoked, then all channels must be configured to set source/destination size to max_size. i.e. transfer sizes of smaller granularity won't work after a TCD reload event.

1.155 ERR005978: MC_CGM: System clock dividers generate unaligned divided clocks when programmed back to back and when the first divider is configured for divide by 3

Description:

System clock dividers in the Clock Generation Module (MC_CGM) generate unaligned divided clocks when all the conditions below are met:

1. Dividers are configured back to back (without any delay in between)
2. The first divider to be programmed is in divide by 3 configuration

In such a scenario the divided system clocks are generated with unaligned active edges, which can lead to unpredictable behavior such as incorrect register and memory reads and writes.

Workaround:

Software should read the System Clock Divider Configuration registers (CGM_SC_DCn) after writing them. This ensures that back to back writes do not happen, and the condition required for the failure to occur is never met.

1.156 ERR005984: INTC: Interrupt controller running on faster clock

Description:

The interrupt controller (INTC) is connected to fast crossbar clock while it should be connected to half speed of the slow crossbar clock. This results in a faster interrupt response time as expected.

Assuming a fast crossbar speed of 160 MHz, the expected interrupt response time is 143.75 ns while the actual interrupt response time is 87.5 ns.

Workaround:

For interrupt performance measurements, take into account that the actual interrupt response time is shorter than expected.

1.157 ERR005994: FLASH: Prefetch mini-cache enable fields are configurable and default to disabled

Description:

The Flash prefetch mini-cache enable fields in the Platform Flash Configuration Registers (PFCR1[P0_BFEN] and PFCR2[P1_BFEN]) are configurable and default to disabled. Leaving the prefetch mini-cache enable field unchanged from their default states may result in lower lifetime performance when accessing the flash data and prevention of flash read disturb cannot be guaranteed. In the future, these bits will default to enabled and will not be changeable.

Workaround:

Set PFCR1[BFEN] and PFCR2[BFEN] during initialization, then do not modify this setting.

1.158 ERR005998: FCCU : Fault occurring during CONFIG mode does not generate reset

Description:

When the Fault Collection and Control Unit (FCCU) is in configuration mode (CONFIG), no reset will be generated on the occurrence of a fault source configured to generate a long or short functional reset as a reaction.

The FCCU acknowledges the fault when leaving CONFIG mode and entering the normal mode, but switches back to the Fault state and a reset is not generated.

Workaround:

Do not expect resets generated for fault captured while in the CONFIG state.

1.159 ERR005999: TDM: DBA and LFPAR fields do not reflect addresses in the system memory map

Description:

In the Tamper Detect Module (TDM), the Last Flash Programmed Address Register (LFPAR) and Diary Base Address (DBA) reported values are relative to the internal flash memory map, rather than the overall system memory map.

The DBA holds the base address of the diary. The LFPAR holds information of the address of the last successful programming operation provided from the flash memory to the TDM.

Workaround:

Use the following table to identify the flash block which correlates to the reported internal flash address in the LFPAR and DBA registers. The following table reports the correspondence of system address (on the left side) versus "internal flash" address (on the right side) for each flash sector, to be used in the LFPAR and DBA registers.

System address - Internal flash address (partitions)

---

sector: UTEST NVM Block - B0UT

BLK_16K_BASE 0x00400000- 0x00800000 (partition 0)

sector: BAF - B0BF

BLK_16K_BASE 0x00404000 - 0x00804000 (partition 0)

sectors: EEPROM - high blocks

BLK0_16K_BASE 0x00800000 - 0x00280000 (partition 2)

BLK1_16K_BASE 0x00804000 - 0x00284000 (partition 2)

BLK2_16K_BASE 0x00808000 - 0x00288000 (partition 2)

BLK3_16K_BASE 0x0080C000 - 0x0028C000 (partition 2)

sectors: Low&Mid blocks - low blocks

BLK1_16K_BASE 0x00FC0000 - 0x00000000 (partition 0)

BLK2_16K_BASE 0x00FC4000 - 0x00004000 (partition 0)

BLK3_16K_BASE 0x00FC8000 - 0x00008000 (partition 0)

BLK4_16K_BASE 0x00FCC000 - 0x0000C000 (partition 0)

BLK0_32K_BASE 0x00FD0000 - 0x00010000 (partition 0)

BLK1_32K_BASE 0x00FD8000 - 0x00018000 (partition 0)

BLK0_64K_BASE 0x00FE0000 - 0x00020000 (partition 0)

BLK1_64K_BASE 0x00FF0000 - 0x00030000 (partition 0)

sectors: Large blocks - 256K blocks

BLK0_256K_BASE 0x01000000 - 0x00040000 (partition 0)

BLK1_256K_BASE 0x01040000 - 0x00080000 (partition 0)

BLK2_256K_BASE 0x01080000 - 0x000C0000 (partition 0)

BLK3_256K_BASE 0x010C0000 - 0x00100000 (partition 0)

BLK4_256K_BASE 0x01100000 - 0x00140000 (partition 1)

BLK5_256K_BASE 0x01140000 - 0x00180000 (partition 1)

BLK6_256K_BASE 0x01180000 - 0x001C0000 (partition 1)

BLK7_256K_BASE 0x011C0000 - 0x00200000 (partition 1)

BLK8_256K_BASE 0x01200000 - 0x00240000 (partition 1)

1.160 ERR006007: FCCU: Once unlocked, Transient Lock on FCCU configuration cannot be reactivated

Description:

The Fault Collection and Control Unit (FCCU) configuration is lockable with a transient lock and a permanent lock. Once the permanent lock is activated by writing 0xFF to the FCCU Permanent Lock Register (FCCU_PERMNT_LOCK[TRANKEY]=0xFF]), it cannot be deactivated until a destructive reset occurs. If the Transient lock is used to unlock the configuration (by writing the transient key value of 0xBC to the FCCU Transient Lock Register [TRANSKEY]), the lock can not be reactivated by writing any value to the FCCU_TRANS_LOCK register. Writing of any key other than 0xBC was supposed to lock the registers but that is not the case.

By default, the transient lock is active (locked) after an FCCU reset (destructive reset).

Workaround:

The FCCU configuration is either completely locked permanently or not locked at all.

1.161 ERR006018: SMPU: errors are not forwarded to the FCCU

Description:

The System Memory Protection Unit (SMPU) concurrently monitors and evaluates system bus transactions using pre-programmed region descriptors that define memory spaces and their access rights. Memory references that have sufficient access control rights are allowed to complete, while references that are not mapped to any region descriptor or have insufficient rights are terminated with an access error response. These access errors are reported on Fault Collection and Control Unit (FCCU) Channel #44. The feature to report the error to FCCU on Channel #44 is not implemented on this device.

Workaround:

Do not expect to see SMPU errors forwarded to the FCCU.

1.162 ERR006019: e200a420: Core0 caches are twice the target size

Description:

The e200z420 used as the main core (also known as core 0) in this device revision implements twice the amount of instruction and data cache memories. The targeted sizes are respectively 4 and 2 Kbytes for instruction and data.

Workaround:

To ensure comparable performance to future versions of the device, limit the cache size used by software by locking invalid content into the additional memory.

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

Description:

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

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

Workaround:

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

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

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

Step 2: Poll the Status Register's Transmit and Receive Status bit (DSPI_SR[TXRXS]) to ensure the DSPI has entered the HALT state and completed any in-progress transmission.

Alternatively, if continuous polling is undesirable in the application, wait for a fixed time interval such as 35 baud clocks to ensure completion of any in-progress transmission and then check once for DSPI_SR[TXRXS].

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

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

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

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

Step 1: When a SPI frame is required to be sent, halt the DSPI as in Step 1 of Workaround 1 above. Step 2: Enable the TCF interrupt by setting the DSPI DMA/Interrupt Request Select and Enable Register's Transmission Complete Request Enable bit (DSPI_RSER[TCF_RE])

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

1.164 ERR006041: GTM: TOM and ATOM in center aligned PWM configuration, with CM0=0 or CM0=1 on a triggered channel does not output correct waveform

Description:

(GTM-IP-136)

When a Generic Timer Module (GTM) Timer Output Module (TOM) or Advanced Routing Unit Connected TOM (ATOM) channels counter, CN0, is reset by the trigger of a preceding channel (RST_CCU0=1 in the channel control register TOM[i]_CH[x]_CTRL / ATOM[i]_CH[x]_CTRL) and a 100% or 0% duty cycle Pulse Width Modulation (PWM) is requested by configuring CM0=0 (compare register 0) (0%) or CM1(compare register 1)>=CM0 (100% when CM0=1) no edge is generated.

In the configuration where CM0=0, no edge is generated because counter CN0 is not running after the reset event.

In the case when CM0=1 and CM1>=CM0, no edge is generated even though counter CN0 is running.

Workaround:

Instead of using the configuration CM0=0 and CM1>0, use the configuration CM1=0 and CM0>CM1 and invert value of Signal Level (SL) bit to generate 0% / 100% PWM signals.

1.165 ERR006042: SARADC: extra clock cycle when chaining conversion

Description:

The Successive Approximation Register Analog to Digital Conversion (SARADC) requires one (1) additional Tck cycle to complete the conversion when chaining to the next channel. The actual formula becomes:

Tconv = Tprechg+Tsample+Teval + 1*Tck

= PRECHG*Tck+INPSAMP*Tck+25*Tck + 1*Tck

Within the reference manual, conversion time is specified as below:

Tconv = Tprechg+Tsample+Teval

= PRECHG*Tck+INPSAMP*Tck+25*Tck with following naming convention:

Tck: SAR ADC clock cycle

Tprechg: precharge period, as defined in CTRn[PRECHG]

Tsample: sampling period, as defined in CTRn[INPSAMP]

Teval: fixed period of 25 SARADC clock cycles (Tck)

Workaround:

Expect one additional clock cycle to be required in case of chained conversion.

1.166 ERR006056: FCCU: Not all Configuration Registers are reset to their default values after CONFIG timeout

Description:

In the Fault Collection and Control Unit (FCCU), the following configuration registers are not reset to the default value in case of a timeout event while in the configuration state (CONFIG). Instead of being reset, they retain their current programmed value:

1. FCCU Delta Time Constant Register (FCCU_DELTA_T)
2. FCCU Non-Maskable Interrupt Enable Register x (FCCU_NMI_ENx), x=0,1
3. FCCU Interrupt Alarm Enable Register y (FCCU_IRQ_ALARM_ENy), y=0,1
4. FCCU Error Output Signaling Enable Register z (FCCU_EOUT_SIG_ENz), z=0,1

Workaround:

After a CONFIG timeout, software should clear all pending faults and reprogram all configuration registers.

1.167 ERR006072: GTM: Wrong PSTC/PSSC value after initialization and restart of DPLL

Description:

(GTM-IP-138)

Before the First Trigger is Detected (FTD = 0 in DPLL_STATUS register) after the Generic Timer Module (GTM) Digital Phase Lock Loop (DPLL) has been initialized, the value of the Actual Calculated Position Stamp in the TRIGGER Register (PSTC) should be set equal to the Measured Position Stamp of the Last TRIGGER Input Register (PSTM). The PSTC value is not initialized with PSTM value on 1st TRIGGER edge. Once the the First Trigger is Detected (FTD = 1), the PSTC is not updated with PSTM value.

After each DPLL stop and restart sequence, the PSTC continues counting without any synchronization to the current position.

This also applies to the Actual Calculated Position Stamp of STATE (PSSC) and Measured Position Stamp at Last STATE Input Register (PSSM).

Workaround:

Write the PSTC and PSSC registers with the contents of the PSTM and PSSM registers via CPU during the DPLL initialization.

1.168 ERR006073: GTM: MCS channel might not be disabled by the MCU core

Description:

(GTM-IP-137)

When a Generic Timer Module (GTM) Multi Channel Sequencer (MCS) channel is requested to be disabled by the MCU's core through the GTM wrapper by clearing the channel enable bit (MCS[i]_CH[n]_CTRL.B.EN = 0x0), it is possible that the disable request is not executed by the MCS channel.

Workaround:

The MCU core should repeat the clearing of the Enable Channel bit (EN) of the MCS Channel Control register (MCS[i]_CH[x]_CTRL) in a loop until EN is cleared.

Repeat clearing the EN bit of the register MCS[i]_CH[x]_CTRL in a loop until it is cleared. For example:

```
while (MCS[i]_CH[n]_CTRL.B.EN == 1)
{
        MCS[i]_CH[n]_CTRL.B.EN = 0x0;
}
```

Note: This loop could take the whole duration of an Advanced Routing Unit (ARU) round trip time, if an ARU transfer is currently running, or the duration of a Wait Until Register Match (WURM) suspension time, if a WURM is currently suspending the channel.

1.169 ERR006077: MC_CGM: Value of CGM_SC_SS[SWTRG] may be incorrect after power-on reset

Description:

After power-on reset, the Switch Trigger cause (SWTRG) field of the System Clock Select Status register (CGM_SC_SS) in the Clock Generation Module (MC_CGM) may incorrectly contain the value 0b000 ("reserved") instead of the expected value of 0b100 ("switch to 16 MHz internal RC oscillator (IRCOSC) due to SAFE mode request or reset succeeded"). The system clock is still correctly sourced by the IRCOSC as expected.

Workaround:

Software should interpret CGM_SC_SS[SWTRG] = 0b000 right after a power-on reset as "switch to 16 MHz internal RC oscillator due to SAFE mode request or reset succeeded". In all other cases, it should be interpreted as "reserved".

1.170 ERR006082: LINFlexD : LINS bits in LIN Status Register(LINSR) are not usable in UART mode.

Description:

When the LINFlexD module is used in the Universal Asynchronous Receiver/Transmitter (UART) mode, the LIN state bits (LINS3:0]) in LIN Status Register (LINSR) always indicate the value zero. Therefore, these bits cannot be used to monitor the UART state.

Workaround:

LINS bits should be used only in LIN mode.

1.171 ERR006084: PFLASH: SAFE_CAL feature allows calibration remap when Word 2 of a redundant pair is mismatched.

Description:

When any of the 3 Calibration Remap Descriptor Words (CRDn.WORD0..2) of a redundant CRD pair are not equal, the Safe Calibration (SAFE_CAL) feature disables calibration remap. This feature works for CRDn.WORD0 and CRDn.WORD1 mismatches. However, a mismatch in CRDn.WORD2 incorrectly keeps calibration remap enabled.

Workaround:

Do not use Flash overlaying in a safety-relevant way in production series cars.

1.172 ERR006085: PSI5: "On the fly" Re-configuration and "On the fly Disabling" do not work

Description:

The Peripheral Sensor Interface (PSI5) cannot return to the DISABLE mode once it entered Normal or Config operating modes.

The PSI5 module has 3 operating modes:

- DISABLE where everything in the module is switched off, including the internal processing
- CONFIG where the register interface is available for configuration
- NORMAL where the module is actually working and receiving and processing PSI5 packets from the transceiver

Currently it is not possible to put the PSI5 module in DISABLE mode, from either the NORMAL or the CONFIG mode irrespective of the configuration of the PSI5 control registers. This prevents the reconfiguration of the module.

Workaround:

Issue a reset to the PSI5 module in order to put it in the DISABLE mode.

A reset of the PSI5 can be triggered by the setting the corresponding bit of the Peripheral Reset Register of the Reset Generation Module (RGM_PRSTx). Pay attention that in this case, the PSI5 module is entirely reset and the whole module must be reconfigured.

1.173 ERR006087: MC_RGM: Reset event during mode transition causes chip to remain in reset

Description:

If a reset due to any 'destructive' reset event, "functional" reset event, external Power On Reset pin (PORST) assertion, or External functional reset pin (ESR0) assertion occurs coincidental with a mode change request triggered via the Mode Control register (ME_CTL) in the Mode Entry module (MC_ME), the MCU enters reset but may not exit reset. The time window during which the reset event must occur for this to happen is one system clock period, so the likelihood is extremely small. However, if it does happen, the only way to recover the chip is through a power-on reset by powering down the chip and then powering it up again.

The time window during which a reset event occurring can cause this problem exists only if the system clock source is changed via the SYSCLK field in the target Mode Configuration register (ME_"target_mode"_MC) and, in the same mode change request, the new system clock source is being turned from off to on via the ME_"target_mode"_MC["clock_source"ON] bit.

Workaround:

Perform a power cycle of the MCU to recover from this event. If this is not possible in all cases, external circuitry should be installed that uses a General Purpose Output to hold off all external resets.

1.174 ERR006088: MC_RGM: Requested peripheral reset not applied as expected

Description:

If a peripheral is requested to be put into reset by writing a 0b1 to the appropriate bit of one of the Peripheral Reset registers (RGM_PRSTn) in the Reset Generation Module (MC_RGM), and then shortly thereafter requested to exit reset by writing a 0b0 to the same bit, it may occur that the peripheral in question does not actually get reset.

In addition, attempting to access any of the Peripheral Reset Status registers (RGM_PSTATn) in the MC_RGM, which are intended to allow software to see if a peripheral is currently in reset, results in an access error.

Workaround:

Software should wait long enough between the peripheral reset request and peripheral reset exit request to allow at least three peripheral clock cycles to occur in between. If a peripheral has multiple clocks, then the clock with the lowest frequency should be used as a gauge.

Software should also check whether the peripheral was actually reset by afterwards reading various registers and comparing them with their expected reset values.

Software should not try to access the RGM_PSTATn register locations.

1.175 ERR006090: MC_CGM: CGM_SC_DIV_RC register does not exist

Description:

The System Clock Divider Ratio Change register (CGM_SC_DIV_RC) in the Clock Generation Module (MC_CGM) does not exist. Accessing this register's location results in an access error.

Workaround:

Software should not try to access the CGM_SC_DIV_RC register location.

1.176 ERR006091: SDADC/SARADC: coupling current on ADC pin may be present when S/D are not enabled

Description:

The analog pins associated with the Analog to Digital Delta/Sigma Converter (SDADC) are shared with the Successive Approximation Register Analog to Digital Converter (SARADC). When none of the SDADC modules are enabled, a current injection (greater than 1mA) on one of the pins may generate a extra leakage (100nA at 25c) on the other SDADC pins.

Workaround:

In case one of these shared pins is to be used as a SARADC channel, at least one SDADC module should be enabled.

A SDADC_x module is enabled by setting (writing '1') the enable field (EN) of its Module Configuration Register (MCR) (SDADC_x_MCR[EN]=1).

In addition, to avoid any influence on input pins assigned to the chosen SDADC_x module, the Analog Channel Selection field (ANCHSEL) should be set to 0b100 (SDADC_x_CSR[ANCHSEL]=4) in order to connect the internal negative reference as input of the converter.

1.177 ERR006099: SDADC: FIFO data corruption is possible in certain configurations of the FIFO threshold

Description:

The Sigma Delta Analog to Digital Converter (SDADC) does not support a First-In-First-Out (FIFO) flush option. This can result in the FIFO having older data if the FIFO control register (FCR) FIFO threshold [FTHLD] value is programmed for a number of datawords less than the maximum depth of the FIFO, as indicated by the read only bits FCR[FSIZE].

In this configuration, if DMA (Direct Memory Access) is enabled and a DMA request is generated on a FIFO full condition (the number of written datawords is equal to the FTHLD value), once the DMA transfer for FTHLD datawords is completed, the FIFO can still contain some unread datawords (FSIZE-FTHLD maximum). This unread data will be present even after the SDADC is disabled by clearing the SDADC module configuration register enable bit (MCR[EN]), or after a change of channel selection by writing the channel selection register (CSR) analog channel select bitfield [ANCHSEL] with a new channel for a new conversion.

This condition can result in a DMA transfer on the next DMA request of a mix of older data (corresponding to the previously selected channel) and new data (corresponding to the next channel selection).

Workaround:

Software should read and discard all the older data from the FIFO before starting any new conversion.

