Mask Set Errata for Mask 1N81M

This report applies to mask 1N81M for these products:

- MPC5748G
- MPC5747G
- MPC5746G
- MPC5748C
- MPC5747C

Mask Specific Information

JTAG identifier	0x0988101D

Table 1. Errata and Information Summary

Erratum ID	Erratum Title
e9321	ADC: Conversions may fail if Pre-Sampling is enabled
e8714	ADC: Conversions on an open channel with the presampling feature enabled do not return the expected results
e8804	ADC: Setting the DMA request to be cleared on read of data registers does not work
e9028	C55FMC: Possibility of a false Address Encode Error during flash read-while-write operation
e10143	CMP_0: At exit from STANDBY the output of Comparator 0 is configured to high-impedance
e9996	DSPI0 and DSPI1: Frame transfer does not restart after DSI frame matches preprogrammed value
e9995	DSPI0 and DSPI1: Frame transfer does not restart in case of DSI parity error in master mode
e10215	DSPI0/1 and SPI0/3 : Frame transfer does not restart in case of SPI parity error in master mode
e10452	eDMA: When master ID replication is enabled, the stored ID and privilege level will change if read by another master.
e9978	eMIOS: Unexpected channel flag assertion during GPIO to MCB mode transition
e9328	ENET: Not possible to set the entire range of the ENET Timer Compare Capture Register (ENET_TCCR0[TCC] and ENET_TCCR1[TCC])
e7885	ENET: Potential sequencing issue with TDAR in Multi-Queue mode
e10594	ENET: The ENET1 (Ethernet) module does not function unless the Peripheral control register (MC_ME_PCTL6) is enabled .
e11046	ENET: The timer mode ENETx_TCSRn[TMODE] settings 1010 and 1001 are unusable
e8042	FCCU: EOUT signals are active, even when error out signaling is disabled
e9076	FCCU: Fault Collection and Control Unit glitch filter behavior is indeterministic

Table continues on the next page...

Table 1. Errata and Information Summary (continued)

Erratum ID	Erratum Title
e8901	Flash: Flash internal regulation mode may lead to power-on-reset when in STANDBY and LPU modes
e7991	FLASH: Rapid Program or Erase Suspend fail status
e8759	FlexCAN: FD frame format not compliant to the new ISO/CD 11898-1: 2014-12-11
e10595	FlexCAN: FLEXCAN1-7 modules will not work unless the Fast External Oscillator (FXOSC) clock source is enabled
e10368	FlexCAN: Transition of the CAN FD operation enable bit may lead FlexCAN logic to an inconsistent state.
e10620	FlexRay: The FS80 clock source should not be selected for the FlexRay protocol clock when the MCU clocking is configured for Linear Dynamic Frequency Scaling
e8770	FlexRAY: Missing TX frames on Channel B when in dual channel mode and Channel A is disabled
e10413	HSM: HSM RAM initialization clock out of specification when Accelerated Low Power Exit is enabled with FMPLL = 160MHz
e8883	HSM: Input Output Control enables pad and alternative pad
e8180	HSM: e200z0 Nexus interface DQTAG implemented as variable length field in DQM message
e10118	HSM: TRNG can only select FIRC for clock source
e9335	IAHB: Default programming of Intelligent AHB Gasket pending read optimisation can lead to masters stalling or receiving incorrect or spurious data
e8938	LINFlexD: Corruption of Tx data in LIN mode with DMA feature enabled (applicable to LIN1)
e8933	LINFlexD: Inconsistent sync field may cause an incorrect baud rate and the Sync Field Error Flag may not be set
e8080	LINFlexD: TX pin gets set to High-Z when in IDLE state
e8939	LINFlexD: Tx through DMA can be re-triggered after abort in LIN/UART modes or can prematurely end on the event of bit error with LINCR2[IOBE] bit being set in LIN mode (applicable only for LIN1)
e10141	LPU: LPU_RUN mode system clock must be preconfigured for undivided FIRC prior to LPU_STANDBY entry
e10132	LPU: Mode transition to LPU_STOP or LPU_STANDBY may not complete
e10609	MC_CGM: CLKOUT_0 and CLKOUT_1 may stop if the clock selection is changed when configured for divide by 2
e10440	MC_ME & LPU: The transition between DRUN/RUN mode to STANDBY or DRUN/RUN mode to LPU_RUN may not complete if a reset is asserted.
e10361	MC_ME & LPU: The transition between DRUN/RUN mode to STANDBY or DRUN/RUN mode to LPU_RUN may not complete if EXR is asserted.
e10362	MC_ME & LPU: The transition between DRUN/RUN mode to STANDBY or DRUN/RUN mode to LPU_RUN may not complete if any LVD is asserted or PORST goes low
e9200	MC_ME and LPU: JTAG TCK pin must be configured to ensure successful exit from STANDBY and LPU modes
e10323	MC_ME: The transition from DRUN/RUN mode to STANDBY will not complete if a wake-up is triggered in a 50nS window.
e8871	ME: In STANDBY/LPU STANDBY modes, if FIRC is disabled all 256kB of STANDBY RAM is retained
e8898	ME: For a supply voltage of greater than 5.3V the MCU may be reset during a LPU_STANDBY to LPU_RUN mode transition
e8880	MEMU: After exit from LPU RUN mode, the MEMU_PROT registers are uninitialized which may impact the usage of MEMU

