Mask Set Errata for Mask 0N36U

This report applies to mask 0N36U for these products:

- S32R372

Table 1. Errata and Information Summary

Erratum ID	Erratum Title
ERR010455	ADC: Calibration routine needs to be repeated after any SoC reset that triggers device built-in self-test
ERR011245	AFE: AFE_OSCSTS[STS] bit does not get cleared in single-ended and differential bypass mode of XOSC
ERR011122	AFE: Device may not exit the reset sequence if a low voltage detection on VDD_HV_RAW or VDD_HV_DAC supplies occurs when XOSC is the source of PLL0/1 clock.
ERR010337	AFE: The AFE_PLLSTS[LOR], AFE_OSCSTS[STS], and MC_ME_GS[S_XOSC] bits do not indicate loss of reference status
ERR011157	BAM: Serial boot loader is not supported by the device default configuration
ERR010393	CGM: CLK_OUT0 and CLK_OUT1 dividers may become stuck if clock selection is changed while dividers with divide by 2 are operational.
ERR011241	CGM: Device may become non-responsive if an LVD event occurs after changing the clock configuration
ERR010287	CGM: Under certain conditions the data or instruction crossbars may hang during a change of system clock dividers using the Divider Update Trigger Register
ERR011407	CMU: Sudden loss of clock does not signal the Fault Collection and Control Unit
ERR011394	CTE: Timing table is not executed when the CTE time base clock is set to 80 MHz and the table execution duration is 1 clock cycle
ERR050090	DSPI/SPI: Incorrect data may be transmitted in slave mode
ERR010538	End to End ECC: An uncorrectable ECC event from a slave to a master can cause invalid data reception
ERR011188	FCCU : Fault NCF[106] may occur when PLL Loss of Lock event occurs during MBIST execution when online STCU-selftest is run.
ERR010900	FCCU: False indication of a fault state for a single safe clock period can be generated on the error output pin
ERR007869	FCCU: FOSU monitoring of a fault is blocked for second or later occurrence of the same fault
ERR008004	FLASH: Array Integrity with Breakpoints enabled may skip addresses for certain RWSC and APC combinations

Table continues on the next page...

Table 1. Errata and Information Summary (continued)

Erratum ID	Erratum Title
ERR007991	FLASH: Rapid Program or Erase Suspend fail status
ERR050246	FlexCAN: Receive Message Buffers may have its Code Field corrupted if the Receive FIFO function is used
ERR009928	FlexPWM: Half cycle automatic fault clearing does not work in PWM submodule 0 under some conditions
ERR010330	JTAGM: An unexpected interrupt will occur if the Idle Interrupt is enabled after the end of frame transfer
ERR010329	JTAGM: JTAGM Data out registers are writable regardless of state of the Data Transfer Mode bit
ERR007433	JTAGM: Nexus error bit is cleared by successful RWA
ERR010328	JTAGM: Software Reset bit does not auto clear when written with 1 when an external tool is not connected
ERR008935	JTAGM: write accesses to registers must be 32-bit wide
ERR007274	LINFlexD: Consecutive headers received by LIN Slave triggers the LIN FSM to an unexpected state
ERR011172	MIPICSI2: Start of Transmission Error can occur at lower speeds
ERR010331	NAL: Trace connections to the device are lost if a device reset occurs
ERR008340	NPC: EVTO_B toggles instead of remaining asserted when used by the DTS if Nexus is not enabled
ERR010479	NPC: Nexus enable required for mode changes when a debugger is attached
ERR010340	NZxC3: ICNT and HIST fields of a Nexus message are not properly reset following a device reset
ERR010638	NZxC3: Nexus messaging becomes corrupted if more than one master (including a device core) is active and the core is subsequently disabled.
ERR010357	PASS: JTAG password match bypasses flash read protection set by PASS.LOCK3[RLx] bits
ERR007905	PIT: Accessing the PIT by peripheral interface may fail immediately after enabling the PIT peripheral clock
ERR050130	PIT: Temporary incorrect value reported in LMTR64H register in lifetimer mode
ERR010750	PMC: Device may not exit the reset sequence if a low voltage detect event occurs during execution of offline or online self-test
ERR010259	PMC: Device may not exit the startup reset sequence under certain power sequencing conditions until the STCU watchdog time-out interval elapses
ERR011156	PMC: During online PMC self-test if RESET_B is asserted then a spurious destructive or power on reset may occur and be reported as an LVD/HVD event in the RGM_DES register.
ERR010414	PMC: Low Voltage Detect (LVD) self test may incorrectly indicate a self test fail if the supply is outwith its operating range during power up
ERR010345	PMC: The Power Management Controller (PMC) module Analog Front End (AFE) Low Voltage Detect (LVD) interrupt may not work as a valid source of interrupt
ERR011032	PMC: Unexpected events when an LVD occurs on a supply with its LVD masked
ERR007791	SIUL2: Transfer error not generated if reserved addresses within the range of SIUL BASE + 0x100 to 0x23F are accessed
ERR009658	SPI: Inconsistent loading of shift register data into the receive FIFO following an overflow event
ERR010879	SPT: Adaptive scaling using count unoccupied bits from bit 23 down mode can introduce harmonics in the FFT output
ERR011329	SPT: After completion of a command sequence there is a time window in which work register access by CPU can cause the system to become non-responsive
ERR010352	SPT: AHB error status can be asserted for the incorrect SPT DMA when performing back to back AHB accesses