1.178 ERR006183: SARADC/SDADC: reduced current injection immunity on channel SARB.AN36 and SD3.P0 associated to general purpose input pin PB[12]

Description:

The port B pin 12 (PB[12]) and its Analog channel 36 (AN36) is associated to both the Successive Approximation Register Analog to Digital Converter B (SARB, also known as SARADC_B) through Analog channel 36 (SARADC_B.AN36) and the Sigma/Delta Analog Digital Converter 3 (SD3, also known as SDADC_3) through positive input channel 0 (SDADC_3.P0).

Setting the Analog Pad Control (APC) bit of the Multiplexed Signal Configuration Register 28 (SIUL.MSCR28) to '1', activates an internal analog path which reduces the immunity of the pin PB[12] to current injection.

If current is injected on this pin, the performance of the SARADC is significantly degraded, impacting the conversion results of all ADC channels.

Workaround:

Either ensure that no current injection can occur on pin PB[12]; or Set MSCR28[APC] bit to '0'. In this configuration:

- SARB channel AN36 cannot be used and other SARB channels are not affected.
- SD3 can use the channel with no limitations.

1.179 ERR006194: e200z225: Nexus RWA to D-MEM simultaneous with core accesses may cause a write failure or ECC exception

Description:

If the e200z225 core (the Input/Output Processor, core 2) is used to perform a Nexus Read/Write Access (RWA) to the local data memory (D-MEM) while the core is also making accesses to this memory, one of two failures could occur: either an Error Correction Coding error may occur (causing an exception) or a write failure may occur.

Workaround:

Errors can be avoided by doing either of the following workarounds.

1. Put the e200z225 core into halt mode any time the Nexus RWA feature is required. or
2. Use the Computational/safety core (core 0) to perform Nexus RWA accesses to the D-MEM.

1.180 ERR006237: SIUL2: Weak pull-up not present on PB[11]

Description:

The pin PB[11] can not be configured with an internal weak pull-up.

In the System Integration Unit Lite2 (SIUL2), when asserting the Weak Pullup Enable bit (WPUE) of the I/O Pin Multiplexed Signal Configuration Register 27 (MSCR27) to '1', the pad is configured in high-impedance rather than in weak pull-up.

Workaround:

Do not configure PB[11] in weak pull-up mode. In case weak pull-up is required, external resistor should be added to the pin.

1.181 ERR006290: TDM: Base address of TDM is 0xFFF3_4000 instead of 0xFC0E_4000

Description:

The Tamper Detect module (TDM) is implemented at address 0xFFF3_4000. However, it should be located at address 0xFC0E_4000.

Workaround:

For this revision device, access the TDM at address 0xFFF3_4000, but be prepared to address it at address 0xFC0E_4000 in the future.

1.182 ERR006292: PFLASHC: Calibration Region Sizes 8 bytes and 16 bytes are not supported

Description:

When defining a calibration region of size 8 or 16 bytes by setting the PFlash Overlay Region Descriptor Calibration Remap Descriptor Size (PFCRDn.Word2[CRDSize]) field, wrong data may be returned on some accesses within the corresponding 32 byte page memory range. Coherency in the page buffers is not properly maintained when the given 32 byte page is composed of data from both the overlay RAM and the Flash array.

Workaround:

When establishing calibration remap regions, avoid defining calibration regions of size 8 or 16 bytes. PFCRDn.Word2[CRDSize] = 0b00011 and PFCRDn.Word2[CRDSize] = 0b00100 should be avoided.

1.183 ERR006349: LINFlexD: Possibility of incorrect break delimiter length in header by LIN master

Description:

When the Local Interconnect Network module (LINFlexD) is used in LIN mode, the length of the break field delimiter can be configured to either 1 or 2 bits using the Two Bit Delimiter bit (TBDE) of the LIN Control Register 2 (LINCR2).

Once LINCR2[TBDE] has been set (selecting 2 bits length), it is no longer possible to clear it (selecting 1 bit length).

This may result in LIN master always transmitting the header with two bits of delimiter in the break field.

Workaround:

To clear LINCR2[TBDE] field, a system reset has to be performed.

1.184 ERR006350: LINFlexD : WLS feature cannot be used in buffered mode.

Description:

The LINFlex module may not operate correctly if the Special Word Length (WLS) for enabling 12-bit data length is selected in the Universal Asynchronous Receiver/Transmitter (UART) Mode Control Register (UARTCR) and configured in the transmit buffered mode.

Workaround:

When WLS mode is required, always use the First In, First Out (FIFO_ mode of the UART LINFLEX module by setting the Transmit FIFO/Buffer mode bit of the UARTCR (UARTCR[TFBM]=1).

1.185 ERR006353: SARADC: Unexpected end of injected chain interrupt generation

Description:

In case no injected channel is selected for conversion by writing in the ICJCMRn (n=0, 1, 2) (Internal Channel Injected Conversion Mask Registers) / TCJCMR (Test Channel Injected Conversion Mask Register) / ECJCMRn (n=0, 1, 2) (External Channel Injected Conversion Mask Registers) and if the bit JSTART (Injected Start conversion) in the MCR (Main Configuration Register) is set to initiate the conversion(s), the bit JECH (End of injected chain conversion interrupt flag) in the ISR (Interrupt Status Register) gets set and an interrupt is generated, if previously enabled by setting the MSKJECH (Mask bit for JECH) in the IMR (Interrupt Mask Register).

Workaround:

Do not to initiate any injected chain conversion if no channel is selected for conversion.

1.186 ERR006361: INTC: Interrupt Priority Inversion can occur on a write to INTC_CPRn[PRI]

Description:

When the External Interrupt Enable (EE) field of the core's Machine Status Register (MSR) is 0 (MSR[EE]=0) and a write occurs to the Priority (PRI) field of the Interrupt Controller's (INTC) Current Priority Register for Processor n (INTC_CPRn) the INTC uses the previous INTC_CPRn[PRI] value when re-evaluating the interrupt requests instead of the one just written. This means that a pending interrupt prior to the store of a higher priority value to the INTC_CPRn[PRI] could be serviced, known as a priority inversion.

Workaround:

Software workaround is to write to the INTC_CPRn[PRI] twice in a row.

1.187 ERR006364: e200zx: Access violations on store references may cause unexpected machine check exception.

Description:

Store references that violate the privileges as defined in the System Memory Protection Unit (SMPU) may be handled as machine check exceptions, rather than data storage exceptions. This applies to illegal store references initiated by any core, but in particular, illegal store references which traverse from the computational shell to the peripheral shell, or vice versa.

Workaround:

Disable burst writes when traversing from computational shell to the peripheral shell, or vice versa by clearing Platform Configuration Module Burst Enable Register Burst Enable bits for Master port 3 and Slave port 0 (IAHB_BE1[BWE_S0] and IAHB_BE1[BWE_M3]).

1.188 ERR006369: PAD_RING: Pull current does not meet updated specification

Description:

The weak pull current of the general purpose input/output port are below the value currently specified in the datasheet. The current in the datasheet is defined with respect to the threshold of the input buffer: input voltage (Vin) = Voltage input High (Vih) for pull-up, Vin = Voltage Input Low (Vil) for pull- down.

As implemented, the weak pull current is defined with respect to supply: Vin= 0.0V for weak pull-up, Vin=VDD_HV for weak pull-down. Weak pull current is thus defined as below:

- The pull-up current is in the range 25uA-100uA for Vin = 0.0V with VDD_HV_IO supply in the range 4.0V < VDD_HV_IO< 5.9V.
- The pull-down current is in the range 30uA-80uA for Vin = VDD_HV_IO with VDD_HV_IO supply in the range 4.0V < VDD_HV_IO< 5.9V

Workaround:

If need, application should implement external resistance to increase pull-up/pull-down current as required

1.189 ERR006370: PMC:Temp Sensor Status bits don't exist in Gauge Status register.

Description:

The Temperature Sensor Over-temperature 2 (TS3), Temp Sensor Overtemp1 (TS2), and Temp Sensor Under-temperature (TS0) flags do not exist in the Gauge Status Register (GR_S) of the Power Management Controller (PMC).

Workaround:

Do not use the TS3, TS2, or TS0 bits in the GR_S register. Instead, poll the flags in the Temperature Event Pending register (EPR_TD): bit fields TEMP_3, TEMP_2, and TEMP_0.

1.190 ERR006373: PLL: PLL0 and PLL1 Loss of Lock Interrupts sent to both Interrupt Controller and FCCU

Description:

Both the Interrupt Controller (INTC) and the Fault Collection and Control Unit (FCCU) implement Phase-locked loop (PLL) Loss of Lock detection mechanisms.

Only the FCCU Loss of Lock feature should be used by software.

Workaround:

Do not use Interrupt Controller Channels IRQ #480 and IRQ #482 to service PLL0 and PLL1 Loss of Lock Interrupts. Use FCCU fault channels #29 and #30 instead.

1.191 ERR006383: SIPI: 16 bit writes/reads may access incorrect addresses

Description:

bit read and writes operations that are not on a 32-bit word boundary alignment will not operate correctly in the Serial Interprocessor Interface. 16-bit accesses to any address that matches 0x2, 0x6, 0xA or 0xE will access an incorrect address.Therefore writes will be to an incorrect address and reads will return data from the wrong address. 16-bits accesses to addresses matching 0x0, 0x4, 0x8, and 0xC will operate properly.

Workaround:

Only use 32- or 8-bit write/reads instead of 16-bit for all SIPI transactions.

1.192 ERR006401: e200zx: Circular Addressing issue on LSP Load/Store instructions, and zcircinc instruction

Description:

The circular addressing mode of the e200z Lightweight Signal Processing (LSP) Auxiliary Processing Unit for the circular increment (zcircinc) and Load/Store (zl*, zs*) instructions do not wrap properly in some cases when using positive offset.

Workaround:

1. Use one of the following options to workaround the issue.
2. Always use negative offset with these instructions; or

For a small buffer size of 1, 2, 3, or 4 double-words (8,16,24, or 32 bytes), a positive offset can be emulated by using a negative offset value equal to the "buffer length in bytes - desired_positive_offset". An example for a buffer length of 2 double-words with a desired offset of 2 bytes, an offset of 2-16=-14 can be used;

or

3. Use ODD index and EVEN positive offset greater than 1.

1.193 ERR006409: GTM: ATOM Force Update does not activate a comparison when in SOMC mode

Description:

(GTM-IP-139)

When the Generic Timer Module (GTM) ARU Connected Timer Output Unit (ATOM) is configured in Signal Output Mode Compare (SOMC) mode, with the Advanced Routing Unit (ARU) Enabled, and if no comparison is active (no valid data was received by ARU, ATOM[i]_CH[x]_STAT.B.DV = 0) a Force Update Request can cause the ATOM to remain in a state waiting for the update event to happen.

A Force Update is requested by first setting CPU Write Request field (WR_REQ) in ATOM[i]_CH[x]_CTRL, then updating the Shadow Counter Registers (ATOM[i]_CH[x]_SRx) and optionally the ATOM Mode Control Bits (ACB) in ATOM[i]_CH[x]_CTRL, and finally updating the Counter Registers (ATOM[i]_CH[x]_CMx) via a Forced Update (ATOM[i]_TGCx_FUPD_CTRnL = 0b11).

Under the above conditions:

- The registers CMx are updated correctly but no new comparison is activated.
- The ACBO bits are erroneously not cleared.
- The ARU read request is canceled because of WR_REQ=1.

Workaround:

After the Forced Update, re-write the desired values to CM0 or CM1 to activate the comparison and to reset the ACBO bits.

1.194 ERR006410: GTM: Write to ATOM_CH_CTRL sets WRF if CCU0 compare match has already occurred, but CCU1 compare match is pending, in ATOM SOMC mode

Description:

(GTM-IP-140)

When the Generic Timer Module (GTM) ARU Connected Timer Output Unit (ATOM) is configured in Signal Output Mode Compare (SOMC) mode, with the Advanced Routing Unit (ARU) Enabled, the capture/compare strategy is 'Serve Last' (ATOM[i]_CH[x]_CTRL.B.ACB42 = 0b1xx), with Counter Compare Unit 0 (CCU0) matched but the CCU1 match is pending, a write access to the ATOM Channel Control register (ATOM[i]_CH[x]_CTRL) will set the "Write Request of CPU Failed for the Last Update" (WRF) field in the Channel Status Register (ATOM[i]_CH[x]_STAT) independently of the status of the CPU Write Request Bit for Late Compare field (WR_REQ) in ATOM[i]_CH[x]_CTRL.

Workaround:

If ATOM[i]_CH[x]_CTRL is written during an active 'Serve Last' comparison without the intention of setting WR_REQ, write to ATOM[i]_CH[x]_CTRL and then reset/clear WRF in ATOM[i]_CH[x]_STAT by writing a '1'.

1.195 ERR006411: GTM: Incorrect Input Signal Characteristics when the TIM channel is in TIEM, TPWM, TIPM, TPIM or TGPS mode, when ECNT is selected as the captured GPRi value.

Description:

(GTM-IP-141)

When a Generic Timer Module (GTM) Timer Input Module (TIM) channel captures a valid rising edge event at TIM[i]_CH[x]_FOUT (post filtering) in TIM Input Event Mode (TIEM), TIM PWM Measurement Mode (TPWM), TIM Input Prescaler Mode (TIPM), TIM Pulse Integration Mode (TPIM) or TIM Gated Periodic Sampling Mode (TGPS), the captured values of the Edge Counter (TIM[i]_CH[x]_ECNT) to the General Purpose Registers (TIM[i]_CH[x]_GPRi) are incorrect. The captured value will be ECNT+2; bit 0 (signal level) will be 0 (Falling Edge). The correct operation would be to capture ECNT+1; bit 1 (signal level) would be 1 (Rising Edge).

This leads to an inconsistency between the ARU signal level bit, bit 0 of the ARU word which shows the captured ECNT value, and TIM[i]_CH[x]_GPRi which shows an inconsistency when comparing GPRi [bits 31:24] to ECNT [bits 7:0].

Workaround:

When using the TIMs captured data the correct data can be reconstructed by:

if ARU_SIGNAL_LEVEL ==1 and ARU_DATA[0] == 0 the ARU_DATA = ARU_DATA -1;

When reading TIM[i]_CH[x]_GPRi by the data can be corrected as long as there is no GPR overflow and no new edge by:

if TIM[i]_CH[x]_GPRi[24] == 1 and TIM[i]_CH[x]_GPRi[0] == 0 then TIM[i]_CH[x]_GPRi[23:0] = TIM[i]_CH[x]_GPRi[23:0] -1;

1.196 ERR006412: GTM: Incorrect Input Signal Characteristics when the TIM channel is in TBCM mode and ECNT is selected as the captured GPR0 value.

Description:

(GTM-IP-142)

When a Generic Timer Module (GTM) Timer Input Module (TIM) channel captures an input pattern match condition in TIM Bit Compression Mode (TBCM), the values captured by the General Purpose Register TIM[i]_CH0_GPR0 are incorrect under the condition EGPR0_SEL=1 with GPR0_SEL= 0 (use ECNT as input).

Starting at t=0 with counter value ECNT(t=0), the captured values of two consecutive edges can be ECNT(t=0)+2 followed by ECNT(t=0)+2 instead of ECNT(t=0)+1 followed by ECNT(t=0)+2. The captured ECNTs do not increment by 1 as expected, and reading TIM[i]_CH0_GPR0 shows an inconsistency when comparing ECNT [bits 31:24] to GPR0 [bits 7:0].

Workaround:

Ignore the captured data via the Advanced Routing Unit (ARU) and construct the edge count value with an independent Multi Channel Sequencer (MCS) counter which increments on each ARU transfer, or when reading TIM[i]_CH0_GPR0 use only TIM[i]_CH0_GPR0[31:24] as EDGE counter; do not use TIM[i]_CH0_GPR0[23:0].

1.197 ERR006427: LINFlexD: Communication failure when LIN timer is used in Output Compare mode

Description:

In the Local Interconnect Network Module (LINFlexD), the LIN state machine can be reset to the IDLE state on a timeout event, setting the Idle on timeout bit (IOT) bit of the LIN Time-Out Control Status Register (LINTCSR).

If the Timeout counter mode is set as Output compare (LINTCSR[MODE]=1), even if the Idle on Timeout bit is set (LINTCSR[IOT]=1), the LIN state machine does not move to IDLE state on timeout event.

Thus, further incoming data would not be interpreted correctly leading to communication failure.

Workaround:

The LIN state machine can be reset to the IDLE state on timeout event only using "LIN mode" as the timeout counter mode (LINTCSR[MODE] =0).

1.198 ERR006444: SARADC: A normal conversion interrupted by an injected channel conversion may restore back to a non-implemented channel

Description:

If in the Successive Approximation Register Analog to Digital Converter (SAR ADC) a normal conversion chain is on-going on non-contiguous channel numbers that have channel numbers that are not implemented in between them and a channel conversion is injected over a channel which is adjacent to an unimplemented channel the following issue occurs:

- The normal channel over which the injected conversion is requested fails to be converted. Instead, an erroneous conversion is done on an unimplemented channel adjacent to the next channel to be converted.
- The normal channel is not converted (loss of conversion), but the previous value of the normal channel is not corrupted.
- The overall conversion time increases as a dummy conversion on an unimplemented channel is done in between normal conversions.

Example of such behavior is provided below:

a) Channels 15,16, 20 and 21 are in the normal scan chain
b) Channels 24 and 25 are in the injected chain
c) Channels 17, 18 and 19 are not implemented
d) Injection of channels 24 and 25 occurs during channel 16 conversion. After completion of injected chain, dummy channel 19 (=20-1) is converted instead of channel 16.

In this example the impact is: Channel 16 conversion is missed because of injected conversion and overall conversion has a dummy conversion on channel 19 which is not implemented.

Workaround:

In Hardware Injected Conversion mode, when the Main Configuration Register Injection trigger enable is set (MCR[JTRGEN] = 1) or the Injection trigger sequence enable is set (MCR[JTRGSEQ] = 1):

Do not use hardware injected conversion mode when a normal conversion is ongoing, in other words, do not initiate and interleave hardware injected and normal conversions together.

In Software Injected Conversion mode, when MCR[JTRGEN and JTRGSEQ]=0:

Initiate Injected conversion only when Main Status Register Channel under measure address MSR[CHADDR] field is pointing to a channel number other than channel number adjacent to an unimplemented channel. For example, if channel17 is unimplemented don't initiate an injected conversion when MSR[CHADDR]=16.

1.199 ERR006445: SARADC: Status bits may not clear as expected if PBRIDGE clock is greater than ADC Clock

Description:

Clearing of Successive Approximation Register Analog-to-Digital Converter (SARADC) Interrupt Status Register End of Conversion / End of Chain ISR[EOC,JEOC,NECH,JECH] bits and Main Configuration Register Normal Start Conversion MCR[NSTART] bit can fail when the Peripheral Bridge (PBRIDGE) Clock frequency is greater than ADC Clock (AD_ck) frequency.

Due to inter-clock domain synchronization, A delay of 4 ADC clocks may be requried to complete.

Note: Clearing of ISR[EOC,JEOC,NECH,JECH]: depending on the clock phase relationship between the PBRIDGE Clock and ADC clock cycle, it may not be possible to clear ISR[EOC,JEOC,NECH,JECH] during the first 4 ADC clock cycles after the bit has been set by the SARADC. After 4 ADC clock cycle, clearing of this bit will always always.

Clearing of MCR[NSTART]: it may not be possible to clear MCR[NSTART] with one write/clear access. Depending on the clock phase relationship between the PBRIDGE Clock and ADC clock cycle, the first write/clear access will stop the conversion chain, but the write may not correctly clear the MCR[NSTART] bit. Successive write/clear operation, occurring within 4 ADC clock cycles after the initial clear may still fail. 4 ADC clock cycle after initial write/clear access, further write/clear access to the MCR[NSTART] bit will always be effective.