Table continues on the next page...

Table 1. Errata and Information Summary (continued)

Erratum ID	Erratum Title
e8870	NEXUS: Mutli core tracing using NEXUS3 gives corrupted traces for the case when the cores are in the ratio of 2:1
e9135	NPC: LPM Debug handshake does not work for STOP Mode using all TCK frequencies
e10603	NPC: Nexus Port Controller (NPC) must be enabled to allow mode changes during debug
e10723	NPC: Repeated Nexus3 Debug Status messages can be observed if more than one master (including a device core) is active and the core is subsequently disabled
e10340	NZxC3: ICNT and HIST fields of a Nexus message are not properly reset following a device reset
e10396	PASS: Password challenge to PASS fails while program erase ongoing in any block in memory partition 0
e9873	PFLASH: Calibration remap to flash memory not supported on 16KB and 32KB flash blocks in address range 0x00F90000-0x00FBFFFF
e10789	PFLASH: EEPROM ECC error suppression is not supported on 16KB and 32KB flash blocks in the address range 0x00F90000-0x00FBFFFF
e11096	SAI: Internal bit clock is not generated when RCR2[BCI]=1 or TCR2[BCI]=1
e11150	SAI: Internally generated receive or transmit BCLK cannot be re-enabled if it is first disabled when RCR2[DIV] or TCR2[DIV] > 0
e8406	SMPU: Process Identifier region hit determination is not available in debug mode
e10103	STCU2: Unexpected STCU self-test timeout can occur when a short functional reset is triggered during execution of online self-test
e8902	STM: The STM Counter Register will not report count value when TEN is cleared
e9322	TDM: Erase of TDR flash block may be blocked by the TDM
e8868	WKPU: Wakeup Pads pull-ups cannot be enabled in Standby Mode

Table 2. Revision History

Revision	Changes
1	Initial revision
2 June 2016	The following errata were removed. • e8179 The following errata were added. • e10143 • e10440 • e10323 • e10141 • e9978 • e9996 • e9995 • e10103 • e10362 • e10361 • e10368 • e10132

Table continues on the next page...