Table continues on the next page...

Table 1. Errata and Information Summary (continued)

Erratum ID	Erratum Title
ERR010997	SPT: Work register contents not deterministic after execution of STOP command
ERR010088	STCU2: Unexpected STCU self-test timeout can occur when a short sequence for external reset is triggered during execution of online self-test
ERR010880	SWT: SWT may not get serviced in Fixed and Incremental address mode

Table 2. Revision History

Revision	Changes	Date
1	Initial revision	Mar 2017
2	The following erratum was added. • ERR011407	May 2018
3	The following errata were added. • ERR050130 • ERR050246 • ERR011329 • ERR050090 • ERR011394 The following errata were revised. • ERR011407 • ERR010479 • ERR007869	Feb 2020

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

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

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

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

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

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

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

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

Workaround: When using single-ended or differential bypass modes of the XOSC, the user should use the MC_ME_GS[S_XOSC] bit instead of the AFE_OSCSTS[STS] bit for accurate status of the oscillator. If the AFE_OSCSTS[STS] interrupt is to be used, and if XOSC needs to be disabled after initial enabling then before enabling the XOSC again, the user needs to first clear the AFE_OSCCTRL[EN_EXT] bit and then set this bit again.

ERR011122: AFE: Device may not exit the reset sequence if a low voltage detection on VDD_HV_RAW or VDD_HV_DAC supplies occurs when XOSC is the source of PLL0/1 clock.

Description: If a low voltage detection (LVD) event occurs on the VDD_HV_RAW or VDD_HV_DAC supplies then the default reaction is not to perform a destructive reset of the device (reset reaction for LVD on AFE supplies is enabled by writing to MCB_AFE_LVD_MASK). However it has been found that when the XOSC is used as the source for PLL0 or PLL1 clock then following an LVD event on these supplies there is a chance that the device may become unresponsive if no reset reaction is enabled, or may not exit the reset sequence if the reset reaction is enabled. This occurence is considered to be extremely rare, but is theoretically possible. Additionally, if the device is still responsive, a CMU0_ISR[OLRI] flag may indicate an OLR event occurred.

Workaround: Any of the below workarounds, 1 or 2, can be used to prevent the device becoming non-responsive:

1. Program the dcl_soc_conf_3 DCF record in the Utest Flash with 0x00000009, and set MCB_AFE_LVD_MASK[AFE_REF_LVD_MASK] = 0b1. This configures the device such that if an AFE-LVD event occurs, it will issue a POR to the device. The device will not become stuck in reset and will restart properly.

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

The below workaround may be used if OLR event is seen.

3. Remove the XOSC as source of other clocks and clear (by writing 0b1 to) the CMU0_ISR[OLRI] flag. The XOSC can then be re-enabled and used.