Workaround:

The user must implement a while loop which clears these bits and waits until the value read back is cleared.

1.200 ERR006459: SIUL2: ESR0 pin cannot be forced low by software

Description:

The External System Reset Assert (ESR0_ASSERT) bit of device Configuration Register 0 (SIUL2.SCR0) in the System Integration Unit Lite2 module (SIUL2) is not implemented. Writing to this bit has no effect, and it always returns a value of 0b0. Therefore, it is not possible for software to force the ESR0 pin low other than forcing a reset using an alternate software method.

Workaround:

In order for software to have the device assert the ESR0 pin, it must trigger a software reset by requesting a mode change to RESET via the Mode Control Register (ME_MCTL) in the Mode Entry module (MC_ME).

If a short 'functional' reset is desired (reset sequence starting at the beginning of reset PHASE3[FUNC]), software must set the BE_SOFT_FUNC bit in the ‘Functional’ Bidirectional Reset Enable register (RGM_FBRE) in the Reset Generation Module (MC_RGM) to 0b0 and the SS_SOFT_FUNC bit in the ‘Functional’ Event Short Sequence register (RGM_FESS) to 0b1 prior to requesting the mode change with ME_MCTL[TARGET_MODE] = 0b0000.

If a long 'functional' reset is desired (reset sequence starting at the beginning of reset PHASE1[FUNC]), software must set the RGM_FBRE[BE_SOFT_FUNC] bit to 0b0 and the RGM_FESS[SS_SOFT_FUNC] bit to 0b0 prior to requesting the mode change with ME_MCTL[TARGET_MODE] = 0b0000.

If a 'destructive' reset is desired (reset sequence starting at the beginning of reset PHASE0[DEST]), software must simply request a mode change with ME_MCTL[TARGET_MODE] = 0b1111.

ESR0 is automatically deasserted by the device on the exit of the reset sequence.

1.201 ERR006477: RGM: minimum PORST pulse is not guaranteed

Description:

When a destructive reset is triggered internal to the device by the Reset Generation Module (RGM), the bidirectional Power On Reset signal (PORST) is asserted. The PORST reset pulse width driven by the device may be lower than the specified 5us and could be as short as 0.5us. This pulse may not be visible depending on external circuitry configuration.

Workaround:

Use the External functional reset with Schmitt trigger (ESR0) signal to externally monitor the internal reset status.

1.202 ERR006532: PAD_RING: wrong strength for output port PA[5], PA[6], PA[8], PA[9] and PC[2]

Description:

The following pads have limited strength configuration:

- PA[5], PA[6], PA[8] and PA[9] implement only SLOW and MEDIUM configuration. It is not possible to select STRONG and VERYSTRONG configuration using the Output Edge Rate Control field of the System Integration Unit Lite2 Multiplexed Single Configuration Register (SIUL2_MSCR[OERC]).

Selecting STRONG pad writing SIUL2_MSCR[OERC] = "2" results in selecting a SLOW configuration. Selecting VERY STRONG pad writing SIUL2_MSCR[OERC] = "3" results in selecting a MEDIUM configuration.

- PC[2] implements only SLOW and MEDIUM configuration. It is not possible to select STRONG configuration using the SIUL2_MSCR[OERC] register field. Selecting STRONG pad writing SIUL2_MSCR[OERC] = "2" results in selecting a SLOW configuration.

Workaround:

Do not use STRONG configuration on general purpose Input output port PA[5], PA[6], PA[8], PA[9] and PC[2].

Do not use VERYSTRONG configuration on general purpose Input output port PA[5], PA[6], PA[8] and PA[9].

1.203 ERR006538: LINFlexD: Stop mode request may be ignored if requested before the end of a frame

Description:

The LINFlexD module fails to enter stop mode when the stop mode request is issued before the second data byte of an ongoing frame transfer.

User requests stop mode by setting the appropriate bit of the Peripheral Control Register of the Mode Entry Module (ME_PCTLx).

Workaround:

If a LIN transmission/reception is in progress, wait until it reaches the frame boundary (complete current frame transfer) before sending a request to enter in stop mode.

1.204 ERR006544: e200zx: Unsuccessful cache line fills may corrupt subsequent references by the cores

Description:

Unsuccessful cache line fills may corrupt subsequent references by the e200zx core. On a subsequent load reference following an unsuccessful cache fill, wrong data may be returned to the core. On a subsequent store reference following an unsuccessful cache fill, the store may be skipped, leaving stale content in the target slave. This behavior applies to all e200zx cores in the device.

Workaround:

Clear the Burst Read Enable S0 (BRE_S0) field and BRE_M3 field in the Intelligent AHB Burst Enable 1 (IAHB_BE1) register located at address 0xFC02_8004.

1.205 ERR006552: MC_RGM: Reset event during mode transition causes chip to remain in reset

Description:

If a reset due to any 'destructive' reset event, 'functional' reset event, external Power On Reset pin (PORST) assertion, or External functional reset pin (ESR0 or RESET, depending on the MCU) assertion occurs coincidental with a mode change request triggered via the Mode Control register (ME_CTL) in the Mode Entry module (MC_ME), the MCU enters reset but may not exit reset. The time window during which the reset event must occur for this to happen is one system clock period, so the likelihood is extremely small. However, if it does happen, the only way to recover the MCU is through a power-on reset by powering down the device and then powering it up again.

Workaround:

Perform a power cycle of the MCU to recover from this event. If this is not possible in all cases, external circuitry should be installed that uses a General Purpose Output to hold off all external resets during mode changes.

1.206 ERR006553: 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.

1.207 ERR006597: SRX: Pad input low level threshold (Vil) variation is not guaranteed to remain within +/-50mV on 1ms window

Description:

The Single Edge Nibble Transmission (SENT) specification requires drift on the low threshold of its input buffer (Vil) to be less than 50mV during a 1ms window. However, noise, linked to activity on pads in the neighborhood of the SENT interface, can create a local voltage drop on the VSS_HV_IO_MAIN/VDD_HV_IO_MAIN leading to up to 200mV of Vil drift when the pins are configured in Automotive level configuration. The Input Level Select (ILS) bits in the Multiplexed Signal Configuration Register (MSCR) select the pin voltage levels. The SENT interface full functionality may not be guaranteed if supply segment associated with the SENT interface is using more of than 25% of the segment capability, as defined in the datasheet IO signal table addendum.

Workaround:

Ensure activity on the power supply segment is limited to 25% of the maximum if the SENT interface must be used.

1.208 ERR006638: PASS:Incorrect Censor reset value

Description:

In the Device Configuration Format (DCF) Records, the reset value for the censorship client has the lower 16-bit value configured to 0x55AA. For the Password and device security module (PASS), this value means the device is not censored out of reset while the default state of the device should be censored, but allowing access to the device via a debugger/flash programmer.

Workaround:

In order to have the device censored out of reset, add a DCF record setting the data for the censorship client to any 32-bit value different from 0xUUUU55AA (only the lower 16 bits are considered).

1.209 ERR006639: GTM: A compare match event does not clear WR_REQ when ATOM is in SOMC mode

Description:

(GTM-IP-146)

If a Generic Timer Module (GTM) ARU Connected Timer Output Module (ATOM) channel is operating in Signal Output Mode Compare (SOMC) mode, with the Advanced Routing Unit (ARU) enabled and a late compare register update is requested by the CPU by setting WR_REQ (CPU Write request bit for late compare register update in ATOM[i]_CH[x]_CTRL), a late update of the Counter Compare Unit (CCU0/1) Compare registers (CM0/CM1) and/or the compare strategy (i.e. ARU Control Bits (ACB) bits altered) is successfully done, but after final compare match the WR_REQ bit is not reset. As a result no new ARU read request is set up after final compare match.

Workaround:

CPU should clear WR_REQ by software after the late update.

1.210 ERR006640: GTM: Valid edge after Timeout event ignored by TIM

Description:

(GTM-IP-150)

When a Generic Timer Module (GTM) Timer Input Module (TIM) timeout event triggers an Advanced Routing unit (ARU) write request with timeout information "timeout detected without valid edge" (ARU Control Bit 2 (ACB2)=1 and ACB1=0), and this request is acknowledged by the ARU at the same time as a new valid edge occurs, the valid edge is

neither acknowledged by setting the bits ACB2=1 and ACB1=1 (timeout detected with subsequent valid edge detected) within the acknowledged transfer nor acknowledged by setting up a subsequent ARU write request for the new valid edge with ACB2=0 and ACB1=0 (valid edge detected).

Workaround:

The workaround for this issue requires an additional plausibility check within the Multi Channel Sequencer (MCS) or CPU via FIFO:

Step 1) store the received data (ARUDATA(47:0)) and ACB0 in temporary variables.

Step 2) if an ARU transfer with ACB2=1 and ACB1=0 is received also check whether the previously received ARUDATA(47:0) and ACB0 values are the same:

If they are not the same values then a timeout with subsequent valid edge has occurred, which means ACB1 must be corrected to 1.

1.211 ERR006642: GTM: THVAL not available immediately after inactive trigger in DPLL

Description:

(GTM-IP-152)

The Generic Timer Module (GTM) Digital PLL (DPLL) Measured TRIGGER hold time value (THVAL) is calculated correctly for each INVALID trigger slope, but this value is only stored into the THVAL memory location in RAM Region 1a with every new ACTIVE edge of the trigger signal.

Workaround:

If the THVAL value is needed immediately with the inactive trigger edge it is necessary to calculate the THVAL value by using Timer Input Module Channels 0/1 (TIM_CH0/1) to obtain the active and inactive slopes in TIM input event mode (TIEM). With these timestamps the CPU is able to calculate the time span.

1.212 ERR006643: GTM: Incorrect timestamp captured in CNTS when TIM operates in TPWM or TPIM modes if CMU_CLK is not equal to system clock

Description:

(GTM-IP-153)

When a Generic Timer Module (GTM) Timer Input Module (TIM) channel operates in the following configuration, the Timebase Unit Channel 0 (TBU_TS0) value is not captured correctly in to the counter register CNTS:

- PWM Measurement Mode (TPWM) or Pulse Integration Mode (TPIM) and
- TBU_TS0 selected as the input to CNTS (CNTS_SEL = 1 in TIM[i]_CH[x]_CTRL) and
- selected clock (CMU_CLKn) of the TIM channel is not the same frequency as the system clock.

Workaround:

Configure the TIM channel to operate on a CMU_CLK (Divider =1) which is identical to the system clock when TBU_TS0 is to be captured in TPWM or TPIM modes.

1.213 ERR006644: GTM: Incorrect duty cycle in TOM PCM mode

Description:

(GTM-IP-154)

The Generic Timer Module (GTM) Timer Output Module (TOM) duty cycle output in Pulse Count Modulation (PCM) mode is always one count less than the configured value in the Compare Match 1 (CM1) register. For example, if the value 1 is written to CM1, a duty cycle of 0% will be generated and if a duty cycle of 100% was configured by writing the maximum value (0xFFFF) in to the CM1 register, the resultant waveform would be a signal with duty cycle less than 100%. A value of zero in the CM1 register results in 100 % duty cycle.

Workaround:

Configure the CM1 value for the targeted duty cycle in the CM1 register with target duty cycle + 1.

To get 0% duty cycle, CM1 = 1.

To get 100% duty cycle, CM1 = 0 and CM0 = 0xFFFF. (Setting CM0 = 0x1000 and CM1 = 0xFFFF will also result in a duty cycle of 100%).

1.214 ERR006645: GTM: Clearing of DPLL PCM1/2 bits after the Missing Pulse Correction Values calculations delayed

Description:

(GTM-IP-158)

The Generic Timer Module (GTM) Digital PLL (DPLL) Pulse Correction Mode bits (PCM1/2 in DPLL_CTRL_1) are expected to be cleared after the Missing Pulse Correction Values (MPVAL1/2) are used to calculate the number of sub_incs for the next increment and to calculate the add_in values by the DPLL State Machine, however the PCM1/2 bit is transferred with an active edge into the dedicated shadow registers, but cleared some time later. If the PCM1/2 bits are written by the CPU in between the point of time of the transfer to the shadow register and the point of time were the PCM1/2 bits are cleared, the bits are cleared and never used.

Workaround:

Do not allow the CPU to write to the PCM1/2 bits until at least 750 system clocks have passed since the previous Trigger Active Slope Detected (TASI) interrupt. This time could be derived by an GTM resource like an ARU connected TOM (ATOM) channel.

1.215 ERR006720: SIUL2: Logic state of LVDS input pads cannot be read via GPDI registers.

Description:

When two adjacent pads are configured as LVDS inputs, the associated System Integration Unit Lite (SIUL2) General Purpose Data Input (GPDI) registers for each pin are still connected to the CMOS input buffer. Therefore, the actual state of the LVDS input cannot be read since the LVDS logic levels don't align to TTL, CMOS, or Automotive input levels. After settling, these GPDI registers will always read zero when the associated pads are configured as LVDS inputs.

Workaround:

There is no work-around available to read the LVDS logic state for diagnostic purposes on this revision of the device. In future revisions, the GPDI register for the lowest numbered port of the LVDS pair will reflect the LVDS logic state of the pin when in LVDS input mode. The SIUL2 Multiplexed Signal Configuration Register Input Buffer Enable (MSCR[IBE]) bit for the associated port must be enabled to read the logic state, in addition to selecting LVDS inputs with the Input Level Selection (MSCR[ILS]) field.

1.216 ERR006738: PAD_RING: No TTL or AUTO levels on JTAG pins

Description:

Only CMOS input levels are implemented on the JTAG pins on this revision of the device. To support 5V operation with connection to 3.3V debug tools, TTL levels are added on future revisions of the device and will be selected by default. It will also be possible to select Automotive levels on the JTAG pins.

Workaround:

In order to communicate with 3.3V debug tools (after a power up), the power supply for the JTAG pins (VDD_HV_IO_JTAG or VDD_HV_JTAG) must be powered to 3.3V +/-10%.

1.217 ERR006792: JDC: The JDC JTAG input IPS data (JIN_IPS) register and JTAG data out (JOUT) register are only reset by JCOMP.

Description:

The JTAG input IPS data (JIN_IPS) register is only reset when JCOMP is driven low. Device destructive reset does not reset JIN_IPS. As a result, assertion of PORST or other device destructive reset will not clear the JIN_IPS contents.

The JTAG data out (JOUT) register is only reset when JCOMP is driven low. Device destructive reset does not reset JOUT. As a result, assertion of PORST or other device destructive reset will not clear the JOUT contents. However, device destructive reset does reset the system clock domain registers containing the JOUT_IPS, JIN_RDY, and JOUT_RDY bits that are read via JOUT. It can take up to three TCK periods for the updated values of the JOUT_IPS, JIN_RDY, and JOUT_RDY bits to be captured into the JOUT register, so any read of JOUT should begin at least three TCK periods after assertion of the device destructive reset to guarantee a correct read of the reset values.

Workaround:

The JCOMP pin must be driven low to clear the contents of the JIN_IPS register.

Wait at least three TCK periods after assertion of device destructive reset to begin execution of the JOUT_READ instruction in order to read the JOUT contents correctly.

1.218 ERR006797: FCCU: SMPU faults are not forwarded to FCCU

Description:

The Fault Collection and Control Unit (FCCU) will not report the two following alarms from the System Memory Protection Unit (SMPU):

1. When the SMPU refuses an access incorrectly, the failure is indicated in the SMPU but not forwarded to FCCU.
2. SMPU does not forward the prevention of an unauthorized access to the FCCU.

Workaround:

Be aware that these two SMPU faults are not forwarded to the FCCU.

1.219 ERR006800: PMC: Additional supplies are required to exit from reset.

Description:

All voltage supplies (important supplies and additional supplies) must be at the correct levels before an exit from reset (functional reset, destructive reset, or Power On Reset (POR)) is allowed.

The important High Voltage supplies are the PMC High Voltage supply (VDD_HV_PMC), Flash 3.3V supply (VDD_HV_FLA), I/O supply (including JTAG) (VDD_HV_IO_MAIN, VDD_HV_JTAG), Oscillator supply (VDD_HV_OSC), and ADC supply (VDD_HV_ADV).

The important Low Voltage supply is the Core and Flash voltage (VDD_LV).

The additional supplies that are needed for this device to exit from reset are:

high voltage Flexray supply (VDD_HV_IO_FLEX, VDD_HV_IO_FLEXE), low voltage PLL supply (VDD_LV), and EBI supply (VDD_HV_IO_EBI) if present on the device).

Note: VDD_LV supplies the Core, Flash, and PLL low voltage power supplies.

In future revisions of the device, only the important power supplies must be at the correct value.

Workaround:

Provide all the supplies (important supplies and additional supplies) to exit from reset - not just the important supplies.

1.220 ERR006803: MEMU: Contents are cleared upon functional reset.

Description:

The Memory Error Management Unit (MEMU) is reset by a functional reset. This will cause the contents of the registers to be reset to their default conditions. Since functional reset occurs after the off-line self-test, any Memory Built-In Self-Test (MBIST) errors detected during the off-line self-test will not be preserved and are cleared by the functional reset.

Workaround:

Restore MEMU contents following reset to prevent new notifications of errors. Perform MBIST using on-line self-test to verify the integrity of memory.

1.221 ERR006804: CJTAG: Performing a mode change from Standard Protocol to Advanced Protocol may reset the CJTAG.

Description:

In extremely rare conditions, when performing a mode change from Standard Protocol to Advanced Protocol on the IEEE 1149.7 (Compact JTAG interface), the CJTAG may reset itself. In this case, all internal CJTAG registers will be reset and the CJTAG will return to the Standard Protocol mode.

Workaround:

If the CJTAG resets itself while attempting to change modes from Standard Protocol to Advanced Protocol and Advanced Protocol cannot be enabled after several attempts, perform future accesses in Standard Protocol mode and do not use the Advanced Protocol feature.

1.222 ERR006806: FCCU: FOSU may issue a reset if only NMI is chosen as a reaction to incoming faults

Description:

When a channel in the Fault Collection and Control Unit (FCCU) forces the FCCU to the FAULT state due to a fault and only the Non-maskable Interrupt (NMI) is enabled as the reaction for this channel, the FCCU Output Supervision Unit (FOSU) will cause a destructive reset. However, the FOSU will not generate reset reaction, if another fault, which has reaction other than NMI enabled, is asserted before FOSU time-out.

Workaround:

For any channel, along with NMI, at least one more reaction must be enabled. The recommended reactions are to enable an Alarm interrupt (IRQ_ALARM_ENx) or an Error-out signal (EOUT_SIG_ENx).

1.223 ERR006812: TDM: DBA register and Software Tamper Override Key Diary[0-5] registers are not available

Description:

The Software Tamper Override Key Diary[0-5] registers, STO_KEY[0-5], are not available on this device revision. The address range corresponding to the STO_KEY[0-5] registers is treated as reserved memory space, and attempted references to this region will result in a bus error. The software mechanism to the tamper detect enforcements is not available on this revision of the device.

The Diary Base Address (DBA) register is not available in this revision of the device. The address range corresponding to the DBA register is treated as reserved memory space, and attempted references to this region will result in a bus error.

Workaround:

Avoid references to the STO_KEY[0-5] registers and the DBA register in the TDM.

1.224 ERR006815: PASS: Debug access in "OEM Production" Life Cycle UNLOCKED if no DCF record is programmed.

Description:

If the Censer Device Configuration Format (DCF) record is not programmed into the User Test flash block (UTEST) then Debug access in "OEM Production" Life Cycle is UNLOCKED.

Workaround:

In this revision of the device, do not expect Debug access to be locked if no DCF record is programmed.

In future revisions of the device, a Censer DCF record is added in Flash, such that the device is Censored from Power On.