Table 2. Revision History (continued)

Revision	Changes
	• e9873 • e10118 • e10214 The following errata were revised. • e8933 • e9200
3, January 2018	The following errata were removed. • e10214 The following errata were added. • e10723 • e11096 • e10340 • e10620 • e11046 • e10452 • e10413 • e10396 • e10215 • e10594 • e10595 • e10609 • e10603 • e8804 • e9135 • e9322 • e9028 • e10789 • e11150 The following errata were revised. • e10143

e9321: ADC: Conversions may fail if Pre-Sampling is enabled

Description: Analog to Digital conversions for ADC_0 and ADC_1 may fail if Pre-Sampling is enabled. In this case, the ADC output may be unreliable. The failure occurs at minimum sampling time and when the internal voltage sample selection (ADC_x_PSCR[PREVALn] = 0, 1 or 2) is configured for VSS_HV_ADCx, VDD_HV_ADCx/8 or VREFLx as pre-sample voltage (x = 0 or 1).

Workaround: For ADC_0 select one of the following workarounds

- Disable pre-sampling
- If pre-sampling is enabled increase sampling time to at least 375nS

For ADC_1 select one of the following workarounds

- Disable pre-sampling

- If pre-sampling is enabled select pre-sample voltage ADC_1_PSCR[PREVALn] = 3 (VDD_HV_ADC1_REF)
- If pre-sampling is enabled increase sampling time to at least 375nS

e8714: ADC: Conversions on an open channel with the presampling feature enabled do not return the expected results

Description: The analog-to-digital converter (ADC) presampling feature will precharge an ADC channel sample capacitor to an internal voltage. The internal voltage is selected by the Internal Voltage Selection for Presamping bit field (PREVAL0) of the ADC's Presampling Control Register (ADC_PSCR). The user has either the option of sampling an ADC reference voltage rail (VDD) or a ground (VSS). If the convert presampled value bit (PRECONV) of the ADC_PSCR register is cleared (ADC_PSCR[PRECONV]=0b0) then the presampling stage is followed by the sampling of the ADC channel input and then the conversion is performed. If the user has selected VSS as the internal sample voltage when this is done on an open or unconnected channel, the conversion result will be closer to 1000 when it is expected to be 0. If the user has selected VDD as the internal voltage the conversion result will be closer to 2000 rather then 4095. If ADC_PSCR[PRECONV]=0b1 then sampling of the ADC channel input is bypassed and the presampled voltage is converted directly. This conversion result is close to the expected value for the presampled voltage.

Workaround: Do not expect conversion result to be close to zero or full-scale on an open channel with presampling enabled and ADC_PSCR[PRECONV] = 0b0.

e8804: ADC: Setting the DMA request to be cleared on read of data registers does not work

Description: The Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC) can generate DMA requests to the DMA controller. If the DMA Clear sequence enable bit in the ADC DMA Enable register (ADC_DMAE[DCLR]) is set to 0b1 the DMA request should be cleared upon a read of the data registers but it is cleared automatically without a read.

Workaround: Do not use the ADC module with the DMA request cleared on read of data registers enabled, ADC_DMAE[DCLR]=0b1. Instead configure ADC_DMAE[DCLR]=0b0. With DMA enabled, the request will only be cleared once the DMA controller acknowledges it and accesses the conversion data register.

e9028: C55FMC: Possibility of a false Address Encode Error during flash read-while-write operation

Description: The Address Encode Error (AEE) logic compares the system bus access address (external to flash) with the flash array address (internal to flash) and under normal operation they should match. During flash read-while-write operations (RWW), it is possible for a Program or Erase operation to corrupt the Address Encode feature of the flash, and falsely give an AEE event. The false AEE event only occurs for RWW operations to partitions in the Low, Mid or High address spaces, and will only occur if the read and write occur both in Even RWW partitions (i.e. 0, 2, 4), or if the read and write occur both in Odd RWW partitions (i.e. 1, 3, 5).

The AEE event is mapped to FCCU non-critical fault 38 (NCF[38]) and the default configuration is no reaction. Note it is possible to configure a long or short reset to react to an AEE event.