ERR010337: AFE: The AFE_PLLSTS[LOR], AFE_OSCSTS[STS], and MC_ME_GS[S_XOSC] bits do not indicate loss of reference status

Description: The Sigma Delta PLL (SDPLL) status register's Loss of Reference bit AFE_PLLSTS[LOR], Oscillator Status Register Fail bit and Status bit (AFE_OSCSTS[FAIL], AFE_OCSTS[STS]), and Mode Entry Global Status External Oscillator Status bit (MC_ME_GS[S_XOSC]) do not provide the correct status of the SDPLL loss of reference when SDPLL reference clock is removed in the system. The incorrect bit states are listed below:

AFE_PLLSTS[LOR] improperly indicates 0 AFE_OSCSTS[STS] improperly indicates 1 MC_ME_GS[S_XOSC] improperly indicates 1

Workaround: Use the Clock Monitor Unit 0 (CMU_0) to detect the status of the oscillator clock.

ERR011157: BAM: Serial boot loader is not supported by the device default configuration

Description: By default the device is configured from factory to safely accept a differential XOSC oscillator input. Since the Boot Assist Module (BAM) serial boot loader functionality (boot from CAN or UART) can only be used with a crystal XOSC input then this functionality will not work by default and the device will hang if a serial boot is attempted.

Workaround: In order to use the BAM serial boot loader mode the user should first program a new version of the DCL_IPS_2 DCF in UTEST, modifying it to 0x00000000 (from the default of 0x00000010). This will enable the serial boot functionality to be used. It should be noted that this DCF record should be reverted to the default value of 0x00000010 if a differential XOSC input is to be used in the final application.

ERR010393: CGM: CLK_OUT0 and CLK_OUT1 dividers may become stuck if clock selection is changed while dividers with divide by 2 are operational.

Description: If clock out functionality is enabled on either CLK_OUT0 and/or CLK_OUT1 and there is a divide by 2 divider operational on these clocks (via CGM_AC14_DC0[DE] and/or CGM_AC9_DC0[DE] = 0b1), then if the clock selection for CLK_OUT is changed via the MCB_CLKOUT_SEL register or a Phase 1,2 or 3 reset occurs then the dividers may become stuck, causing no clock to be output from the divider. This will not clear until a power on reset occurs. This is only true if the divider is using divide by 2 (CGM_AC14_DC0[DIV] and/or CGM_AC9_DC0[DIV] = 0b1).

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

1. Configure the divider value to divide by 3 by setting CGM_AC14_DC0[DIV] and/or CGM_AC9_DC0[DIV] = 0b10, or any value not equal to 0b0 or 0b1

2. Wait for the divider update to complete, shown by the divider update status bit CGM_DIV_UPD_STAT[17]=0b0 and/or CGM_DIV_UPD_STAT[22]=0b0

3. Disable the divider by setting CGM_AC14_DC0[DE] and/or CGM_AC9_DC0[DE] = 0b0

4. Wait for CGM_DIV_UPD_STAT[17]=0b0 and/or CGM_DIV_UPD_STAT[22]=0b0

5. Change the clock selection required in the MCB_CLKOUT_SEL register

6. Wait for CGM_DIV_UPD_STAT[17]=0b0 and/or CGM_DIV_UPD_STAT[22]=0b0

7. Enable the divider by setting CGM_AC14_DC0[DE] and/or CGM_AC9_DC0[DE] = 0b1

8. Wait for CGM_DIV_UPD_STAT[17]=0b0 and/or CGM_DIV_UPD_STAT[22]=0b0

9. Configure the divider value to the desired division factor by setting CGM_AC14_DC0[DIV] and/or CGM_AC9_DC0[DIV] accordingly

10. Wait for CGM_DIV_UPD_STAT[17]=0b0 and/or CGM_DIV_UPD_STAT[22]=0b0

11. If usage of CGM_AC14_DC0[DIV] =0b1 and/or CGM_AC9_DC0[DIV]=0b1 is only required temporarily then it is advised to re-configure the divider after use to CGM_AC14_DC0[DIV] and/or CGM_AC9_DC0[DIV] = 0b10 or any value not equal to 0b0 or 0b1

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

If the clock monitoring unit (CMU) is being used to monitor CLK_OUT0 (via CMU_6) then this must also be disabled before CGM_AC14 is disabled during the steps above.