1.225 ERR006816: PASS: Debug port may be enabled during functional reset

Description:

Independent of the lifecycle setting, the debug port is enabled during a functional reset if the devices is uncensored. The device is uncensored if the Censorship Device Configuration Format (DCF) client is set to its reset value of 0x55AA.

Workaround:

To disable the debug port, censor the microcontroller by using a Censorship DCF record with a value other than 0x55AA.

1.226 ERR006819: Flash: Flash read protection may be active in life cycle stage 'OEM production'

Description:

Flash secure read protection may be active in life cycle stage 'OEM production' and needs to be unlocked to read the flash when a debugger is attached.

Workaround:

The Flash can be unlocked by updating the LOCK3_PGn[RL4:0] bits in the PASS module.

This may be done in the application code, through the debugger or by using Device Configuration Format (DCF) records to initialize the device with unlocked Flash during system boot.

1.227 ERR006822: FLEXRAY: Capability to disable the FlexRay module by writing 0 to the FR_MCR[MEN] bit is not supported

Description:

The user cannot write a 0 to the FlexRay Mode Configuration Register Mode Enable bit (FR_MCR[MEN]) to disable the FlexRay module. The FR_MCR[MEN] disabled state (MEN = 0) is a read only indication.

In future device revisions, this option is implemented through the existing FR_MCR.MEN bit by making it writable with 0 to disable the FlexRay module. Note: The application software must ensure the FlexRay module is in Protocol Operation Control (POC): Default Config state before disabling the module.

Workaround:

Do not expect clearing the FR_MCR.MEN bit will disable the FlexRay module.

1.228 ERR006828: e200zx: Local Instruction and Data Memories are not accessible during corresponding e200z4/e200z7 core reset.

Description:

Local Instruction Memory (IMEM) and Data Memories (DMEM) are not accessible while its corresponding e200z4 / e200z7 core is in reset. Write accesses from cores not in reset will be ignored and read accesses will return all zeros (0x0) in the data and Error Correction Code checker bits. This will result in an End-to-End Error Correction Code (E2E ECC) error. The issue occurs when an e200z core reset is triggered by writing to the Mode Entry Core Control register (MC_ME.ME_CCTL0,1,3) RESET bit.

Workaround:

Applications should not access the IMEM and DMEM memories for e200z4 / e200z7 cores that are being held in reset via the ME_CCTLx registers.

1.229 ERR006836: DCF: DCF record for initial IVPR cannot be used

Description:

The Device Configuration Format (DCF) record for setting the initial value on the Interrupt Vectors base address is not recognized.

The DCF records for the initial IVPR uses DCF Chip Select 7 (Miscellaneous Client) addresses 0x10, 0x14, and 0x18 to set the IVPR for the different cores.

Core	DCF Address
Core 0	0x10
Core 1	0x14
Core 2	0x18

Workaround:

Write software such that it does not rely on the initial IVPR value to be set by a DCF record.

1.230 ERR006839: RGM: Out of temperature range Destructive Reset enable / disable feature is available in RGM_DERD[D_TSR_DEST].

Description:

Both the Power Management controller (PMC) and the Reset Generation Module (RGM) implement out of temperature range Destructive Reset configurations. Using both the RGM and the PMC to configure out of temperature range Destructive Reset may cause conflicts and/or confusion.

In future revisions of the device, the RGM out of temperature range Destructive Reset enable will not be supported.

Workaround:

Do not use RGM's out of temperature range Destructive Reset enable in the Destructive Event Reset Disable Register RGM_DERD[D_TSR_DEST]. Use the Power Management Controller (PMC) for out of temperature event configuration.

1.231 ERR006840: FLEXRAY: missing single/double error generation when reading listed FLEXRAY registers

Description:

During the FlexRay initialization phase of the Look-up table RAM (LRAM) in the Controller Host Interface (CHI), the data content of the following registers mapped in the LRAM may be corrupted:

- Message Buffer Cycle Counter Filter Register: FR_MBCCFRn
- Message Buffer Data Field Offset Register: FR_MBDORn
- Message Buffer Frame ID Register: FR_MBFIDn
- Message Buffer Index Register: FR_MBIDXR0n
- LRAM ECC Error Test Register: FR_LEETRz

with n is the number of Message buffers and z the number of LRAM banks.

If the application tries to read any of these registers during the FlexRay LRAM initialization phase, the data may be corrupted, but no single nor double Error Correction Coding (ECC) error is generated.

After completion of the FlexRay initialization phase, read accesses to these registers return correct data and errors are correctly reported if needed.

Workaround:

Do not read these registers until the CHI LRAM initialization is completed.

1.232 ERR006841: LFAST: default state of the termination resistor is off

Description:

The reset state of the Low Voltage Differential Signalling (LVDS) Fast Asynchronous Serial Transmission (LFAST) Control Register (LCR) termination resistor control fields (LVRXOP) is off (0b000). The reset state should be enabled (0b100).

Workaround:

The termination resistor should be enabled to use the LFAST interface.

1.233 ERR006846: Debug Zipwire: Clock not transmitted on pin PA[6] in the Production Device

Description:

When the Port A pin 6 (PA[6]) is enabled for operation as the reference clock for the debug Zipwire interface, the slew rate of the pad is set to slow. Therefore, it cannot generate a 10 or 20 MHz clock signal for the external Low Voltage Differential Signalling Fast Asynchronous Serial Transmission (LFAST) Interface.

Workaround:

Use pin PF[14] (The interprocessor communication interface Zipwire) as the debug Zipwire clock source or do not use the debug Zipwire interface.

1.234 ERR006847: INTC: PLL interrupts are implemented in IRQs 480, 482, 484-487

Description:

The Phase-Lock Loop (PLL) Interrupt Requests (IRQs) 480, 482, 484-487 are implemented in this version of the device.

In future revisions, the following interrupt sources assigned to PLL0 and PLL1 Status Registers are not supported: IRQ 480 PLL0SR Loss of lock flag (PLL0SR[LOLF]), IRQ 482

PLL1SR[LOLF], IRQ 484 PLL0SR External Power Down cycle Complete indication interrupt flag (PLL0SR[EXTPDF]), IRQ 485 PLL1SR[EXTPDF], IRQ 486 PLL0SR[CLKSW], and IRQ 487 PLL1SR[CLKSW].

The Fault Collection and Control Unit (FCCU) supports PLLnSR[LOLF] fault indication, therefore, a separate INTC supported IRQ is not necessary. The Mode Entry Module (MC_ME) ensures the proper on/off sequence of the PLLs, software doesn't need to get involved, and an interrupt for PLLnSR[EXTPDF] is not needed. The Clock Generation

Module (MC_CGM) manages clock switching, the PLLnSR[CLKSW] interrupts are not necessary.

Workaround:

Do not enable interrupt sources at slots 480, 482, 484-487 since the functionality of these IRQs will be removed on a future version of this device.

1.235 ERR006849: PSI5:Corrupt frame if the last data bit overlaps the Manchester disable time

Description:

When operating in synchronous mode, if the last slot data of a Peripheral Sensor Interface (PSI5) frame being transmitted is extended into the Manchester disable phase of a frame, the electrical error flag (E) will be set in the PSI5 Message Received Register high (PSI5_PMRRH) register and subsequent two frames after next sync pulse will be corrupted.

Workaround:

Set a gap of at least 8 us between the last bit of data in the last slot and Manchester disable.

1.236 ERR006852: FCCU: Peripheral reset to the FCCU may cause a functional reset of the MCU

Description:

A peripheral reset from the Reset Generation Module (RGM) to the Fault Collection and Control Unit (FCCU) may cause a functional reset of the MCU.

Glitches may cause the FCCU to go from a normal state to a FAULT state (even in the absence of incoming fault) and cause a system reset from the FCCU.

Workaround:

Check fault status registers; in the absence of any fault latched, assume the reset generated by FCCU was spurious.

1.237 ERR006855: PSI5: Late message arrival in synchronous mode causes corruption of next two subsequent messages

Description:

If the Peripheral Sensor Interface (PSI5) is In Synchronous Mode and the reception of slot(n) is such that it overlaps with the slot(n+1) message, slot(n+1) and the subsequent frame will be corrupted with both a Cyclic Redundancy Check/Parity error (C-bit in the PSI5 Message Receive Register Low [PSI5_PMRRL]) and an Electrical error (E-bit in the PSI5 Message Receive Register High [PSI5_PMRRH])) error in addition to corruption of the slot(n+1) message.

Workaround:

The start bit of slot (n+2) should programmed for an additional 8 us to provide additional margin.

1.238 ERR006858: PSI5: Gap of less than one bit period between last bit of frame and first start bit of next frame leads to idle gap violation

Description:

Setting the Peripheral Sensor Interface (PSI5) idle gap between slot(n) frame and slot(n+1) frame is less than 1 bit period (8us for 125kbps, 5.29us for 189kbps) can cause an idle gap violation. This results in electrical error in slot(n) frame and F bit error in slot(n+1) frame.

In addition, slot(n) frame may also have a timing error (T bit) in the PSI5 Message Receive Register High (PSI5_PMRRH[T]) if the idle gap has crossed its slot boundary, as idle gap is treated as part of this frame.

Workaround:

Always set the idle gap between two frames greater than 8 us.

1.239 ERR006860: PRAMC: PRCR1[P0_BO_DIS] and PRCR1[P1_B0_DIS] always read as zero

Description:

The Platform RAM Configuration Register Port p0 read burst optimization disable bit PRCR1[P0_BO_DIS] and Port p1 read burst optimization disable PRCR1[P1_B0_DIS] registers always read as zero. PRCR1[P0_BO_DIS] and PRCR1[P1_B0_DIS] are writable and control burst optimization behavior of the system RAM controller as defined, however these fields always return a zero when read.

Workaround:

Expect PRCR1[P0_BO_DIS] and PRCR1[P1_B0_DIS] to always read as zero, even when these fields are set.

1.240 ERR006863: Core_0: Safety core cannot be started if a non-maskable interrupt (NMI) is active

Description:

The safety core (Core_0) cannot be started if a non-maskable interrupt (NMI) is active:

The e200z7 cannot be started if NMI is active. If it is started with NMI active, a machine check will be executed before Interrupt Vector Prefix Register (IVPR) can be loaded with a valid value, resulting in additional machine checks.

The cache invalidate operation will be aborted if an NMI request is pending, resulting in status bits of the Data Cache and the Instruction Cache Abort bits in the Level 1 Cache Status Register (L1CSR0[DCABT] or L1CSR1[ICABT]) being set.

Workaround:

Non-maskable interrupt (NMI) requests must be cleared before reset/boot or cache invalidate operations.

1.241 ERR006865: FCCU: Flash errors during reset sequence may not be reported

Description:

The Flash module does not have a signal to the Fault Collection and Control Unit (FCCU) channel #5 to notify of an error during the Flash reset sequence.

Workaround:

Errors during the flash reset will not be reported to the FCCU.

1.242 ERR006873: PMC: Bondwire or pad failure of the VDD_LV cold sense signal may not be detected

Description:

In the Power Management Controller (PMC) a low voltage on the VDD_LV cold sense signal (LVD112_C) is indicated using the VD4_C bit in the PMC Pending Gauge Register (GR_P_[VD4_C]). The LVD112_C signal has its own dedicated device input pad (internal to the package). A failure of this pad or its bondwire will not be detected by the PMC and the EVPR_VD4[LVD4_C] bit will not flag. A failure of this nature would not affect the VDD_LV supplies as they have separate dedicated pads.

This is a safety feature, occurrence of failure is very low.

Workaround:

Customer software should not rely on the (EVPR_VD4[LVD4_C]) to detect a failure of the LVD112_C bondwire or pad.

1.243 ERR006896: PBRIDGE: AIPS0_MPRB and AIPS1_MPRB registers not available and Tamper Detect Module base address moved

Description:

The Master Privilege Registers (AIPS_MPRB) register in Peripheral Bridge A (PBRIDGE_A) and PBRIDGE_B are not implemented.

The address location, 0xFC00_0004, which corresponds to AIPS0_MPRB is treated as reserved memory space, and attempted references to this region will result in a bus error.

The address location, 0xF800_0004, which corresponds to AIPS1_MPRB is treated as reserved memory space, and attempted references to this region will result in a bus error.

Also, the Tamper Detection Module (TDM) base address is located at 0xFFF34000, which corresponds to AIPS0 off-platform slot #40.

AIPS0 on-platform peripheral slot #57 at base address 0xFC0E4000 is not available, and attempted references to this region will result in a bus error. Also, the reset value for Peripheral Access Control Registers H (AIPS0_PACRH[4:7]) is cleared, where these fields correspond to AIPS0 on-platform peripheral slot #57.

Defining peripheral access privilege levels for the following system bus masters is not supported on this revision of the device: Core_0 debug, Core_1 debug, Core_2 debug, Direct Memory Access (DMA_1), FlexRay_1 and Hardware Security Module (HSM).

Workaround:

Do not expect AIPS0_MPRB or AIPS1_MPRB registers to be available.

Reference the base address for the Tamper Detect Module at 0xFFF3_4000, instead of 0xFC0E_4000.

1.244 ERR006902: RGM: short functional reset may cause multiple ESR0 pulses when Progressive Clock Switching is configured

Description:

If Progressive Clock Switching (PCS) is enabled, the External System Reset 0 (ESR0) signal may assert multiple times during a short functional reset. Multiple assertions of ESR0 may occur under the following conditions:

1. PCS is configured and enabled.
2. Before entering reset, the system clock is configured to not use the internal RC oscillator (IRCOSC) and the system clock is programmed to a frequency higher than the IRCOSC.
3. Trigger the short functional reset with ESR0 as reset indicator. ESR0 itself is also configured as short reset.

The behavior is only seen on the external ESR0 pin of the MCU. The internal functional reset signal in the MCU does not see multiple assertions.

Workaround:

There are several ways that the multiple resets can be avoided:

- Do not use the PCS mechanism or
- Do not use the short functional reset feature

  or
- If using the short functional reset, do not configure ESR0 as reset indicator

Alternatively, if conditions 1, 2 and 3 in the Errata Description are present, then ignore multiple external pin assertions of ESR0.

1.245 ERR006904: DSPI: Reads the RXFRx causes failures of subsequent DSPI register reads

Description:

When any Deserial Serial Peripheral Interface (DSPI) Receive FIFO register (RXFRx) is read, the subsequent reads from any other registers return incorrect values.

Workaround:

Do not read any of the Receive FIFO registers (RXFRx). Received data should always be read from the POP FIFO register (POPR).

1.246 ERR006905: DSPI: When Extended SPI Mode is used to transmit frames of size > 16 bits, outgoing frames can be corrupted.

Description:

In the Deserial Serial Peripheral Interface (DSPI), when Extended SPI (XSPI) mode is used and frames with size greater than 16 bits are transmitted, it can lead to corruption of the Transmit First-In- First-Out (TXFIFO) read pointer, thus corrupting subsequent outgoing data transmissions with wrong data.

Note: Extended SPI mode allows the user to send up to 32-bit SPI frames. Command Cycling is also enabled which allows the user to send mutiple Data Frames using a single Command Frame.

Workaround:

The workaround depends on the FIFO_DEPTH definition of the device. Most devices implement a depth of 4 (EVEN).

If the TX FIFO Depth is EVEN:

a) Always sending frames with size < 17 bits does not cause any issue.

b) To continuously send frames with size > 16 bits, initially send a dummy frame (with no chip select) having size < 17. Do not clear the FIFO by asserting MCR[CLR_TXF] after this. Now frames with with size > 16 may be sent. Again, do not mix frames having size < 17 with these frames.

c) In order to switch between sending frames with size > 16 and those with size < 17, always perform a CLEAR FIFO operation (by setting MCR[CLR_TXF]) and then populate fresh data.

If the TX FIFO Depth is ODD:

a) Transmitting only frames with size < 17 bits causes no issues.

b) Transmitting only frames with size > 16 bits causes no issues.

c) Data transmissions must fall under category (a) or (b) and should not be mixed. In order to switch categories, perform a CLEAR FIFO operation (by setting Mode Control Register Clear FIFO (MCR[CLR_TXF])) and then populate fresh data.

1.247 ERR006906: 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.

1.248 ERR006908: I2C: Debug entry status bit is not set until frame transfer is complete

Description:

When the Inter-Integrated Circuit (I2C) module is transmitting multiple frames (in master or slave mode) and debug mode is entered in the middle of the on-going transmission, the debug entry status IBDBG[IPG_DEBUG_HALTED] bit in the I2C Bus Debug Register will only be set on the start of the next frame. In case of a single frame transmission, the IBDBG[IPG_DEBUG_HALTED] bit in the I2C Bus Debug Register will be set at the end of the frame.

Workaround:

Check the I2C Bus Busy Flag IBSR[IBB] to check if a transmission is on-going and enter debug mode only when it is set to '0'.

1.249 ERR006909: I2C: DMA transfers fail to resume after exiting debug mode

Description:

When the Inter-Integrated Circuit (I2C) module is in the process of transmitting or receiving data from a Direct Memory Access (DMA) and the user tries to enter debug mode throught I2C Bus Debug Register's Debug Enable bit (IBDBG[IPG_DEBUG_EN] = 1), the I2C module successfully enters the debug mode after transmitting or receiving the current data byte. When exiting the debug mode (IBDBG[IPG_DEBUG_EN] = 0), the I2C module fails to resume the transmission or reception of the rest of the data bytes.

Workaround:

Only enter the debug mode (IBDBG[IPG_DEBUG_EN] = 1) when DMA requests from I2C are not enabled by the I2C Bus Control Register's DMA enable bit (IBCR[DMAEN]=0).

1.250 ERR006915: LINFlexD: Erroneous receiver interrupt generation in UART FIFO mode

Description:

In the Local Interconnect Network Interrupt Enable Register (LINIER), the Data Reception Interrupt enable bit (LINIER[DRIE]) and Data Transmission Interrupt enable bit (LINIER[DTIE]) are significant only in LIN mode and Universal Asynchronous Receiver/Transmitter (UART) buffer mode.

Enabling these bits in UART First-In-First-Out (FIFO) mode will lead to an erroneous receiver interrupt being generated when the Receiver FIFO empty flag (UARTSR[RFE]) or Tx FIFO full flag (UARTSR[TFF]) in the UART Mode Status register are set.

Workaround:

Do not enable LINIER[DRIE] and LINIER[DTIE] bits in UART FIFO mode as these bits have functional significance only in LIN mode or UART buffer mode.

1.251 ERR006916: M_CAN: Rx FIFO overwrite mode, transmit pause and CAN FD 64-byte frames not supported

Description:

The new features included in Controller Area Network (M_CAN) Revision 3, such as Receive First-In- First-Out (Rx FIFO) overwrite mode, transmit pause, and support of CAN Flexible Data Rate (FD) 64- byte frames are not available.

As a result:

a) When the Rx FIFO is full, no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented,

b) Such cases are not supported where one ECU sends a burst of CAN messages that cause another node's CAN messages to be delayed because that nodes other messages have a lower CAN arbitration priority, and

c) CAN FD message length is limited to 8 bytes

Note: Setting the Rx FIFOn Configuration FIFOn Operation Mode bit (RXFnC[FnOM] = 1) enables RxFIFO overwrite mode. Setting the CAN Core (CC) Control Register Transmit Pause bit (CCCR[TXP] = 1) enables the transmit pause feature.

Workaround:

Do not use Rx FIFO overwrite mode, transmit pause or CAN FD 64-byte frames. Do not write software accessing RXFnC[FnOM], CCCR[TXP], and registers supporting CAN FD 64-byte frames.