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

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

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

Workaround: The user should be aware that a false AEE event could be flagged for the erratum condition. The default state of the MCU will not react to such an AEE event so the user need only manage the event if,

(A) the application software is reading and handling the C55FMC_MCR[AEE] flag, or

(B) the application software configures the FCCU to react to such an AEE event.

e10143: CMP_0: At exit from STANDBY the output of Comparator 0 is configured to high-impedance

Description: Irrespective of the configuration of GPR_CTL[CMP0_STDBY] the output of the Comparator 0 (CMP0_O) will be configured to high-impedance when the MCU exits STANDBY. As a result the Comparator output value is not valid until the CMP_0 is reconfigured in the subsequent RUN mode.

Workaround: To retain the Comparator output value during STANDBY mode and during the STANDBY exit sequence to DRUN, the Pad Keeper function can be enabled at PMCDIG_RDCR[PAD_KEEP_EN].

For the case the Pad Keeper is enabled:

(1) The Comparator 0 output will be static whilst in STANDBY mode.

(2) The Pad Keeper function affects all outputs in the STBY_LPU power segment.

e9996: DSPI0 and DSPI1: Frame transfer does not restart after DSI frame matches preprogrammed value

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

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

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

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

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

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

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

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

e9995: DSPI0 and DSPI1: Frame transfer does not restart in case of DSI parity error in master mode

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

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

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

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

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

5. Parity error is detected on received frame

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

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

e10215: DSPI0/1 and SPI0/3 : Frame transfer does not restart in case of SPI parity error in master mode

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

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

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

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

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

5. Parity error is detected on received frame.

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

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

e10452: eDMA: When master ID replication is enabled, the stored ID and privilege level will change if read by another master.

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

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

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

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

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

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

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

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

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

e9328: ENET: Not possible to set the entire range of the ENET Timer Compare Capture Register (ENET_TCCR0[TCC] and ENET_TCCR1[TCC])

Description: It is not possible to set the ENET Timer Capture Compare range ENET_TCCRn[TCC] to zero or close to zero.

n=0 or 1

Workaround: The range set for the ENET_TCCRn[TCC] needs to be restricted to the following range

ENET_ATINC[INC] ≤ TCC ≤ (ENET_ATPER[PERIOD] - ENET_ATINC[INC])

Note: If restriction is not followed the associated event/interrupt my not be generated

e7885: ENET: Potential sequencing issue with TDAR in Multi-Queue mode

Description: When the 10/100-Mbps Ethernet Media Access Control (ENET MAC) module is in Multi-queue mode, there is a potential sequencing issue between the module clearing the ENET Transmit Descriptor Active Register (ENET_TDARn_TDAR) bit and the software setting it. This can cause the module to hang.

Workaround: ENET_TDARn_TDAR should be set by software after it is cleared by the ENET. This is achieved by introducing a short delay after a new Transmit Buffer Descriptor (TxBD) is prepared and written into a designated memory.

- Software prepares a new TxBD and stores/writes it into a designated memory
- Software introduces a delay by reading the relevant ENET_TDARn_TDAR 4 times as shown by the following pseudo-code :

```
For (i=0; i<4; i++) // 4 Reads should be sufficient
{
//Read TDAR
If (TDAR == 0)
{
tdar_trigger = 1
exit_for_loop
}
else
tdar_trigger = 0
end
}
If (tdar_trigger)
Set TDAR = 1 (i.e., set ENET_TDARn_TDAR to 1)
Else
Do_nothing (i.e., don’t trigger TDAR)
End
```

Therefore the software can set the TDAR bit as soon as it is detected as zero.

e10594: ENET: The ENET1 (Ethernet) module does not function unless the Peripheral control register (MC_ME_PCTL6) is enabled .

Description: The Ethernet module ENET1 does not function unless the MLB Peripheral control register (MC_ME_PCTL6) is enabled. This does not apply to ENET0.

Workaround: Enable MC_ME_PCTL6 if using the ENET1 module.

e11046: ENET: The timer mode ENETx_TCSRn[TMODE] settings 1010 and 1001 are unusable

Description: The following timer modes are unavailable in ENET0_TCSRn[TMODE] and ENET1_TCSRn[TMODE]:

1010: Timer Channel is configured for Output Compare – clear output on compare, set output on overflow.