ERR011241: CGM: Device may become non-responsive if an LVD event occurs after changing the clock configuration

Description: If online STCU (Self Test Control Unit) BIST (Built in Self Test) is required to be run at 240MHz, then the clock configuration must be updated by changing the lower nibble of dcl_ips_0 register to b0100 to facilitate this. If a voltage glitch occurs on the 3.3V supply causing an LVD event on the VDD_HV_IO and VDD_HV_PMU domains after the dcl_ips_0 register is changed until the time when online BIST completes, then the device has a possibility to remain stuck in reset. The issue occurs only if the VREG_SEL pin is High (device is operating in internal regulation mode) and will require a power on reset to recover.

Workaround: There are several possible workarounds for this issue as follows:

1) Use the device in external regulation mode by keeping VREG_SEL low.

2) If online BIST is not required, then do not write dcl_ips_0

3) If online BIST will be run, write dcl_ips_0 immediately before running online to reduce the window of potential failure to the duration of online STCU i.e. just less than 40ms.

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

5) Write dcl_soc_conf_3 DCF in UTEST to a value of b1011 to elevate all LVD events to POR (power on reset).

Any of the above workarounds can be used.

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

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

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

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

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

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

Further, since the CMU monitored clocks are derived from one of the system clock sources (IRC_CLK, XOSC_CLK, PLL0_PHI_CLK, PLL1_PHI_CLK or SDPLL_CLK) if the loss of the monitored clock is caused by the loss of the source clock then this will be detected and reported to FCCU by existing source clock loss detection mechanisms.

CMU_5 is an exception because it is possible to source the monitored LFAST_CLK from external LFAST_REF_CLK input instead of a system clock source. In the externally provided LFAST_CLK use-case the loss of LFAST_REF_CLK can only be detected by loss of the LFAST frame transmit/receive functionality which leads to interruption in the LFAST communication.

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

Description: The CTE time base (datapath) clock is derived from the 80 MHz CTE module clock. It is possible to configure the time base clock frequency to an undivided 80 MHz by setting CTE Control Register 1 CTE Clock divider field to 0 or 1 (CTE_CNTRL1[CTECK_DV] = 0 or 1). If this is done and the Timing Table (TT) execution duration is set to 1 clock cycle by

programming TT0/TT1 Execution Duration Register to 1 (CTE_LUT_DUR = 1 or CTE_LUT_DUR1 = 1 depending on whether TT0 or TT1 is to be executed) then the CTE will not execute the timing table and no timing signals will be generated.

Workaround: To execute a timing table for 1 clock cycle the CTE time base clock frequency must be less than 80 MHz. This can be achieved by setting CTE_CNTRL1[CTECK_DV] field to any value greater than 1 which results in a divided CTE time base clock less than 80 MHz.

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

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

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

4 X IPG_CLOCK_PERIOD + 4 X DSPI_CLOCK_PERIOD + 0.5 x SCK_CLOCK_PERIOD

Where:

IPG_CLOCK is the internal bus clock (“system” clock)

DSPI_CLOCK is the protocol clock.

SCK_CLOCK is the Line-Side Serial Communication Clock.

Workaround: When operating in slave mode, software must ensure that the time interval between PCS assertion to start of SCK Clock is greater than 4 X IPG_CLOCK_PERIOD + 4 X DSPI_CLOCK_PERIOD + 0.5 x SCK_CLOCK_PERIOD.

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

ERR010538: End to End ECC: An uncorrectable ECC event from a slave to a master can cause invalid data reception

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

Case 1 Master DMA:

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

Case 2 Signal Processing Toolbox (SPT):

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

Workaround: Use one of the two following workarounds:

1. Use the NCF for Uncorrectable ECC Errors on all the slaves to determine that such a fault has occurred and take appropriate action. Depending on system usage this could be for example discarding of data, resetting an IP module or even resetting the whole device.

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

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

2. Clear Pending Read Enable bit for the masters

Signal Processing Toolbox – PCM_IAHB_BE5[PRE_SPT] - Set to 0

DMA – PCM_IAHB_BE1[PRE_DMA]= Set to 0

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

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

Description: If a PLL loss of lock (LOL) event occurs during the Direct Memory Access (DMA) RAM Memory BIST (MBIST) portion of the online STCU Self Test then the Self Test will be aborted and the Fault Collection and Control Unit (FCCU) will report a Non-Critical Fault NCF[7] (STCU NCF). This will be indicated by NCF[7] being set in the FCCU_NCF_S0 register. It has been found that in addition to this the FCCU may unexpectedly also report a DMA RAM Alarm fault. This will be indicated by NCF[106] being set in the FCCU_NCF_S3 register.