CAN FD extended frame size support is shown below. Do not use these registers in this revision of the device:

Register/Bitfield	Description	Notes
RXESC	Rx Buffer / FIFO Element Size Configuration	
TXESC	Tx Buffer Element Size Configuration	
R1.DLC	Rx Buffer and FIFO Element R1 Data Length Code	(1)
Rn.DBm	Rx Buffer and FIFO Element Rn (n = 3...17) Data Byte m (m = 8...63)	
T1.DLC	Tx Buffer Element T1 Data Length Code	(1)
Tn.DBm	Tx Buffer Element Tn (n = 3...17) Data Byte m (m = 8...63)	
E1.DLC	Tx Event FIFO Element E1 Data Length Code	(1)
CCCR.CME	CAN Core Control Register CAN Mode Enable	
CCCR.CMR	CAN Core Control Register CAN Mode Request	

1. Do not use CAN FD frame size of 12/16/20/24/32/48/64 data bytes as specified below:
   0-8= CAN + CAN FD: received frame has 0-8 data bytes
   9-15= CAN: received frame has 8 data bytes
   9-15= CAN FD: received frame has 12/16/20/24/32/48/64 data bytes

1.252 ERR006932: NAR: Nexus timestamps are not implemented for the Nexus clients in the e200zx cores

Description:

The Nexus clients in the e200zx cores do not implement timestamp packets on the Nexus trace messages.

Workaround:

Do not expect timestamps on the Nexus messages from the core, even though timestamps are enabled in the device.

1.253 ERR006967: eDMA: Possible misbehavior of a preempted channel when using continuous link mode

Description:

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

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

Workaround:

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

1.254 ERR006990: 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.

1.255 ERR007041: PSI5: DMA SMC Frame Register (DSFR) updated independently from DMA_EN_SF

Description:

The Peripheral Sensor Interface (PSI5) module incorrectly updates the DMA SMC Frame register (PSI5_DSFR) as soon as data is written in the SMC Frame Register 1 (PSI5_SFR1) even when the DMA mode is disabled for SMC frames (bit DMA_EN_SF = 0 in the DMA Control register (PSI5_DCR)).

As a consequence an erroneous DMA transfer will occur as soon as the DMA mode is enabled.

Workaround:

Clear the PSI5_DSFR registers before enabling the DMA mode (by setting the bit PSI5_DCR[DMA_EN_SF]).

1.256 ERR007046: PMC: ADC power supply (VDD_HV_ADV) must be shorted with to VDD_HV_IO

Description:

The power supply for the analog to digital converters (VDD_HV_ADV) must be powered at the same time as the supply for the Input/output pins (VDD_HV_IO).

Workaround:

Ensure that VDD_HV_ADV supply is shorted with VDD_HV_IO.

1.257 ERR007049: SARADC: wrong behavior when aborting the second-last conversion of a chain

Description:

When using the Successive Approximation Analog to Digital Converter (SARADC), If during a chained conversion, the ABORT (Abort Conversion) bit is set when the second-to-last conversion evaluation phase is ongoing, then no data for the next channel (last of the chain) is returned and no valid bit is set. In addition, the NECH (End of normal chain conversion interrupt flag) does not get set and no interrupt is triggered.

For instance, assuming a normal chain conversion of channel x (CHx), channel y, (CHy) and channel z (CHz), if the ABORT bit is set during the evaluation phase of CHy, then the conversion of CHz is also aborted.

Workaround:

Do not abort a chain conversion when the second last channel execution is ongoing. The ongoing conversion can be checked by reading the CHADDR (Channel under measure address) field of the MSR (Main Status Register).

In the example above, the ABORT bit shall not be set if CHADDR = y.

1.258 ERR007051: SARADC: Peripheral bridge clock requires minimum clock ratio with respect to SARADC clock

Description:

If the peripheral bridge clock frequency (PBRIDGEx_CLK) is less than twice the frequency of the Successive Approximation Register Analog to Digital Converter (SARADC) clock (SAR_CLK), the VALID (Data valid flag) bit and the EOC_CH[x] (End of conversion interrupt pending bit for channel x) bit of the following registers of the first channel in a chain may randomly not be set during chained conversions.

Bit name	Register	Register description
VALID (Data valid flag)	ICDRn (n=0...95)	Internal Channel Data Registers
VALID (Data valid flag)	TCDRn (n=96...127)	Test Channel Data Register
VALID (Data valid flag)	ECDRn (n=128...255)	External Channel Data Registers
EOC_CH[x] (End of conversion interrupt pending bit for channel x)	ICIPRn (n=0...2)	Internal Channel Interrupt Pending Registers
EOC_CH[x] (End of conversion interrupt pending bit for channel x)	TCIPR	Test Channel Interrupt Pending Register
EOC_CH[x] (End of conversion interrupt pending bit for channel x)	ECIPRn (n=0...3)	External Channel Interrupt Pending Register

Workaround:

Keep the PBRIDGEx_CLK frequency more than 2 times the SAR_CLK frequency when possible.

If the peripheral bridge clock frequency must be lower than twice the SARADC clock frequency, then first channel of a chain should be used as a dummy channel and its results should be ignored.

1.259 ERR007053: M_CAN: Accesses to disabled M_(TT)_CAN modules causes device to hang

Description:

If the Register Bus Abort Enable bit (SSCM_ERROR[RAE]) in the System Status and Control Module Error register is set to 0, and any of the CAN (Controller Area Network) or CAN RAM modules are disabled, any internal bus accesses to a disabled module will cause the MCU to hang.

Workaround:

Do not access the CAN or CAN RAM modules when the corresponding module is disabled and SSCM_ERROR[RAE] = 0.

1.260 ERR007055: CGM : Divider field of AC0_DC3 register is 8-bit wide instead of 9

Description:

In the Clock Generation Module (CGM) the Divider division value field of the Auxiliary Clock 0 Divider 3 Configuration Register (CGM_AC0_DC3[DIV]) is 8 bits wide instead of 9.

No error is generated when writing a 9-bit value: the most significant bit is ignored and any value between 0x100 and 0x1FF is seen as from 0x00 to 0xFF.

Workaround:

Do not use division factor above 0xFF.

The CGM_AC0_DC3 is configuring the clock for the DSPI4 and DSPI5. Take into account the input clock limitation when computing the DSPIx baudrate.

1.261 ERR007057: SIUL2: Incorrect MSCR reset value for pins PA[9:4], PB[11] and PC[2]

Description:

After reset, the following configurations are applied to the pads listed below:

- PA[4]: general purpose input, weak pull-up enabled
- PA[5]: input buffer and weak pull-down enabled, JTAG function selected (MSCR5[SSS]=5)
- PA[6]: input buffer and weak pull-down enabled, JTAG function selected (MSCR6[SSS]=5)
- PA[7]: input buffer and weak pull-up enabled, JTAG function selected (MSCR7[SSS]=5)
- PA[8]: input buffer and weak pull-up enabled, JTAG function selected (MSCR8[SSS]=5)
- PA[9]: output high impedance, JTAG function selected (MSCR9[SSS]=5)
- PB[11]: output high impedance, FCCU function selected (MSCR27[SSS]=5)
- PC[2]: input buffer and weak pull-up enabled, FCCU function selected (MSCR34[SSS]=5)

where MSCRx is the Multiplexed Signal Configuration Register x in the System Integration Unit Lite2 (SIUL2).

Workaround:

Take into account the reset configuration of the listed pads.

1.262 ERR007058: STCU2: Nexus interface of Peripheral Core_2 is not reset after an LBIST execution

Description:

After completion of an off-line logic built-in self-test (LBIST) via the Self Test Control Unit (STCU2), the Nexus interface of the Peripheral Core_2 is not part of the reset module and may remain in an unknown state.

This will have no effect when debug interface is not used and the associated JCOMP input of the JTAG interface is tied low. It may instead prevent correct code execution in case debug session is active.

Workaround:

When debugging the MCU, reset the Nexus interface after an off-line self-test completion via JCOMP reset input: toggle JCOMP pin from high to low and then back to high.

1.263 ERR007059: CJTAG: cJTAG maximum frequency is 2MHz

Description:

The IEEE 1149.7 Compact JTAG Test Access Port Controller (CJTAG) clock is limited to 2MHz, reducing the bandwidth of the debug operations.

Workaround:

Limit the clock frequency of the debug link to 2MHz in cJTAG mode.

When higher debug bandwidth is needed, consider using the 5 pin JTAG interface mode.

1.264 ERR007061: SPU: Reserved location can be written

Description:

Read accesses to reserved locations of the Sequence Processing Unit (SPU) do not return 0. Instead the last written value will be read.

Note: the SPU registers are accessible only via JTAG and are used by debugging tools to create triggers combining internal events of the device.

Workaround:

Do not access to reserved locations of the SPU. Alternatively, always write 0 to the reserved locations.

1.265 ERR007062: STCU2: Hardware Abort condition may not be correctly reported by the ABORTHW flag

Description:

The On-line hardware abort flag (ABORTHW) bit of the Self-Test Control Unit (STCU2) Error Register (STCU_ERR_STAT) is expected to be set following an hardware abort request of an on-line self-test. Nevertheless, when an hardware abort is triggered to the STCU2 as the consequence of a functional reset (which requires to stop the Self-Test execution), depending on the timing of the hardware abort fired to the STCU2:

- The self-test execution is correctly stopped
- The STCU_ERR_STAT[ABORTHW] flag may not be set, preventing the user to identify the event.

Workaround:

Do not rely on the ABORTHW flag.

1.266 ERR007063: STCU2: Off-Line LBIST execution may generate a reset

Description:

When the Self-Test Control Unit (STCU2) is executing an Off-Line LBIST (Logic Built-In Self-Test) sequence, an unwanted reset is triggered by the test of partition 2.

Workaround:

To avoid an unwanted reset, the LBIST partitions 1 and 2 need to be run in parallel and run at the end of the Off-Line Self-Test sequence.

1.267 ERR007064: FCCU: The Error signals EOUT[0]/EOUT[1] cannot be masked by error source in dual rail mode

Description:

The Fault Collection and Control Unit (FCCU) supports different error communication protocols to report the occurence of an error.

User can enable or disable (mask) the error signaling for each source of error connected to the FCCU. This configuration is done via the FCCU EOUT Signaling Enable Registers (FCCU_EOUT_SIG_ENx, x=0 to 3).

When the dual rail protocol is selected the content of the FCCU_EOUT_SIG_ENx registers is ignored. Therefore the error signaling is activated for any source of error connected to the FCCU.

Workaround:

Don't use the dual rail protocol if individual error source masking is needed.

1.268 ERR007066: PASS: Hardware Interlock for BAF not implemented

Description:

In the Password and device security module (PASS), the hardware interlock feature is not implemented.

As a consequence, setting the Hardware Interlock bit (HI) of the Production Disable DCF client has no effect. The programming of the BAF block is controlled only by the Utest NVM Lock bit (TSLOCK) of the Password Group n - Lock 0 Status register (LOCK0_PGn).

Workaround:

Rely only on the TSLOCK bit of LOCK0_PGn registers to control the programming of the BAF block.

1.269 ERR007067: IRCOSC: Reduced accuracy of the software trimming

Description:

The software trimming accuracy of the Internal RC Oscillator digital (IRCOSC) is within -1%/+1% range.

Workaround:

Take into account that the reduced accuracy of the IRCOSC software trimming.

1.270 ERR007068: STCU: FlexRay corrupted after the online execution of the MBIST

Description:

The execution of the online Memory Built-In Self Test (MBIST) may cause a very short, temporary loss of the FlexRay clock causing the FlexRay module to enter an unexpected state.

Workaround:

The FlexRay must be reset and then re-initialized after the execution of the online MBIST. FlexRay can be individually reset via the FlexRay_x_RST bit in the corresponding Peripheral Reset register (RGM_PRSTy) of the Reset Generation module (RGM).

1.271 ERR007083: GTM: The DPLL's SORI, TORI, MTI, and MSI interrupts may not be asserted

Description:

(GTM-IP-161)

The Generic Timer Module (GTM) Digital PLL (DPLL) Missing Trigger(MTI), Missing State (MSI), Trigger Out Of Range (TORI) and State Out of Range (SORI) interrupts are not set (1) when the 3 following conditions are met:

- The upper 24 bits of Timebase Timestamp Channel 0 (TBU_TS0) are used as the input (DPLL_STATUS[LOW_RES]=1)
- The trigger/state input time stamps have an 8 times higher resolution than TBU_TS0 (DPLL_CTRL1[TS0_HRT/S] = 1)
- The upper three bits of TBU_TS0 are not equal to "000".

The DPLL Lock Status bits for SUBINC1 and SUBINC2 (LOCK1/2) and the MTI/MSI flags in the DPLL_STATUS register are incorrect if the above case occurs.

Workaround:

Do not use the configuration DPLL_STATUS[LOW_RES]=1 with DPLL_CTRL1[TS0_HRT/S] = 1.

1.272 ERR007084: GTM: An active edge input, that is rejected by the DPLL trigger plausibility check, does not assert a Missing Trigger Interrupt

Description:

(GTM-IP-162)

The Generic Timer Module (GTM) Digital PLL (DPLL) Missing Trigger Interrupt (MTI) is not asserted during a gap in the trigger profile, stored in RAM region 2c, when an active input signal edge is rejected by the Plausibility Value of Trigger (PVT) check.

In this case, the DPLL's internal check for MTI is performed when the invalid active input occurs. The check for the first and valid active inputs is not done for this gap. As a result, when monitoring the DPLL synchronization using the MTI in a gap, the application may report a synchronization problem which is not real.

Workaround:

The activated PVT check is reported by the activation of the Plausibility Window Violation of TRIGGER (PWI) interrupt. This interrupt can be used to check if a gap condition in the profile has occurred. This information can be used to correct the incorrect synchronization information out of the DPLL.

1.273 ERR007085: GTM: A TIM timeout occurs when the TDU is re-enabled

Description:

(GTM-IP-163)

The Generic Timer Module (GTM) Timer Input Module (TIM) Timeout Detection Unit (TDU) indicates a timeout event when it is re-enabled and the new TDU Timeout Value (TOV) is lower than the previous TOV.

After stopping the TDU, the Time Out Counter (TO_CNT) field will have an arbitrary value

less than or equal to (<=) the timeout value. When TOV is reconfigured to a value less than or equal to (<=) TO_CNT, and the TDU re-enabled an incorrect timeout is signaled. This is because if at the same time as the TDU is enabled the selected clock has an active edge, TO_CNT is greater than or equal to (>=) TOV because TO_CNT is not reset to zero.

Workaround:

If the TDU needs to be changed to a TOV value which is less than the previous value either:

1. Postpone disabling the TDU until the TO_CNT is less than (<) the new TOV. Then disable the TDU, configure TOV, and re-enable TDU Unit.
2. Disable the TDU and then write TOV with the value 0xFF. Then enable the TDU unit and re- configure TOV to the desired value.

1.274 ERR007086: GTM: TIM PWM and PIM modes may capture the wrong timestamp

Description:

(GTM-IP-164)

When the Generic Timer Module (GTM) Timer Input Module (TIM) channel is configured with Counter Select (CNT_SEL) of the Channel Control Register (CTRL) set, and an input edge is detected by that channel before a rising edge on the clock, the Channel Counter Shadow Register (CNTS) will capture the value of Channel Count Register (CNT) instead of the timebase (TBU_TS0) in PWM measurement mode (TPWM) and pulse integration mode (TPIM).

Workaround:

To avoid incorrect timestamp captures in the TIM PWM and PIM mode, the follow steps must be taken:

1. Select a TIM clock source which is identical to SYS_CLK.
2. Use the input event mode (TIEM) to capture TBU_TS0 for rising and falling input edges.
3. In TPWM mode, use CNT register as input (CNTS_SEL=0) with CMU_CLK source selected. With TBU_TS0 selected as the input to General Purpose Register 0 (GPR0), and with CNT selected as the input to General Purpose Register 1 (GPR1), calculate the correct timestamp: GPR0 - GPR1 + CNTS.

1.275 ERR007087: GTM: The DPLL's Address Pointer Extension value is added to the Address Pointer when the Address Pointer Status bit is 0

Description:

(GTM-IP-166)

If the Generic Timer Module (GTM) Digital PLL (DPLL) Address Pointer Extension field (APT_2b_EXT/APS_1c2_EXT) in the Address Pointer Trigger/State Synchronization Register (DPLL_APT_SYNC) is not zero during synchronization, it is added to the Address Pointer for Trigger/State (APT_2b/APS_1c2) in DPLL_APT/DPLL_APS registers regardless of the state of the status bits (APT_2b_STATUS/APS_1c2_STATUS) in DPLL_APT_SYNC.

Workaround:

If the pointers should remain unchanged after synchronization, APT_2b_EXT/APS_1c2_EXT must be set to zero before the synchronization is performed.

1.276 ERR007088: GTM: When ATOM is in SOMP mode the SR0/SR1 registers could be updated twice in one PWM period

Description:

(GTM-IP-167)

When the Generic Timer Module (GTM) Advanced Routing Unit (ARU) Connected Timer Output Module (ATOM) is in Signal Output Mode PWM (SOMP) mode with the ARU enabled, and the channel is configured to be updated by the preceding channel (ARU_EN=1 and RST_CCU0=1 in ATOM[i]_CH[x]_CTRL), an update of the channel's Shadow Registers (ATOM[i]_CH[x]_SR0/SR1) via ARU is requested when Counter 0 (ATOM[i]_CH[x]_CN0) reaches Compare 0 (ATOM[i]_CH[x]_CM0).

In this case, if CN0 reaches CM0, CN0 is not reset and continues counting until it is reset by the trigger of the preceding channel. As a result, the ATOM channel updates the SR0/SR1 registers after the update of CM0/CM1, which is not synchronous with the counter reset. Depending on the time between CM0 and the value of CN0 when it was reset by the trigger, the SR0/SR1 registers may be updated twice in one period.

Workaround:

If new data via ARU is provided by Multi Channel Sequencer (MCS) ensure, through software, that only one value per period of new data for SR0/SR1 register can be read. For example, this can be achieved by starting a 'master period' which triggers the reset of CN0 on a time base value and provides the start value and period to the MCS. The MCS can then calculate a minimum time it must wait before providing new ARU data.

If the above workaround is unsuitable (for example if new data via the ARU is provided by the FIFO) do not use SOMP mode with ARU_EN=1 and RST_CCU0=1.

1.277 ERR007099: 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].

1.278 ERR007103: MC_CGM: Incorrect cause for the latest clock source switch may be reported by the CGM if a safe mode request arrives when the system clock is the IRC

Description:

If the current system clock source is the Internal RC oscillator (IRC) as reported in the Clock Generation Module System Clock Select Status Register System Clock Source Selection field (CGM_SC_SS.SELSTAT = 0b0000) and the Clock Generation Module System Clock Select Status Register Switch Trigger Cause shows the cause for the latest clock switch as MC_ME succeeded (CGM_SC_SS.SWTRG = 0b001) indicating that a successful Mode Entry mode change was the cause of the last clock change then the CGM_SC_SS.SWTRG will incorrectly continue to show the cause for the latest clock switch as MC_ME succeeded after a safe mode request is generated. If a subsequent safe mode request is generated CGM_SC_SS.SWTRG switches to report the correct status value of 0b100 (switch to system clock source 0 due to SAFE mode request or reset succeeded).

Workaround:

If the CGM_SC_SS.SELSTAT shows the system clock as IRC (0b0000), then software should check the Mode Entry Global Status register Current Mode field (ME_GS.CURRENT_MODE) and the Mode Entry Interrupt Status Register Safe mode Interrupt (ME_IS.I_SAFE) to establish the cause of the switch.

1.279 ERR007108: SARADC: Spikes on external multiplexer signals

Description:

When a Successive Approximation Analog to Digital Converter (SARADC) conversion is triggered in injected mode for an external multiplexer channel, with or without a normal conversion running in background, a spike of two ADC clock cycles duration appears on the 3-bit external multiplexer address decode select lines. This spike is generated in the first cycle of the evaluation phase after the sampling phase is over for the injected channel.