1001: Timer Channel is configured for Output Compare – set output on compare, clear output on overflow.

Workaround: The user must refrain from using these timer modes when using ENET0 and ENET1, and instead use another available mode.

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

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

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

Workaround: There are three possible workarounds:

1) Enable EOUT signaling for all enabled error sources.

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

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

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

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

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

d1) Validation is performed while EOUT is asserted

d2) Timeout occurs but no validation and EOUT is still asserted.

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

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

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

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

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

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

e9076: FCCU: Fault Collection and Control Unit glitch filter behavior is indeterministic

Description: The FCCU Error In (EIN) signal is asynchronous to the FCCU glitch filter and as there is no synchronisation logic between these domains there is a possibility of metastability leading to indeterministic filter behaviour.

Workaround: As the behavior of the FCCU glitch filter may be unpredictable it must be bypassed at FCCU_CTRL[FILTER_BYPASS]. If the EIN signal is required an external glitch filter should be implemented.

e8901: Flash: Flash internal regulation mode may lead to power-on-reset when in STANDBY and LPU modes

Description: For the case when a long functional reset or destructive reset is asserted in STANDBY and LPU modes this may lead to a POR (power-on-reset). This issue is only observed when the flash is configured for internal regulation mode. When the flash is supplied externally this issue is not observed.

Workaround: In the event of a reset the application software has to restart so the POR should not add to this overhead.

e7991: FLASH: Rapid Program or Erase Suspend fail status

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

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

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

e8759: FlexCAN: FD frame format not compliant to the new ISO/CD 11898-1: 2014-12-11

Description: This version of the device implements a Flexible Controller Area Network (FlexCAN) module version that implements a Flexible Data (CAN-FD) frame format according to ISO/WD 11898-1: 2013-12-13. However, it is not compliant with the new ISO/CD 11898-1: 2014-12-11 format. The frame format was updated during the ISO standardization process.

The limitations are the following:

- the FD frame format is incompatible, the Cyclic Redundancy Check [CRC] does not include the added stuff bit count field
- the FD CRC computation is incompatible, a different seed value is used.

As a consequence this device is not suitable for use in CAN-FD networks that use the new FD frame format according to ISO/CD 11898-1: 2014-12-11.

FlexCAN3 with CAN FD feature enabled is affected by this defect.

Workaround: Use CAN-FD mode in networks that only includes devices that conform to the ISO/WD 11898-1: 2013-12-13 frame format.

The Classic CAN mode is unaffected and can be used without restrictions.

e10595: FlexCAN: FLEXCAN1-7 modules will not work unless the Fast External Oscillator (FXOSC) clock source is enabled

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

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

e10368: FlexCAN: Transition of the CAN FD operation enable bit may lead FlexCAN logic to an inconsistent state.

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

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

1. If FlexCAN is already in freeze mode, go to step 3, otherwise set the HALT and FRZ bits of the CAN_MCR register.

2. Wait the FRZACK bit of the CAN_MCR register to be set by the hardware.

3. Set the LPB (Loop Back Mode) bit of the CAN_CTRL1 register.

4. Configure only one message buffer to be transmitted. The frame should be a classical one (non-FD) with IDE =0, RTR =1 DLC =0x5 and STD_ID =0x682.