Workaround: If the application makes use of the DMA module and the DMA RAM Alarm NCF[106] occurs then the software reaction to this fault should first ensure that the alarm is genuine by ensuring that no STCU online self test was in operation and was aborted due to a PLL LOL event. This can be determined by checking the STCU2_ERR_STAT[LOCKESW] bit. If STCU2_ERR_STAT[LOCKESW] is not equal to 0b1 then the alarm is genuine and software can take appropriate action to this alarm, otherwise it can be cleared (by writing 0x400 to FCCU_NCF_S3) and ignored. If STCU online self test is not used in the application then no action need be taken and if NCF[106] occurs it can be considered genuine and be dealt with accordingly.

It should be noted that following a failed STCU online self test due to the conditions described the STCU2_ERR_STAT[LOCKESW] bit will remain set until a subsequent self test is run. This needs to be considered as if no action to the failing self test (e.g re-run self test) is taken then this would mean that a genuine NCF[106] occurrence could be interpreted as a false flag.

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

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

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

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

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

During the first phase, software should do the following:

1) move the FCCU to the CONFIG state

2) configure the FCCU including the error out protocol, but without setting the FCCU_CFG.FCCU_SET_AFTER_RESET flag to 0b1 (leave as 0b0)

3) exits to the NORMAL state

During the second phase, software should do the following:

4) move the FCCU to the CONFIG state

5) set the FCCU_CFG.FCCU_SET_AFTER_RESET flag to 0b1

6) exit to the NORMAL state

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

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

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

1. Reset is programmed as the only reaction for the fault.

2. Assertion of the fault coincides with the long/short functional reset reaction to a fault previously asserted.

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

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

Description: For certain combinations of the Flash Read Wait State Control (RWSC) and Address Pipeline Control (APC) settings in the Platform Flash Configuration Register (PFLASH_PCFR1) the Flash's array integrity (AI) check when run with breakpoints enabled may skip addresses resulting in an incorrect Multiple Input Signature Register (MISR) value or in the case of back to back ECC event errors (EER) or Single Bit Correction (SBC) events, a skipped breakpoint. This occurs for the following combinations:

RWSC=1 and APC=1

RWSC=3 and APC=2

RWSC=5 and APC=3

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

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

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

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

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

ERR007991: FLASH: Rapid Program or Erase Suspend fail status

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

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

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

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

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

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

1- A message is received and transferred to an MB (i.e. MBx)

2- MBx is locked by software for more than 20 CAN bit times (time determines the probability of erratum to manifest).

3- SMB0 (Serial Message Buffer 0) receives a message (i.e. message1) intended for MBx, but destination is locked by the software (as depicted in point 2 above) and therefore NOT transferred to MBx.

4- A subsequent incoming message (i.e. message2) is being loaded into SMB1 (as SMB0 is full) and is evaluated by the FlexCAN hardware as being for the FIFO.

5-During the message2, the MBx is unlocked. Then, the content of SMB0 is transferred to MBx and the CODE field is updated with an incorrect value.

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

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

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

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

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

1. Read the Control and Status word of that mailbox.

2. Check if the BUSY bit is deasserted, indicating that the mailbox is not locked. Repeat step 1) while it is asserted.

3. Read the contents of the mailbox.

4. Clear the proper flag in the IFLAG register.

5. Read the Free Running Timer register (TIMER) to unlock the mailbox

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

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

Description: When

a) the EXT_SYNC signal is selected to cause initialization by setting the Submodule 0 Control 2 Register FlexPWM_SUB0_CTRL2[INIT_SEL] = 11 and

b) a specific FAULTx input is associated with the submodule 0 outputs using the Submodule 0 Fault Disable Mapping Register (FlexPWM_SUB0_DISMAP) and

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

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

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

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

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

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

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

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

Description: The Data Transfer Mode bit (JTAGM_MCR[DTM]) is intended to control the ability of application software to write JTAG data to the JTAGM Data Output Registers (JTAGM_DORn). However JTAGM_DOR0,JTAGM_DOR1,JTAGM_DOR2 and JTAGM_DOR3 are writable regardless of the state of the JTAGM_MCR[DTM] bit.