Workaround:

There are 3 possible workarounds.

a) Mask the external multiplexer address select lines by programming SIUL2 (System Integration Unit Lite 2) MSCR (I/O Pin Multiplexed Signal Configuration Registers) when not performing external channel conversions.

b) Do not perform injected mode conversion on external channels.

c) Do not used an external multiplexer to expand the number of ADC channels.

1.280 ERR007109: I2C: In master receive mode, data remains latched in I2C data I/O register (IBDR) until new data is received

Description:

When the Inter-Integrated Circuit (I2C) is configured in master mode and receiving data from a slave which is transmitting data bytes on an irregular basis, there is no way for the master to know if the data received in the I2C data Input/Output register (IBDR) is the old latched data or the new data received from the slave.

Workaround:

When slave is configured to transmit data on an irregular basis, in other words intermittently, it should not send 2 consecutive bytes with the same data. When 2 consecutive data bytes are different, a dummy read of the I2C data I/O register (IBDR) can be initiated before the actual read. These 2 bytes can be compared to know if it is the new data or the old data.

1.281 ERR007115: DSPI: Mixing 16 and 32 bits frame size in XSPI Mode can cause incorrect data to be transmitted

Description:

The Deserial Serial Peripheral Interface (DSPI) features an Extended SPI mode (XSPI) supporting frames of up to 32 bits.

When the XSPI Mode is enabled, transferring a mixture of frames having a size up to 16 bits and those having size above 16 bits can cause an incorrect data transmission to occur. This happens when the First In/First Out (FIFO) queue read pointers roll-over and a frame needs to be extracted from both the bottom of the FIFO and the top of the FIFO when the Frame Size is greater than 16 bits.

Workaround:

Even number of Transmit FIFO Register (TXFR) registers:

Do not mix frames that have data sizes of less than 16 bits with those having a size more than 16 bits in XSPI Mode.

Odd number of TXFR registers :

Do not mix frames that have data sizes of less than 16 bits with those having a size more than 16 bits in XSPI Mode.

If the frame size is greater than 16, initially send a dummy frame (a frame with no chip select, but containing data) of less than or equal to 16 bits. Continue sending a dummy frame after each (number of TXFR Registers - 1) / 2 frames.

1.282 ERR007116: 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.

1.283 ERR007126: MEMU: Instead of Byte 1 of MEMU CTRL Register, Byte 3 is currently protected

Description:

The Memory Error Management Unit (MEMU) Control (CTRL) register has Byte 3 protected instead of Byte 1 (using Byte numbering 3 to 0). Therefore the Software Reset bit (SWR) does not have register protection features.

Workaround:

Write software such that it does not rely on register protection features for MEMU CTRL Byte 1 SWR. Note that if the SWR bit is inadvertently set, only the status flags and overflow register will be cleared. The reporting tables are not cleared when SWR is set.

1.284 ERR007129: FLASH: Incorrect signature of Array Integrity Check or Margin Read operations if both breakpoint and suspend features are used

Description:

In the Flash module, the Array Integrity Check and Margin Read operations can be configured to stop when a single bit correction or a double bit detection occurs. This is called breakpoint and is enabled by setting the Array Integrity Break Point Enable bit (AIBPE) of the User Test 0 register (UT0).

User can also pause the array integrity and margin read operations by setting the Array Integrity Suspend bit (AISUS) of UT0.

Suspending the operation while the breakpoint function is enabled leads to the corruption of the signature result readable in the User Multiple Input Signature 0-9 registers (UM0 to UM9).

Note that both the breakpoint functionality and the suspend functionality are fully operational when used separately.

Workaround:

Do not issue a Flash Array Integrity Suspend request (UT0[AISUS]=1) when breakpoint is enabled (UT0[AIBPE]=1).

In case breakpoint is enabled and user must interrupt the operation, issue an Abort request by clearing the Array Integrity Enable bit (AIE) of UT0, then restart the entire operation.

1.285 ERR007130: FLASH: Incorrect termination of program sequence when the Program bit is cleared before the Enable High Voltage bit is set

Description:

In the Flash module, an incorrect address could be programmed after a cancelled programming attempt, even if a program operation is properly started by executing the following steps in this order:

1. Set the Program bit (PGM) in the Module Configuration Register (MCR): MCR[PGM]=1
2. Issue an interlock write (flash module write that defines first address to be programmed with the programmed data)
3. Set the Enable High Voltage bit (EHV) in the MCR: MCR[EHV]=1

Once the interlock write occurs, the address is internally latched and is not cleared even if the MCR[PGM] is cleared in order to cancel the operation.

As a consequence the next program or erase operation will occur at the address of the cancelled operation even if the 3 steps have been correctly followed.

In addition the Low/Mid Address Space Block Locking register (LOCK0), High/Data Address Space Block Locking register (LOCK1) and 256K Address Space Block Locking register (LOCK2) remain unwritable.

Workaround:

If a program operation must be terminated and the interlock write after setting MCR[PGM] has occurred, it is mandatory to set then clear MCR[EHV] resulting in a program abort sequence.

A program abort sequence can occur without limitations. MCR[EHV] clear will occur within 10us from being set without unwanted data being programmed.

1.286 ERR007131: FLASH: Incorrect termination of erase sequence when Erase bit is cleared before Enable High Voltage bit is set

Description:

In the Flash module, an incorrect address could be programmed after a cancelled erase attempt, even if an erase operation is properly started by executing the following steps in this order:

1. Set the Erase bit (ERS) in the Module Configuration Register (MCR): MCR[ERS]=1
2. Issue an interlock write (flash module write that defines first address to be programmed with the programmed data)
3. Set the Enable High Voltage bit (EHV) in the MCR: MCR[EHV]=1

Once the interlock write occurs, the address is internally latched and is not cleared even if MCR[ERS] is cleared (before doing step 3) in order to cancel the operation.

As a consequence the next program or erase operation will occur at the address of the cancelled operation even if the 3 steps have been correctly followed.

In addition the Low/Mid Address Space Block Locking register (LOCK0), High/Data Address Space Block Locking register (LOCK1) and 256K Address Space Block Locking register (LOCK2) remain unwritable.

Workaround:

If an erase sequence must be terminated and the interlock write after MCR[ERS] setting has occurred, it is mandatory to set then clear MCR[EHV] resulting in an erase abort sequence.

Erase abort sequence can occur without limitations. If MCR[EHV] is cleared within 10us from being set, the actual erase process is not started.

1.287 ERR007134: RCCU: If any accesses to the I-MEM or D-MEM of the safety and checker core are performed while the cores are disabled, the cores will get out of lockstep when enabled

Description:

If any accesses to the local Instruction Memory (I-MEM) or Data Memory (D-MEM) of the safety (Main Core_0) and checker core (Checker Core_0s) are performed while the cores are disabled, the cores will get out of lockstep when enabled. A disablement of Main Core_0 and Checker Core_0s is defined to be when the corresponding status bits of the Core Status Register (ME_CS) are set to 0. Most typically, this will happen after a destructive RESET where the system is booted up with the I/O Processor (IOP) and Hardware Security Module (HSM) enabled, but the safety and checker cores are disabled.

Memory Built In Self Test (MBIST) and Logic Built In Self Test (LBIST) are not affected by this errata.

Workaround:

Do not read or write the I-MEM or D-MEM of the safety core while it is disabled. Insure that the Main Core_0 and Checker Core_0s are enabled and executing out of some memory (I-MEM / D-MEM, system RAM, Flash, ROM, external memory) when any bus master is accessing the safety core I- MEM / D-MEM.

1.288 ERR007137: MEMU: incorrect indication when a correctable error is signaled by the e200zx core cache

Description:

The Memory Error Management Unit (MEMU) incorrectly reports a non correctable error in case of a double bit error in the data array of the instruction or data cache of the e200zx core.

There are two types of errors in the e200zx core cache that the MEMU should detect as a correctable error:

1. Single bit error and not an address error
2. Double bit error in data array and not an address error

In this revision of the device, the MEMU can only detect the first type as a correctable error. When an error is a double error in data array, MEMU is not able to report it as correctable error.

Note: For the second type, the expected behavior for cache double bit error in the data array is the e200zx core logic detects the error and invalidates the cache line. This is defined as a correctable error event and should be reported as such to MEMU in normal operation.

Workaround:

Do not expect the MEMU to contain a correctable error report for a double bit error in the cache data array. Use the e200zx cache correction / auto-invalidation and cache line lockout features to achieve desired cache memory error management. Correction, auto-invalidation, and lockout features are configured in the L1 Cache Control and Status Registers (L1CSR[0,1]).

1.289 ERR007138: SARADC: Missed conversion after ABORT of the last channel of an injected chain

Description:

In the Successive Approximation Register Analog-to-Digital Converter (SARADC), when a chain conversion is injected over a normal chain conversion and an abort conversion command is initiated by setting the abort conversion bit of the Main Configuration Register (SARADC.MCR[ABORT]), when the last channel of the injected chain is in the sampling phase then a conversion will be missed. The conversion of the next normal channel after the resume is skipped.

Expected behavior: the conversion of the last injected channel is aborted, and the normal chain is resumed.

Errata behavior: the conversion of the last injected channel is correctly aborted, but the conversion of the next normal channel is incorrectly aborted.

For example: If channels 0, 1, 2, 3, 4, and 5 are to be converted in Normal chain and channels 6, 7, 8, and 9 are injected when the channel 3 conversion was ongoing the following behavior occurs: nch0 --> nch1 --> nch2 --> nch3 (aborted by injected conversion chain) --> jch6 --> jch7 --> jch8 --> jch9 (MCR[ABORT] set at this time) --> nch3 (restarted after end of injected chain) [(a)] --> nch5

Workaround:

Do not issue a conversion abort request when the conversion of the last channel of an injected chain is on going.

The user can read the Channel under measure address field (CHADDR) of the SARADC Main Status register (SARADC_MSR) to identify the channel under conversion.

a. The conversion of the normal channel 4 is missing.

1.290 ERR007140: SAFETY: Not all spatial separation measures against CCFs implemented

Description:

Not all spatial separation measures against Common Cause Failures (CCF) of replicated channels have been implemented. Redundant I/O blocks have not been physically separated. Logic in the main core_0 shares a common well with logic in the checker core. Routing from the main core_0 is not physically separated from routing in the checker core.

Workaround:

Take these items into account in the safety analysis. This revision is not intended for series production of a safety-relevant application.

1.291 ERR007190: GTM: Simultaneous Core and DPLL accesses to RAM Region 2 may lead to the DPLL reading erroneous data

Description:

(GTM-IP-168)

A core or DMA access to the Generic Timer Module (GTM) RAM Region 2 at the same time as a Digital PLL (DPLL) accesses that memory may result in the DPLL reading erroneous data. As a result, calculations of the DPLL may be wrong and this may lead to loss of synchronization.

For example, if the DPLL accesses RAM Region 2 to read a value from the profile (for calculation of TRIGGER) and at the same time the core or DMA has initiated a second read/write operation of RAM Region 2 via the Automotive Electronics Interface (AEI), it is possible that the output data of the core or DMA RAM read/write request is used as read data for the DPLL initiated read instead of the Number of Real and Virtual Events to be Considered for the Current or Last Increment (syn_t and syn_t_old) values.

Workaround:

Option 1: Synchronize core and DMA accesses to phases where the DPLL is not accessing RAM Region 2.

Example 1: synchronize DPLL RAM2 accesses to the TRIGGER signal using the TRIGGER Active Slope Interrupt (TASI) and checking that the RAM2 access is finished before the next active TRIGGER edge.

Example 2: use an Advanced Routing Unit (ARU) Connected Timer Output Module (ATOM) channel in Signal Output Mode PWM (SOMP) one shot mode to output a 200 SYS_CLK pulse when the DPLL calculation starts. Use a Timer Input Module (TIM) channel to generate an active edge interrupt based on the DPLL calculation synchronized ATOM output. With the ATOM's Counter Compare Unit 1 (CCU1) interrupt (200 SYS_CLKs later) the DPLL sub increment calculation should be complete. At start of the ATOM interrupt service routine

check if any action calculation is ongoing in the DPLL by reading the Calculation of Actions In Progress (CAIP2) flag in the DPLL_STATUS register. If this flag is zero (0), RAM Region 2 access by the core and DMA are safe to do.

Option 2: allow core and DMA accesses in tooth profile phases where DPLL is not accessing RAM2. For example: use a Multi-channel Sequencer (MCS) to calculate and set flags that indicate the non critical phases the tooth profile when the DPLL does not access RAM2.

1.292 ERR007191: GTM: The DPLL's SORI and TORI interrupts are not asserted

Description:

(GTM-IP-169)

The Generic Timer Module (GTM) Digital PLL (DPLL) Trigger Out Of Range (TORI) and State Out of Range (SORI) interrupts are not set (1) when the following conditions are met:

- The upper 24 bits of Timebase Timestamp Channel 0 (TBU_TS0) are used as the input (DPLL_STATUS[LOW_RES]=1)
- The trigger/state input time stamps have an 8 times higher resolution than TBU_TS0, and the estimated time point value is multiplied by 8 (DPLL_CTRL1[TS0_HRT/S] = 0)

Workaround:

In this configuration use a Timer Output Module (TOM) or ARU Connected TOM (ATOM) to generate an interrupt on the time out of TRIGGER/STATE.

With every TRIGGER/STATE edge, adapt the (A)TOM period to the current speed and reset Counter 0 (CN0). If CN0 is not reset by the next TRIGGER/STATE event, (A)TOM raises an edge interrupt at the end of the period.

1.293 ERR007194: MC_ME: IMTS bIt 26 is S_MRIG

Description:

Bit 26 (Most significant bit = 0) of the Mode Entry Invalid Transition Status register (MC_ME_IMTS) is the Mode Request Ignored Status bit (S_MRIG). This bit is set whenever a new mode is requested while a transition to the SAFE mode is in progress and the mode entry change has been ignored.

This bit is also set when if a transition to HALT0/STOP0 mode is requested while a wake-up event is active. This bit is cleared by writing 1 to the bit.

Workaround:

Keep in mind that IMTS[S_MRIG] could be set on ignored mode entry transitions. Software may need to clear this bit and perform additional steps to handle the cause of the Mode Entry transition failure to complete a valid mode entry transition.

1.294 ERR007196: MC_CGM: The address of the CGM_PCS_DIVS and CGM_PCS_DIVE registers for all clock sources are swapped

Description:

The address of the Progressive Clock Switching Divider Start (CGM_PCS_DIVSn) and PCS Divider End (CGM_PCS_DIVEn) registers for all clock sources may be swapped in the device documentation.

Register	Incorrect Address offset	Correct Address offset
CGM_PCS_DIVE1	0x70C	0x708
CGM_PCS_DIVS1	0x708	0x70C

Similarly, the addresses of all the CGM_PCS_DIVSn registers needs to be swapped with the addresses of CGM_PCS_DIVEn registers.

Workaround:

Swap the addresses of CGM_PCS_DIVEn registers with the addresses of CGM_PCS_DIVSn registers.

1.295 ERR007202: SENT: Increased tolerance to noise for Nibble length measurement is not available

Description:

The Single Edge Nibble Transmission (SENT) Receiver (SRX) supports the SAE J2716 (January 2010) standard that supports a maximum of 12.5% tolerance for the nibble length adjustment for noise and clock jitter. It does not support up to 50% that may be required for some applications.

Workaround:

Include extra noise filtering on the target board or use the per channel noise filter to increase tolerance to noise by programming the appropriate value in Input Filter Sample Count field of the Channel Configuration Register (CHn_CONFIG[FIL_CNT]). In addition, writing a 1 to the Nibble Length Variation Limit bit (GBL_CTRL[NIB_LEN_VAR_LIMIT]) of the SRX Global Control Register will not enable the optional 50% variation support.

1.296 ERR007203: SENT: In debug mode SENT message data registers appear to lose contents

Description:

The message read registers [Channel ‘n' Fast Message Data Read Register (n = 0 to (CH-1)) (CHn_FMSG_DATA), Channel ‘n' Fast Message CRC Read Register (n = 0 to (CH-1)) (CHn_FMSG_CRC), Channel ‘n’ Fast Message Time Stamp Read Register (n = 0 to (CH-1)) (CHn_FMSG_TS), Channel ‘n’ Serial Message Read Register (Bit 3) (n = 0 to (CH-1)) (CHn_SMSG_BIT3), Channel ‘n’ Serial Message Read Register (Bit 2) (n = 0 to (CH-1))

(CHn_SMSG_BIT2), Channel ‘n’ Serial Message Time Stamp Read Register (n = 0 to (CH-1)) (CHn_SMSG_TS), DMA Fast Message Data Read Register (DMA_FMSG_DATA), DMA Fast Message CRC Read Register (DMA_FMSG_CRC), DMA Fast Message Time Stamp Read Register (DMA_FMSG_TS), DMA Slow Serial Message Bit3 Read Register (DMA_SMSG_BIT3), DMA Slow Serial Message Bit2 Read Register (DMA_SMSG_BIT2) and DMA Slow Serial Message Time Stamp Read Register (DMA_SMSG_TS)] will appear to lose their contents in the following conditions:

The very first message is being received but not yet completely received and the MCU enters debug or freeze mode, the current message reception will get discarded and message read registers (as mentioned above) will read zeros

Auto clear functionality is enabled (GBL_CTRL[FAST_CLR] = 1). In this case, when first message is read, it will get clear the message read registers (due to auto clear functionality being enabled). On reading again, the message read registers (as mentioned above) might read zeros.

Workaround:

If the MCU requests entry to debug or stop mode, the message being received currently by SENT Receiver is discarded and the MCU enters debug/stop mode immediately. This does not affect the messages received completely, prior to entering debug mode and these messages will still be present on the message buffer and registers until they are read out.

Thus allow one message to be received completely and do not enable "Auto Clear" (GBL_CTRL[FAST_CLR] = 0), to allow messages to be read in debug or stop mode.

1.297 ERR007204: 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: *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.

1.298 ERR007211: MC_ME: Core register IAC8 is cleared during a mode change when the core is reset

Description:

If a core is reset (ME_CADDR[0,1,2].RMC =1) in the Core Address register during a Mode Entry module (MC_ME) mode change then the Instruction Address Compare 8 (IAC8) register within the core which receives the reset will be cleared. In this implementation IAC8 is used as the Security watchdog service address. If a watchdog time-out occurs after this mode change and no valid service address exists, the core will attempt to execute code from the invalid address potentially resulting in an exception.

The watchdog (SWT) associated with that core is not reset by this change and retains its configuration. If fixed address execution is configured by the Service Mode in the software watchdog control register (SWT_CR.SMD= 0b10) when IAC8 is cleared to 0, it will not be possible to update IAC8 with the correct value. For other service modes the IAC8 register will be cleared to 0, but can be updated.

Workaround:

If the software watchdog mode is in fixed address execution (SWT_CR.SMD= 0b10), do not reset the corresponding core upon mode change. For all other modes, IAC8 must be updated by software immediately after the mode transition completes.

1.299 ERR007222: SARADC: Minimum value of precharge must be greater than or equal to 2 ADC clock cycles

Description:

The Successive Approximation Register Analog-to-Digital Converter (SARADC) requires a minimum valid value of the Precharging phase duration field (PRECHG) of the Conversion Timing Register (SARADC_x.CTRz[PRECHG]) must be 2, meaning the precharge phase duration is two (2) SARADC clock cycles. This is incorrectly defined as '1' in some revisions of the documentation.

Workaround:

Take into account the minimum value of 2 when configuring the precharge duration in the SARADC_x.CTRz[PRECHG].

1.300 ERR007223: 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."

1.301 ERR007227: 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. When a fault input is disabled in the FCCU and a fault occurs, the edge detection will be blocked until it gets initialized by a FCCU reaction or a power-on reset of the device.

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. 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.