5. Set the FDEN bit of the CAN_MCR register.

6. Clear the HALT bit of the MCR register to leave freeze mode.

7. Wait the FRZACK bit of the CAN_MCR register to be cleared by the hardware.

8. Wait the respective bit of the CAN_IFLAG register to be set (successfully transmission in loop back mode).

9. Clear the respective bit of the CAN_IFLAG register by writing 1.

10. Set the HALT and FRZ bits of the CAN_MCR register.

11. Wait the FRZACK bit of the CAN_MCR register to be set by the hardware.

12. Clear the LPB (Loop Back Mode) bit of the CAN_CTRL1 register.

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

1. If FlexCAN is already in freeze mode, go to step 3, otherwise set the HALT and FRZ bits of the CAN_MCR register.

2. Wait the FRZACK bit of the CAN_MCR register to be set by the hardware.

3. Set the SOFTRST bit of the CAN_MCR register.

4. Wait the SOFTRST bit of the CAN_MCR register to be cleared by the hardware.

5. Set again the SOFTRST bit of the CAN_MCR register.

6. Wait the SOFTRST bit of the CAN_MCR register to be cleared by the hardware.

e10620: FlexRay: The FS80 clock source should not be selected for the FlexRay protocol clock when the MCU clocking is configured for Linear Dynamic Frequency Scaling

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

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

e8770: FlexRAY: Missing TX frames on Channel B when in dual channel mode and Channel A is disabled

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

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

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

e10413: HSM: HSM RAM initialization clock out of specification when Accelerated Low Power Exit is enabled with FMPLL = 160MHz

Description: The HSM RAM will be clocked greater than the 80MHz maximum specification for the following configuration

- The HSM RAM initialization is enabled, and
- Accelerated Low Power Exit is configured to use the FMPLL at 160MHz

At the exit from low power mode prior to DRUN mode entry, the FMPLL will be preconfigured to run at 160MHz. In this phase prior to the DRUN mode the HSM RAM will be initialized with the 160MHz clock. This may result in the incorrect initialization of the HSM RAM and thus the reporting of ECC errors.

Workaround: For the erratum configuration descried the HSM RAM clock specification can be adhered to if the FMPLL is preconfigured for 80MHz for the Accelerated Low Power Exit. Note the FMPLL must be configured in a RUN mode prior to the low power mode entry.

e8883: HSM: Input Output Control enables pad and alternative pad

Description: There is no way to select between the 2 ports available for each of the functions HSM_DO0 and HSM_DO1

HSM_IOCTL[DO_EN0] will enable output on PD[12] and PI[6]

HSM_IOCTL[DO_EN1] will enable output on PB[12] and PI[7]

Workaround: When enabling the HSM IO function the user must consider that each HSM output will require the allocation of 2 ports.

e8180: HSM: e200z0 Nexus interface DQTAG implemented as variable length field in DQM message

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

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

e10118: HSM: TRNG can only select FIRC for clock source

Description: The user must not select FXOSC as a clock source for the HSM TRNG. The HSM TRNG clock source is selected at MC_CGM_AC3_SC[SELCTL].

Workaround: The user should select the FIRC for the HSM TRNG clock by clearing MC_CGM_AC3_SC[SELCTL] (this is the default setting

e9335: IAHB: Default programming of Intelligent AHB Gasket pending read optimisation can lead to masters stalling or receiving incorrect or spurious data

Description: The eDMA, ENET_0/1, uSDHC, MLB, USB_0/1 and e200z2 masters can stall, receive wrong read data or get a spurious read access when the masters initiates a back-to-back read accesses and the first access is found to have an uncorrectable End-to-end Error Correction Code (e2eECC). This occurs when the pending read enabled optimization in the Intelligent AHB (IAHB) is enabled at the Bus Bridge Configuration Register PCM_IAHB_BEx[PRE_y], which is the default configuration out of reset.

Workaround: The following workaround are available for each master.

DMA:

1) Clear Pending Read Enable bit for the eDMA (PCM_IAHB_BE0[PRE_DMA]=0). Note: This workaround could have a small impact on performance

2) Set the Halt On Error bit DMA_CR[HOE] within the DMA. With this the bus error on the first access will be correctly handled, the DMA will disregard the second access and ignore all further DMA requests.

ENET0 and 1:

1) Clear Pending Read Enable bit for ENET_0 (PCM_IAHB_BE0[PRE_ENET]=0) and ENET_1 (PCM_IAHB_BE1[PRE_MLB]=0).