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

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.

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

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

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

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

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

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

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

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

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

Workaround: The following three steps should be followed -

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

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

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

Note:

THeader_Nominal = 34 * TBit

TResponse_Nominal = 10 * (NData + 1) * TBit

THeader_Maximum = 1.4 * THeader_Nominal

TResponse_Maximum = 1.4 * TResponse_Nominal

TFrame_Maximum = THeader_Maximum + TResponse_Maximum

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

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

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

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

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

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

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

ERR008340: NPC: EVTO_B toggles instead of remaining asserted when used by the DTS if Nexus is not enabled

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

Workaround: Always set the MCKO_EN bit to 1 when using the DTS trigger out function.

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

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

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

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

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

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

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

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

On devices that bypass the Boot Assist Flash (BAF) or Boot Assist Module (BAM) after reset (in other words, the System Status and Configuration Module [SSCM] boots directly to user code if a valid Reset Configuration Half-Word is found), perform an indirect branch instruction shortly after reset to reset the ICNT (and HIST if Branch History mode is enabled). A full program trace synchronization message will be generated after 256 direct branches even if there is no indirect branches. This will allow the tool to recover the trace reconstruction from that point onward.

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

ERR010638: NZxC3: Nexus messaging becomes corrupted if more than one master (including a device core) is active and the core is subsequently disabled.

Description: This errata applies to the condition where there is more than one master active on the Nexus Port Controller (NPC) module, and one or more of these masters is a device core. In this situation, if a mode transition is initiated to a mode where that core is disabled, with the clock gated (as configured in the relevant core control register MC_ME_CCTLx for the requested mode) then the Nexus3 interface issues a core debug status message indicating the transition into core STOP mode (core disabled with clock gated). It is possible during this message

transmission, that the core Nexus clock gets disabled. This blocks the Nexus interface, and the message data can be left pending on the interface until the core clock resumes. This causes overflow of the NAL (Nexus Aurora Link) FIFO and corrupts the Aurora protocol.

Workaround: Possible workarounds are:-

1) If Nexus traces are enabled then mode transitions that disable device cores should not be executed.

2) If Nexus traces are enabled and a core is needed to be stopped, then that core should use PowerPC ‘wait' instruction instead of stop mode.

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

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

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

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

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

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

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

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

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

ERR050130: PIT: Temporary incorrect value reported in LMTR64H register in lifetimer mode

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

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

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

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

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

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

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

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

Workaround: To prevent the issue from occurring, VREG_POR_B should be asserted anytime the VDD_HV_IO and VDD_HV_PMU supplies are below the operating range stated in the device data sheet (this is already a requirement for external regulation mode).

ERR011156: PMC: During online PMC self-test if RESET_B is asserted then a spurious destructive or power on reset may occur and be reported as an LVD/HVD event in the RGM_DES register.

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

1. PMC module clock configured to be less than 16MHz. In this case, when reset occurs, the CGM clock divider gets bypassed and the PMC module clock changes to 16MHz, and this causes PMC self-test timing window to reduce, which in turn causes failures and a resulting reset (destructive or POR) to occur.

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

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

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

1. PMC module clock frequency is greater than or equal to 16MHz or if the PMC module clock is configured to be less than 16MHz then programming the PMC_STTW register with a value of 0x172 (corresponding to 16MHz + 10% value) then no issue will occur.

2. PCS ramp-down is disabled on the selected clock source for the PMC module.

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

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

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

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

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

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

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

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

a) Optionally during normal functional mode using the PMC_PMCCR register.

b) During online and offline STCU self-test, by default all LVDs and HVDs are masked except VDD_HV_IO LVD using the DCL_IPS_0 control in UTEST.

c) During PMC self test the LVD/HVD for the supply under PMC self test and the subsequent supply to be under self test are automatically masked

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

1) If LVD occurs on VDD_HV_PMC while it is masked: During online STCU, PMC self-test or normal operation, reset will not be masked and a reset event (corresponding to External Reset) would occur. During offline STCU operation where RESET_B is already asserted, LBIST 4 partition will fail self-test and the FCCU and MC_RGM status will not reflect the LVD event.