1.302 ERR007232: FCCU: FCCU registers do not revert to default after configuration timeout

Description:

In the Fault Collection and Control Unit (FCCU), when a configuration timeout occurs, the following registers keep the last programmed value instead of reverting to the reset default value:

- FCCU Delta T Register (FCCU_DELTA_T)
- FCCU IRQ Alarm Enable Registers (FCCU_IRQ_ALARM_EN[0:3])
- FCCU NMI Enable Registers (FCCU_NMI_EN[0:3])
- FCCU EOUT Signaling Enable Registers (FCCU_EOUT_SIG_EN[0:3])

Workaround:

In case of configuration timeout, when re-configuring the FCCU_DELTA_T, FCCU_IRQ_ALARM_ENx, FCCU_NMI_ENx and FCCU_EOUT_SIG_ENx registers, take into account that they may not contain the reset value.

1.303 ERR007234: 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.

1.304 ERR007246: SARADC: First conversion after exit from stop mode may be corrupted

Description:

In the Successive Approximation Analog to Digital Converter (SARADC), if a chain conversion is on going and a transition to stop mode request is done, the result of the first conversion after exit of the stop mode (in other words, the conversion that was interrupted when going into stop mode) will be corrupted in the following cases:

Case A: the peripheral bridge clock (PBRIDGEx_CLK) becomes lower than the SARADC clock (SAR_CLK)

Note: This might be the case if the input of the PBRDIGEx_CLK divider is changed during the mode transitions (from the output of the PLL to the internal RC Oscillator for example)

Case B: the PBRIDGEx_CLK is resumed after the SAR_CLK, with a delay greater than 10 cycles of SAR_CLK

Workaround:

The following workarounds are possible:

1. Disable PBRIDGEx_CLK during stop mode and enable it only with a configuration such that it is greater than SAR_CLK.

OR

2. Verify that no analog conversion is ongoing before issuing a stop mode request by reading the ADC status field of the Main Status Register (SARADC_x.MSR[ADCSTATUS] = 0b000, also known as IDLE).

OR

3. Ignore the result of the first conversion after stop mode exit, considering it as corrupted.

1.305 ERR007259: e200zx: ICNT and branch history information may be incorrect following a nexus overflow

Description:

If an internal Nexus message queue over-flow occurs when the e200zx core is running in branch history mode (Branch Method bit [BTM] in the Development Control register 1 [DC1] is set [1]), the instruction Count (ICNT) and branch history (HIST) information in the first program trace message following the Program Correlation message caused by an over-flow of the internal trace buffers, will contain incorrect ICNT and HIST information.

This can also occur following an overflow of the internal Nexus message queues in the traditional branch mode (BTM in the DC1 is cleared [0]). Traditional branch mode Nexus messages do not include HIST information, since all branches generate a trace message.

Workaround:

There are two methods for dealing with this situation.

1. Avoid overflows of the Nexus internal FIFOs by reducing the amount of trace data being generated by limiting the range of the trace area by utilizing watchpoint enabled trace windows or by disabling unneeded trace information, or by utilizing the stall feature of the cores.
2. After receiving an overflow ERROR message in Branch History mode, the ICNT and HIST information from the first Program Trace Synchronization message and the next Program Trace message with a relative address should be discarded. The address information is correct, however, the ICNT and previous branch history are not correct. All subsequent messages will be correct.

In traditional branch mode, the ICNT information should be discarded from the Program Trace Sync message and the next direct branch message.

1.306 ERR007274: LINFlexD: Consecutive headers received by LIN Slave triggers error interrupt

Description:

As per the Local Interconnect Network (LIN) specification, the processing of one frame should be aborted by the detection of a new header sequence.

In LINFlexD, if the LIN Slave receives a new header instead of data response corresponding to a previous header received, it triggers a framing error during the new header's reception. The LIN Slave still waiting for the data response corresponding to the first header received.

Workaround:

The following three steps should be followed -

1. Set the MODE bit in the LIN Time-Out Control Status Register (LINTCSR[MODE]) to '0'.
2. Set Idle on Timeout in the LINTCSR[IOT] register to '1'.
3. Configure master to wait until the occurrence of the Output Compare flag in LIN Error Status Register (LINESR[OCF]) before sending the next header. This flag causes the LIN Slave to go to an IDLE state before the next header arrives, which will be accepted without any framing error.

1.307 ERR007297: LINFlexD: Response timeout values is loaded in LINOCR[OC2] field instead of LINOCR[OC1]

Description:

In the LINFlex module, the response timeout value calculated by hardware is loaded onto the OC2[7:0] (Output Compare 2) bits of LINOCR (LIN Output Compare Register) instead of being loaded into the OC1[7:0] (Output Compare 1) bits of the same register as stated in the documentation.

This applies when the Time-out counter mode is enabled by clearing the MODE (Time-out counter mode) bit in the LINTCSR (LIN Timeout Control Status Register).

Workaround:

Expect that OC2[7:0] (Output Compare 2) bits are loaded by hardware with the response timeout value when the MODE (Time-out counter mode) bit in LINTCSR is cleared.

1.308 ERR007305: e200zx: JTAG reads of the Performance Monitor Counter registers are not reliable

Description:

Reads of the Performance Monitor Counter (PMC0, PMC1, PMC2, and PMC4) registers through the IEEE 1149.1 or IEEE 1149.7 (JTAG) interfaces may return occasional corrupted values.

Workaround:

To ensure proper performance monitor counter data at all times, software can be modified to periodically read the PMCx values and store them into memory. JTAG accesses could then

be used to read the latest values from memory using Nexus Read/Write Access or the tool could enable Nexus data trace for the stored locations for the information to be transmitted through the Nexus Trace port.

1.309 ERR007339: 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.

1.310 ERR007352: DSPI: reserved bits in slave CTAR are writable

Description:

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

Workaround:

There are two possible workarounds.

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

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

1.311 ERR007356: 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.

1.312 ERR007362: SDADC: Additional DMA request generated after single read access

Description:

The Sigma-Delta Analog-to-Digital Converter (SDADC) issues an extra transfer request when the FIFO full Direct Memory Access (DMA) channel is configured to read only 1 data value from the SDADC.

Therefore, when the FIFO (First-In-First-Out) Threshold (FTHLD) field of the SDADC FIFO Control Register (FCR) is 0 (1 conversion) or when the FIFO Enable bit (FE) of FCR is 0 (FIFO is disabled), the extra read request will return invalid data.

The first DMA read access to the SDADC (correct read) returns good data, the second one

(extra access, unwanted) returns the contents of the FIFO (undefined, old conversion results).

Workaround:

Workaround 1:

Configure the SDADC FIFO threshold to a number N greater than 0 and its FIFO full DMA channel to read at least 2 conversion results at a time.

Workaround 2:

If available (not in use within the application software), use the SDADC watchdog DMA channel, that is not generating extra requests, instead of the SDADC FIFO full channel. Use the following settings:

- FIFO threshold set to 1
- Watchdog high and low threshold set to 0

1.313 ERR007404: SENT: Message overflow in SENT Receiver can lead to stall condition in the MCU

Description:

Under certain conditions, the Single Edge Nibble Transmission (SENT) Receiver (SRX) stalls and the Fast Message Data Ready bit for the SENT channel (FMSG_RDY[F_RDYn]) will no longer get set to indicate that a fast message is available. Reads of any of the fast message registers by the MCU core will stall and not complete. The registers affected are:

Register	Register Name
DMA_FMSG_DATA	Direct Memory Access (DMA) Fast Message Data Read Register
DMA_FMSG_CRC	DMA Fast Message Cyclic Redundancy Check Register
DMA_FMSG_TS	DMA Fast Message Time-stamp Register
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

A stall may occur if an overflow status condition is detected in the SENT Receiver Channel Status register (CHn_STATUS[FMSG_OFLW] = 1).

The overflow occurs when two messages are allowed to queue in the internal buffers of the SENT Receiver.

Workaround:

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 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 eDMA access 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 SENT 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).

If the stall occurs, a reset will be required to clear the stall condition. A Software Watchdog Timer (SWT) should be enabled to force a reset of the MCU if the device becomes stalled.

1.314 ERR007409: FLASH: Do not issue an interlock write while in program suspend within erase suspend

Description:

In the Flash module, when a program operation is started then suspended while an erase operation was already suspended, performing an interlock write access to the Flash may lead to the corruption of the location in program suspend.

Workaround:

Do not make any interlock write access once a program suspend has been requested within an erase suspend.

1.315 ERR007415: JTAG: PA[9] = JTAG TDO pad is not pull-up during reset

Description:

The GPIO PA[9] which is also used as JTAG TDO is configured as high impedance during power-up and while JTAG is under reset.

Workaround:

Use an external pull-up on pin PA[9] in case a high level is required during power-up and reset.

1.316 ERR007417: SDADC: Gain error above specification after calibration

Description:

Even after calibration, the Sigma-Delta Analog-to-Digital Converter (SDADC) may have a gain error of up to 0.25% instead of 0.1%.

Workaround:

Expect up to 0.25% gain error after calibration.

1.317 ERR007425: 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 2* contains information due to erratum behavior. *Table 3* contains clarification of normal NUM_EDGES_ERR behavior.

Table 2. 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 errors
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 (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.	Ignore NUM_EDGES_ERR error

Table 3. 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

1.318 ERR007433: JTAGM: Nexus error bit is cleared by successful RWA

Description:

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

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

Workaround:

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

1.319 ERR007455: FCCU: A FCCU failure to react does not generate destructive reset

Description:

The Fault Collection and Control Unit (FCCU) is expected to generate a destructive reset if it does not react to an incoming Non Critical Fault (NFC) in a timely manner. After receiving a NCF the FCCU enters the ALARM state. At this point there are two internal timers that start to run. The FCCU timer set by the user in the NCF Time-Out register (FCCU.NCF_TO) and the FCCU Output Supervisor Unit (FOSU) timer. If the fault is recovered before the

FCCU timer expires there is no FCCU reaction. If the FCCU timer expires the FCCU moves to the FAULT state and executes the programmed reaction. Once an appropriate reaction occurs (reset, IRQ) the FOSU timer is reset. If the FCCU cannot provide an appropriate reaction to the NCF due to some fault occurring in FCCU itself then the FOSU timer will expire and the FCCU is expected to generate a destructive reset.

Workaround:

You cannot rely on the FCCU to properly generate a destructive reset based on the failure of the FCCU to react to an NCF.

1.320 ERR007528: GTM: Action not always calculated immediately by DPLL

Description:

(GTM-IP-170)

If the Generic Timer Module (GTM) Digital PLL (DPLL) action calculation is interrupted by a new input event, the next TRIGGER/STATE input event action calculation (after the sub increment calculation is finished) starts at the previous internal action number. If new action data arrives during the sub increment calculation it is only used after the next input event. New Position Minus Time (PMT) data for an action with a higher action number is not recognized immediately.

Normally, the calculation of sub increments and PMT are not done in parallel because of

resource sharing. When the DPLL is doing the action calculation it has exclusive access rights to RAM Region 1a which contains the PMT request values, so the DPLL cannot accept new PMT requests via the Advanced Routing Unit (ARU). Therefore requested actions are not calculated regularly with every tooth.

Workaround:

The GTM should only request actions which are not "past" with every new tooth. The synchronization of the Multi Channel Sequencer (MCS) task to Timer Input Module (TIM) input event can be done by routing the TIM edge capture event value via ARU to MCS. If new PMT data arrives after the action number has reached the value zero, the action is calculated immediately starting with the highest action number again.

You can request the action calculation tooth by tooth until an action runs in to the past. Additional PMT requests can be placed earlier while the DPLL is performing sub increment calculations because RAM Region 1a is exclusively used for PMT requests via ARU.

Send PMT requests at least 3 teeth before the action has to be executed. This ensures that the MCS and ARU Connected Timer Output Module (ATOM) get action results from a calculation an input event cycle before.

1.321 ERR007529: GTM: TIM overflow bit is not set and the signal level bit has inverse value when sent to ARU in some cases

Description:

(GTM-IP-172)

When the Generic Timer Module (GTM) Timer Input Module (TIM) is in Timer Input Event Mode (TIEM, TIMn_CHx_CTRL[TIM_MODE] = 2), with Advanced Routing Unit (ARU) enabled (TIMn_CHx_CTRL[ARU_EN] = 1), and Input Signal Level high (ISL, TIMn_CHx_CTRL[ISL] = 1, the Overflow Bit (ACB1) might not be set and the signal level bit (ACB0) will be incorrect.

This error occurs when two input signals change in close proximity (faster than the ARU routing time), for example, an edge initiates an ARU transfer and one system clock before the ARU request is serviced the second input signal changes. Note that the Interrupt Request bit associated with the Overflow is set correctly.

Workaround:

Workaround 1:

Use the TIM channel input filter to remove signal changes smaller than the ARU routing time by configuring the filter parameters for rising and falling edges (TIMn_CHx_FLT_FE/TIMn_CHx_FLT_RE) with a delay which is greater than the ARU routing time.

Workaround 2:

Select the Edge Counter (TIMn_CHx_ECNT or TIMn_CHx_CNT) to be transferred in the ARU data to the Mutli Channel Sequencer (MCS) and use the MCS to reconstruct the correct TIM data as follows:

```
Last_CNT = -1
For each ARU_DATA
  If ARU_DATA(ACB1) ==0
    If Last_CNT != -1
      If Last_CNT+1 != ARU_DATA(CNT)
        Message(Hit on ERRATA:  Detected overflow condition) ARU_DATA(ACB1) = 1
        ARU_DATA(ACB0) = not ARU_DATA(ACB0)
      else
      Message(No signal level present yet, cannot apply workaround)
    Last_CNT = ARU_DATA(CNT)
```

1.322 ERR007530: GTM: New DPLL Position Minus Time data not received

Description:

(GTM-IP-173)

When the Generic Timer Module (GTM) Digital PLL (DPLL) receives Position Minus Time (PMT) requests after a TRIGGER/STATE event, only that request can be considered. The DPLL blocks new PMT requests for about 200 ns. New PMT requests are only accepted after the calculation of the pending action calculations are performed. This calculation starts in the state machine, about 10 us after the input event and completes depending on the number of actions (A) to be calculated A*3.7 us later. After this time the PMT request is accepted, but it is not possible to adjust the action calculation with updated data. The "old" value is always calculated.

The PMT result is calculated based on older PMT input data because the pending data transfer with newer input data to the DPLL cannot be executed.

Workaround:

When the calculated action is transmitted to the Multi Channel Sequencer (MCS), check if there was an Advanced Routing Unit (ARU) transfer with new data for this action blocked by the ARU because the DPLL was not ready to receive new data within this time. If the ARU transfer was just completed, the corresponding action contains only the older PMT

requirements. Ignore this action value and wait for the new value which appears about 3.7 us after the PMT requirement update was transmitted.

1.323 ERR007531: GTM: DPLL Position Minus Time result is not sent to the ARU

Description:

(GTM-IP-174)

The Generic Timer Module (GTM) Digital PLL (DPLL) has a state where there is a delay between when the New Output Data Values Concerning To Action t bit (DPLL_ACT_STA[ACT_N(t)]) is reset and when the corresponding shadow bit (DPLL_ACT_STA_shadow[ACT_N(t)]) is set to "1", which starts the transfer of the output data via the Advance Routing Unit (ARU).

If during this delay a new input event occurs (DPLL_ACT_STA[ACT_N(t)]) and the internal state controller changes to process this new input event, DPLL_ACT_STA_shadow[ACT_N(t)] is not yet set, so there is no request to transmit the output data to the ARU. In this case, an action calculation is finished without transferring the data via the ARU. PMT calculations where the result is not "past" are not affected by this issue. The time frame in which a incoming input signal causes the issue is about 25 system clock cycles.

Workaround:

Use a Multi Channel Sequencer (MCS) channel to read the PMT data from DPLL via non blocking ARU reads using the NARD or NARDI instructions. The data that is read should be tested to check whether the requested action is 'out of time' by reading the Time Base Unit Time Stamp Channel 0 (TBU_TS0), or 'out of angle' by reading the Time Base Unit Time Stamp Channel 1 or 2 (TBU_TS1/2). If the TBU_TSx values are not within an acceptable window of the PMT value, the MCS can request the old value again, or if the requested event is in the past, request a new value.

Additionally, the Timer Input Module 0 (TIM0) interrupt can be routed to the MCS to check if an active edge occurred. Each action which delivers a PMT result should be checked only once by the MCS. If the PMT result is transferred directly from the DPLL to the ARU connected Timer Output Module (ATOM), the MCS should be prepared to send a default value to the ATOM which is not in the past to ensure that even if the DPLL fails to send the PMT result to the ATOM, the ATOM does not miss the event completely.

1.324 ERR007532: M_TTCAN: Incorrect value of Reference Trigger Offset status for time slaves

Description:

When the Time Triggered Modular CAN (M_TTCAN) module is configured as time slave, read accesses to the Reference Trigger Offset (RTO) field of the TT Operation Status register (TTOST) always return the value 0x7F when the Error Level (EL) 2-bit field of TTOST is greater than 0b00, signalling the presence of error in the TTCAN operation.

The M_TTCAN should return the value configured in the Initial Reference Trigger Offset (IRTO) field of the TT Operation Configuration register (TTOCF).

Workaround:

Ignore the value of the RTO field when reading the TTOST register for time slaves.

1.325 ERR007567: M_(TT)CAN: RX buffer and CAN FD 8-byte frames not supported

Description:

The following new features included in the Modular Controller Area Network (M_CAN) or the Time Triggered Modular CAN (M_TTCAN) are not available:

- Receive Buffer
- Support of CAN Flexible Data Rate (FD) 8-byte frames or 64-byte frames

In addition, the bitfields of the Bit Timing and Prescaler (BTP) register are defined as follow:

- Bits 0:11, Reserved
- Bits 12:15, BRPE[3:0]: Baud Rate Prescaler Extension
- Bits 16,Reserved
- Bits 17:19, TSEG2[2:0]: The time segment after the sample point
- Bits 20:23, TSEG1[3:0]: The time segment before the sample point
- Bits 24:25, SJW[1:0]: (Re) Synchronization Jump Width
- Bits 26:31: BRPL[5:0]: Baud Rate Prescaler Low

Workaround:

Take into account the BTP register definition. Do not use RX Buffer feature or CAN FD.

As a consequence, do not access the following registers or bitfields in this revision of the device:

Register/Bitfield	Description
FBTP	Fast Bit Timing & Prescaler Register
TEST.TDCV	Test register Transceiver Delay Compensation Value
CCCR.FACT	CAN Core Control Register Fast Frame Mode Active flag
CCCR.LACT	CAN Core Control Register Long Frame Mode Active flag
CCCR.CME	CAN Core Control Register CAN Mode Enable
CCCR.CMR	CAN Core Control Register CAN Mode Request
PSR.REDL	Protocol Status Register Received a CAN FD Message
PSR.RBRS	Protocol Status Register BRS flag of last received CAN FD Message
PSR.RESI	Protocol Status Register ESI flag of last received CAN FD Message
PSR.FLEC	Protocol Status Register Fast Last Error Code
IR.DRX	Interrupt Register Message stored to Dedicated Rx Buffer
IE.DRXE	Interrupt Enable Message stored to Dedicated Rx Buffer Interrupt Enable
ILS.DRXL	Interrupt Line Select Message stored to Dedicated Rx Buffer Interrupt Line
NDAT1	New Data 1 register (new data stored in Rx buffers 0 to 31)
NDAT2	New Data 2 register (new data stored in Rx buffers 32 to 63)
RXBC	Rx Buffer Configuration
RXF1S.DMS	Rx FIFO 1 Status Debug Message Status(1)

1. For devices with support for debug via CAN.

1.326 ERR007589: LINFlexD: Erroneous timeout error when switching from UART to LIN mode

Description:

When the LINFlexD module is enabled in Universal Asynchronous Receiver/Transmitter (UART) mode and the value of the MODE bit of the LIN Timeout Control Status register (LINTCSR) is 0 (default value after reset), any activity on the transmit or receive pins will cause an unwanted change in the value of the 8-bit field Output Compare Value 2 (OC2) of the LIN Output Compare register (LINOCR).