2) If LVD occurs on VDD_HV_IO while it is masked: The 16MHz IRCOSC will stop working and its output may glitch, but will restart when supply recovers above LVD level. This could lead to the device becoming stuck in reset.

3) If LVD occurs on VDD_HV_ADC while it is masked: The SAR-ADCs will stop converting, but recommence when supply recovers above LVD level. The ADC cannot be used during STCU execution but should not be relied upon during PMC self-test execution or if PMCCR[LVD_ADC_ENABLE] has been written to 0b0 and the supply can fall below operating level.

4) If LVD occurs on VDD_HV_FLA while it is masked: flash reads and write/erase operations can be impacted.

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

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.

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

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

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

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

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

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

Workaround: There are two possible workarounds:

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

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

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

Description: When the SPT completes execution of a command sequence there is a short time window in which access to an SPT work register by CPU can potentially cause peripheral bus 1 (PBRIDGE_1/AIPS_1) to become non-responsive and the CPU to stop executing instructions because it has stalled waiting on the peripheral bus transaction to complete. Any other core that attempts to access peripheral memory over peripheral bus 1 will also stall and the system can only be recovered by asserting power on reset (POR).

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

Practically, the only way this can occur is if the CPU polls for assertion of SPT_CS_STATUS0[PS_STOP] or SPT_CS_STATUS0[PS_ASYNCSTOP] then immediately makes repeated work register accesses so that one of these accesses coincides with SPT ‘soft reset’ signal assertion. Access to the SPT status registers cannot cause the failure.

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

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

The possible worst-case duration can be calculated:

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

Workaround: An interrupt service routine (ISR) should be used to react to SPT command sequence completion because the latency between the interrupt request being raised on assertion of CS_STATUS0[PS_STOP] and execution of the first instruction in the ISR is greater than the worst-case time window duration for assumed configuration. However if CPU polling must be used then measures must be taken before safely accessing the work registers upon completion of the command sequence:

• Check CS_STATUS0[PS_STOP] = 0b1 (This bit indicates that SPT state machine has been in STOP state and so has executed the STOP command.)

• Once CS_STATUS0[PS_STOP] assertion has been detected the user must clear it by writing 0b1.

• Check CS_STATUS3[CS_DMA_ON] = 0b0 (Ensures that all Command Sequencer fetches have been completed.)

Alternatively, enforcing a time delay greater than the worst-case window duration between execution of STOP command and accessing a work register would ensure the time window has elapsed. The delay must begin from the detection of CS_STATUS0[STOP] = 0b1.

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

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

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

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

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

Workaround: The CPU can re-initialize the WRs after STOP command has concluded SPT command execution by writing any value to the WRs. The CPU should not rely on read access to the work registers (WR) between a STOP command and the launching of a new command

sequence, instead the WAIT and EVT commands should be used to manage the timing of accesses when both SPT and CPU must interact with work registers during command sequence execution.

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

Description: While an online self-test is in progress there is a finite window during the self-test execution during which if an external reset is asserted (RESET_B pulled low) and this reset is configured as short sequence for external reset by setting the Short Sequence for External Reset bit in the Reset Generation Module Functional Event Short Sequence Register (RGM_FESS[SS_EXR] = 1b1), or if another functional reset source is triggered during this window, the time after which the part waits for self-test to complete is longer than expected. This time-out value is governed by the watchdog time-out value set in the Watchdog End of Count Timer field in the STCU2 Watchdog Register Granularity register (STCU2_WDG[WDGEOC]). Further, the self-test does not issue a hardware abort (STCU2 Error Register On-line Hardware Abort Flag (STCU2_ERR_STAT [ABORTHW]) will not be set to 1b1) but signals a time-out (STCU2_ERR_STAT [WDTOSW] = 1b1). If the online self-test is being run with PLL enabled then an unexpected PLL unlock event is also observed (STCU2_ERR_STAT[LOCKESW] = 1b1)

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

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

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

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

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support