As a consequence, if the module is reconfigured from UART to Local Interconnect Network (LIN) mode, an incorrect timeout exception is generated when a LIN communication starts.

Workaround:

Before enabling UART communication, set to 1 the MODE bit of the LIN Timeout Control Status register (LINTCSR) (selecting the output compare mode). This is preventing the LINOCR.OC2 field from being updated during UART communications.

Then, after reconfiguring the LINFlexD to LIN mode, reset the LINRCSR.MODE bit (selecting the LIN mode) before starting LIN communications

1.327 ERR007788: SIUL2: A transfer error is not generated for 8-bit accesses to non-existent MSCRs

Description:

An 8-bit access attempt to non-existent MSCRs (Multiplexed Signal Configuration Registers) in the SIUL2 (System Integration Unit Light 2) address space does not generate a transfer error. 16-bit or 32-bit accesses to non-existent MSCRs will generate a transfer error.

Workaround:

Do not expect transfer errors on 8-bit accesses to non-existent MSCRs in the SIUL2 address space.

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

Description:

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

Workaround:

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

1.329 ERR007824: DCI: Avoid asserting system reset when switching JTAG operating modes

Description:

Assertion of system reset during the transition of the debug pin operating mode, either from JTAG pin mode to the LVDS Fast Asynchronous Serial Transmission (LFAST) pin mode, or from LFAST pin mode to JTAG pin mode, could result in a loss of synchronization with the debugger or a reset of the debug system.

Workaround:

Tools should not assert system reset while the Debug and Calibration Interface (DCI) is switching the pin operating mode from JTAG to LFAST, or from LFAST to JTAG. If a system reset occurs due to any other conditions, the tool may lose communication with the microcontroller. If this occurs, the tool should reset the JTAG interface by toggling JCOMP (DEBUG_RXN) low (ground) while holding the TDO (DEBUG_RXP) pin either high or low. This forces the interface operation back to JTAG operating mode. This requires that the Enable Escape mode feature be enabled in the DCI Control Register (DCI_CR[EN_ESC_MODE] = 1).

1.330 ERR007847: GTM: MCS's CAT status may be incorrect

Description:

(GTM-IP-178)

The Generic Timer Module (GTM) Multi-channel Sequencer's (MCS) Advance Routing Unit (ARU) blocking read/write instructions, such as ARD, AWR, ARDI, and AWRI, describe the use of the Cancel ARU Transfer (CAT) status field to check whether the last ARU transfer was successful (CAT=0) or canceled (CAT=1). Because CAT can be written by software to cancel an ARU transfer at any time, CAT does not reliably reflect the last ARU transfer status.

Workaround:

Check data consistency of the ARU transfer by inspecting the transferred data (for example, check for the linear increment of the edge counter (ECNT) for data transfers from Timer Input Module (TIM) to the MCS) instead of relying on the CAT field.

1.331 ERR007848: GTM: Bit 0 of TIM edge counter register may not indicate the actual signal level

Description:

(GTM-IP-181)

When a Generic Timer Module (GTM) Timer Input Module (TIM) channel is enabled, bit 0 of the Edge Counter register (ECNT) may not reflect the current signal level of the filtered input TIM[i]_CH[x]_FOUT until the next input edge occurs. This issue occurs when the ECNT register is not read before re-enabling the channel.

This erratum does not affect TIM Bit Compression Mode (TBCM).

Workaround:

After disabling the TIM channel, ensure that the ECNT register is read at least once before the TIM channel is re-enabled. Alternatively. before re-enabling a TIM channel, issue a TIM channel reset and reconfigure the TIM channel control registers.

1.332 ERR007855: SENT: Integer division during calibration pulse measurement causes reduced robustness

Description:

The calculation of the compensated microticks during the calibration pulse requires a division by 56. This constant integer division introduces quantization error that accumulates over each microtick while receiving the Status, Data and Cyclic Redundancy Check (CRC) nibbles.

This accumulated error leads to a reduced jitter tolerance window. As a result, incorrect values of nibbles may get sampled. Compensated tick period can be read from Channel ‘n' Clock Control Register (SRX_CHn_CLK_CTRL), Compensated Prescalar value (CM_PRSC) field.

Workaround:

The Nibble Length Variation Limit (NIB_LEN_VAR_LIMIT, bit 16 [Least Significant Bit is 31]) bit in the SENT Global Control (SRX_GBL_CTRL) register should be set to allow for additional tolerance to jitter and frequency variation in the received SENT signal.

In addition, use a nominal micro-tick duration greater than or equal to 8us (with worst case u-tick duration of -25% which is greater than or equal to 6us) if the High Frequency (protocol clock) is 80MHz.

For a 40 MHz protocol clock, the nominal u-tick duration must be greater than or equal to 13us (with worst case u-tick duration of -25% which is greater than or equal to 10us).

For a 100 Mhz protocol clock, the nominal u-tick duration must be greater than or equal to 7us (with worst case u-tick duration of -25% which is greater than or equal to 5us).

The protocol clock frequency (40, 80, or 100 MHz) should be set per the device capabilities.

1.333 ERR007886: SENT: Jitter tolerance is limited to 1/8 of the utick time

Description:

The Single Edge Nibble Transmitter (SENT) Receiver does not properly round off incoming data to nearest nibble. The SAE J2716 (SENT) specification dated January 2010 (revision 3) jitter specification is not met which leads to an incorrect rounding of nibble measurement. As a result, the Channel n Fast Message Data Read Register (CHn_FMSG_DATA) or Channel n Fast Message Cyclic Redundancy Check register (CHn_FMSG_CRC) values may be incorrect for the Status, Data, and Cyclic Redundancy Check values or the Message is not received at all because of the CRC mismatch.

Workaround:

When the total accumulated jitter added by the SENT transmitter is less than or equal to 10% of the total utick duration, message reception at the SENT receiver would be correct. Use SENT transmitter devices that have total accumulated errors that results in 10% of the uTick period or less.

1.334 ERR007947: XOSC: Incorrect external oscillator status flag after CMU event clear

Description:

If an external oscillator (XOSC) is enabled and it becomes unstable (or the crystal fails), the Oscillator Lost Reference status flag in the Clock Monitor Unit Interrupt Status register (CMU0.CMU_ISR[OLRI]) will be set. In addition, the Crystal Oscillator Status flag in the Mode Entry module Global Status Register (MC_ME_GS.S_XOSC) will be cleared (1 = stable clock, 0 = no valid clock). However, if the CMU_ISR[OLRI] is cleared while the oscillator is still in a failing condition, the MC_ME_GS.S_XOSC will incorrectly be set, indicating a valid crystal oscillator.

Workaround:

Monitor the XOSC external oscillator status using the MC_ME_GS.S_XOSC before the CMU0.CMU_ISR.OLRI flag is set. After the CMU0.CMU_ISR.OLRI flag has been set, the MC_ME_GS.S_XOSC flag is valid only after a functional reset. Alternately, the response to the OLRI flag after loss of XOSC clock, can be set in the FCCU to cause a functional reset to clear the MC_ME_GS.S_XOSC flag.

1.335 ERR007996: PSI5: Incorrect SMC message decoding and timestamp generation in case of late last sensor message overlapping with next SYNC period pulse

Description:

As stated in section 6.6 of Peripheral Sensor Interface (PSI5) Standard v2.0 and v2.1, PSI5 sensor frames should not overlap with the SYNC pulse. This overlap is considered to be an error condition. In extension to the standard requirement, the PSI5 module implemented in this device allow the possibility to manage this overlap error condition.

In case of such overlap condition:

- PSI5 message extraction happens correctly
- Timing bit error [T] and CRC error [C] flags are correctly set within the PSI5 message
- The serial messaging channel (SMC) frame counter gets reset on Sync pulse and the extraction of SMC message does not happen correctly, resulting in loss of SMC message

The PSI5 module correctly handles this error condition for the PSI5 message, but is not able to handle the SMC message correctly.

Also during this overlap condition, the timestamp appended with the overlapped slot is the new sync pulse timestamp. Whenever a sync pulse comes, the internal sync pulse timestamp capture registers are updated with the timestamp of the new sync pulse. Hence if any message overlaps with the sync pulse and that message slot is configured to capture the timestamp of the SYNC pulse (PSI5_SnFCR[TS_CAPT] = 1), then the timestamp appended for that slot is the timestamp corresponding to the new sync pulse and not of the previous sync pulse belonging to the PSI5 frame.

Workaround:

The PSI5 message is properly received and analysed with the appropriate error flags set. Application software can identify from the properly received PSI5 message that an error on the bus occurred. If an SMC message is configured to be present in the slot, application software can request an immediate halt and restart of the SMC or it can wait until the SMC reception has finished and check the CRC of the SMC message

- If incorrect, a resend of the SMC message can be re-started at that point in time. Further, application software can check the timestamp of the previous message in the final slot from the previous SYNC period to identify if the wrong timestamp has been captured
- If the difference between the two messages is equivalent to two SYNC periods, application software can correctly identify that an overlap of the final message with the following SYNC pulse has occurred assuming the slot is configured to capture the SYNC pulse timestamp (PSI5_SnFCR[TS_CAPT] = 1) rather than the message timestamp (PSI5_SnFCR[TS_CAPT] = 0)

1.336 ERR008039: 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 (VDD_HV_ADV).

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 Clock Generation Module (CGM) or Select the External Modulator Mode (EMSEL) by setting the MCR[EMSEL] bit along with the clearing the MCR[EN].

1.337 ERR008042: 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) which are configured by setting the Safe MODE Control bit in the System Integration Unit Lite 2 Multiplexed Signal Configuration Register (SIUL2_MSCR_IO_n [SMC]) to not be disabled when the FCCU moves into a fault condition.

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 (e.g. 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.

1.338 ERR008054: PIT: DMA request stays asserted when initiated by PIT trigger, until PIT is reset

Description:

When a Periodic Interrupt Timer 0 (PIT0) channel trigger is used to initiate a Direct Memory Access (DMA) transfer, the DMA request does not negate at the end of the DMA transfer. The result is that if that DMA channel is re-enabled, a subsequent PIT-triggered DMA transfer will be initiated.

Workaround:

Either do not use the PIT0 to initiate DMA transfers, or write software such that anytime a PIT0 channel trigger is used to initiate a DMA transfer, the PIT0 module is then reset after that DMA transfer completes, prior to re-enabling the DMA channel that was used for that transfer. The PIT0 module should be reset by setting the PIT_RTC_0 reset bit in the

Peripheral Reset Register 0 in the Reset Generation Module.

Other timer systems, such as the System Timer Module (STM), can be used instead of the PIT to trigger the DMA.

1.339 ERR008082: 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

1.340 ERR008117: MC_ME: Restrictions on enabling FlexRay in low power modes

Description:

The FlexRay module is dependent on the Auxiliary (AUX) Clock 2, for which the only source clock is Phase Lock Loop 0 (PLL0). This dependency between FlexRay and PLL0 results in the following restrictions during entry to the low power modes, STOP0 and HALT0:

Entry into STOP0 with FlexRay enabled (by setting MC_ME_PCTLx[LP_CFG=n] and MC_ME_LP_PCn[STOP0]=1) is not possible, even if the Crystal Oscillator (XOSC) is selected as the FlexRay protocol clock source (FR_MCR[CLKSEL]=0).

Entry into HALT0 with FlexRay enabled (by setting MC_ME_PCTLx[LP_CFG=n] and MC_ME_LP_PCn[HALT0]=1) is possible only if the PLL0 is enabled during HALT0 (by setting MC_ME_HALT0_MC[PLL0ON]=1).

(For MC_ME_PCTLx, x=107 for FlexRay0 and x=235 for FlexRay1.)

Workaround:

To enter STOP0, the FlexRay must be disabled (MC_ME_PCTLx[LP_CFG=n] and MC_ME_LP_PCn[STOP0]=0). To enter HALT0 with FlexRay enabled, enable the PLL0 in HALT0 (by setting MC_ME_HALT0_MC[PLL0ON]=1) prior to entering HALT0.

(For MC_ME_PCTLx, x=107 for FlexRay0 and x=235 for FlexRay1.)

1.341 ERR008122: GTM: (A)TOM's CCU1 event interrupt is not generated when CM1=0 or 1 and RST_CCU0=1

Description:

(GTM-IP-202)

If a Generic Timer Module (GTM) Timer Output Module (TOM) or Advanced Router Unit (ARU) connected TOM (ATOM) channel is configured with the reset source of the channel as the previous channels trigger (CHn_CTRL[RST_CCU0]=1) and the counter 0 (CN0) counts from 0 to MAX, the Counter Compare Unit 1 (CCU1) event interrupt is not generated if the Compare Register 1 (CM1) is 0 and Compare Register 0 (CM0) is greater than 0. If the Compare Register (CM1) is 1 and Compare Register 0 (CM0) is MAX+1 only one CCU1 interrupt will be generated.

Workaround:

To trigger channel x+1 which is the channel that triggers when channel x counter CN0 is reset use the configuration of CM0=MAX and CM1=1.

- When the duty cycle configuration is CM1=0 and CM0>0 on channel x use the CCU0 interrupt of triggering channel x+1 instead of CCU1 interrupt.
- When the duty cycle configuration is CM1=1 and CM0=MAX+1 on channel x use the CCU1 interrupt of triggering channel x+1 instead of CCU1 interrupt on channel x.

1.342 ERR008131: SPC57BD1: Boundary scan of the interconnect between PD and BD is not available

Description:

Boundary Scan of the interconnect between the Emulation Device "Buddy Die (BD)" to the main production MCU die is not available. There are no boundary scan cells for the interconnect to the BD and the BSDL file for the device does not include the interconnect in the boundary scan chain.

Workaround:

Do not attempt to perform Boundary Scan operations to test the interconnect between the BD and main production MCU. Boundary Scan of the main production MCU is possible, but the interconnect between the BD and main production MCU is not included in the boundary scan chain.

Appendix A Further information

A.1 Reference document

1. *SPC574Kxx - 32-bit Power Architecture® based MCU for automotive applications* (RM0334, Doc ID 023671).
2. *32-bit Power Architecture® based MCU for automotive applications* (SPC574Kxx datasheet, Doc ID 023601).

A.2 Acronyms

Table 4. Acronyms

Acronym	Name
PIR	Processor Identification Register
HSM	Hardware Security Module
SPR	Special Purpose Register
SWT	System Watchdog Timers
MEMU	Error Management Unit
MBIST	Memory Built-in Self Test unit
PSRAM	Peripheral SRAM
FEC	Fast Ethernet Controller
RIF	Receive Interface FIFO
MIB	Message Information Block
ECC	Error Correction Code
GTMDI	GTM Debug Interface
AMU	Arithmetic Modeling Unit
SIU2	System Integration Unit Lite 2
SSS	Source Signal Select
MCSR	Multiplexed Signal Configuration Register
ADC	Analog to Digital Converter
DPLL	Digital Phase Lock Loop
FCCU	Fault Collection Control Unit
MC_ME	Mode Entry Module
MC_CGM	Clock Generation Module
BAF	Boot Assist Flash
FEC	Fast Ethernet Controller
PISR	Parallel Input Select Registers
DSPI	Deserial Serial Peripheral Interface module

Table 4. Acronyms (continued)

Acronym	Name
FOSU	Fault Collection and Control Unit Output Supervision unit
ADR	Flash Address Register
PSI5	Peripheral Sensor Interface
LFAST	LVDS Fast Asynchronous Serial Transmission
RGM	Reset Generation Module
IMEM	Instruction Memory
SARADC	Successive Approximation Register Analog to Digital Converter
SENT	Single Edge Nibble Transmission
OS	Operating System
CLC	Crossbar Lock Control module
SFEC	Standard Filter Element Configuration
EFEC	Extended Filter Element Configuration
LINSR	LIN Status Register

Revision history

Table 5. Document revision history

Date	Revision	Changes
15-May-2013	1	Initial release.
18-Sep-2013	2	Updated Disclaimer.
18-Jul-2014	3	Removed functional problems: – ERR003924 – ERR005019 – ERR005022 Added functional problems: – ERR003632 – ERR003660 – ERR003881 – ERR003922 – ERR004764 – ERR004986 – ERR005006 – ERR005039 – ERR005049 – ERR005073 – ERR005107 – ERR005137 – ERR005584 – ERR005604 – ERR005748 – ERR005824 – ERR005856 – ERR005857 – ERR005860 – ERR005884 – ERR005912 – ERR005984 – ERR005999 – ERR006018 – ERR006019 – ERR006026 – ERR006042 – ERR006084 – ERR006085

Table 5. Document revision history

Date	Revision	Changes
18-Jul-2014	3 (continued)	– ERR006087 – ERR006090 – ERR006099 – ERR006194 – ERR006237 – ERR006290 – ERR006292 – ERR006349 – ERR006350 – ERR006353 – ERR006361 – ERR006364 – ERR006370 – ERR006373 – ERR006383 – ERR006401 – ERR006409 – ERR006410 – ERR006411 – ERR006412 – ERR006427 – ERR006444 – ERR006445 – ERR006459 – ERR006477 – ERR006538 – ERR006544 – ERR006552 – ERR006553 – ERR006597 – ERR006638 – ERR006639 – ERR006640 – ERR006642 – ERR006543 – ERR006644 – ERR006645 – ERR006720 – ERR006738 – ERR006792 – ERR006797 – ERR006800 – ERR006803

Table 5. Document revision history

Date	Revision	Changes
18-Jul-2014	3 (continued)	– ERR006804 – ERR006806 – ERR006812 – ERR006815 – ERR006816 – ERR006819 – ERR006822 – ERR006828 – ERR006836 – ERR006839 – ERR006840 – ERR006841 – ERR006846 – ERR006847 – ERR006849 – ERR006852 – ERR006855 – ERR006858 – ERR006860 – ERR006863 – ERR006865 – ERR006873 – ERR006896 – ERR006902 – ERR006904 – ERR006905 – ERR006906 – ERR006908 – ERR006909 – ERR006915 – ERR006916 – ERR006932 – ERR006967 – ERR006990 – ERR007041 – ERR007046 – ERR007049 – ERR007051 – ERR007053 – ERR007055 – ERR007057 – ERR007058

Table 5. Document revision history

Date	Revision	Changes
18-Jul-2014	3 (continued)	– ERR007059 – ERR007061 – ERR007062 – ERR007063 – ERR007064 – ERR007066 – ERR007067 – ERR007068 – ERR007083 – ERR007084 – ERR007085 – ERR007086 – ERR007087 – ERR007088 – ERR007099 – ERR007103 – ERR007108 – ERR007109 – ERR007115 – ERR007116 – ERR007126 – ERR007129 – ERR007130 – ERR007131 – ERR007134 – ERR007137 – ERR007138 – ERR007140 – ERR007190 – ERR007191 – ERR007194 – ERR007196 – ERR007202 – ERR007203 – ERR007204 – ERR007211 – ERR007222 – ERR007223 – ERR007227 – ERR007232 – ERR007234 – ERR007246 – ERR007259 – ERR007274

Table 5. Document revision history

Date	Revision	Changes
18-Jul-2014	3 (continued)	– ERR007297 – ERR007305 – ERR007339 – ERR007352 – ERR007356 – ERR007362 – ERR007404 – ERR007409 – ERR007415 – ERR007417 – ERR007425 – ERR007433 – ERR007455 – ERR007528 – ERR007529 – ERR007530 – ERR005731 – ERR007532 – ERR007567 – ERR007589 – ERR007788 – ERR007791 – ERR007824 – ERR007846 – ERR007847 – ERR007848 – ERR007855 – ERR007886 – ERR007947 – ERR007996 – ERR008039 – ERR008042 – ERR008054 – ERR008082 – ERR008117 – ERR008122 – ERR008133