Note: This workaround could have a small impact on performance

uSDHC:

1) Clear Pending Read Enable bit for uSDHC (PCM_IAHB_BE1[PRE_uSDHC]=0).

Note: This workaround could have a small impact on performance

MLB:

1) Clear Pending Read Enable bit for MLB (PCM_IAHB_BE1[PRE_MLB]=0).

Note: This workaround could have a small impact on performance

USB_0 and USB_1:

1) Clear Pending Read Enable bit for USB_0 (PCM_IAHB_BE1[PRE_USB0]=0) and USB_1 (PCM_IAHB_BE1[PRE_USB1]=0).

Note: This workaround could have a small impact on performance

e200Z2:

1) Clear Pending Read Enable bit for Z2_INST (PCM_IAHB_BE2[PRE_Z2_INST]=0) and Z2_DATA (PCM_IAHB_BE2[PRE_Z2_DATA]=0).

Note: This workaround could have a small impact on performance

e8938: LINFlexD: Corruption of Tx data in LIN mode with DMA feature enabled (applicable to LIN1)

Description: The LINFlexD module is driven by two different clocks. The transmit/reception logic is controlled by the module clock (LIN_CLK) and register accesses are controlled by the peripheral bus clock (PBRIDGEx_CLK). In LIN mode, the re-synchronization of the “Idle on bit error” between the two clocks may cause the Direct Memory Access (DMA) Finite State Machine inside the LINFlexD module to move to the idle state while a transmission is in process. This unwanted idle state transition could lead trigger a new DMA request, potentially overwriting the Buffer Identifier Register (BIDR) and the Buffer Data Registers (BDRL and BDRM).

Workaround: Do not enable the “Idle on bit error” of LIN Control Register 2 (ILINCR2[IOBE] = 0). Instead of using the “Idle on bit error", use the bit error interrupt of LIN Interrupt Enable Register (LINIER[BEIE] = 1) to trigger an Interrupt service routine and force the LIN into idle mode through software if needed.

e8933: LINFlexD: Inconsistent sync field may cause an incorrect baud rate and the Sync Field Error Flag may not be set

Description: When the LINFlexD module is configured as follows:

1. LIN (Local Interconnect Network) slave mode is enabled by clearing the Master Mode Enable bit in the LIN Control Register 1 (LINCR1[MME] = 0b0)

2. Auto synchronization is enabled by setting LIN Auto Synchronization Enable (LINCR1[LASE] = 0b1)

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

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

Workaround: There are 2 possible workarounds.

Workaround 1:

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

1. Set the LIN state Interrupt enable bit in the LIN Interrupt Enable register (LINIER[LSIE] = 0b1)

2. When the Data Reception Completed Flag is asserted in the LIN Status Register (LINSR[DRF] = 0b1) read the LIN State field (LINSR[LINS])

3. If LINSR[LINS]= 0b0101, read the Counter Value field of the LIN Time-Out Control Status register (LINTCSR[CNT]), otherwise repeat step 2

4. If LINTCSR[CNT] is greater than 0xA, discard the frame.

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

1. Set the LIN State Interrupt Enable bit in the LIN Interrupt Enable register (LINIER[LSIE])

2. When the Data Reception Completed flag bit is asserted in the LIN Status Register (LINSR[DRF] = 0b1), read the LINSR[LINS] field

3. If LINSR[LINS]= 0b0101, store LINTCSR[CNT] value in a variable (ValueA), otherwise repeat step 2

4. Clear LINSR[DRF] flag by writing LINSR[LINS] field with 0xF

5. Wait for LINSR[DRF] to become asserted again and read LINSR[LINS] field

6. If LINSR[LINS] = 0b0101, store LINTCSR[CNT] value in a variable (ValueB), else repeat step 4

7. If ValueB – ValueA is greater than 0xA, discard the frame

Workaround 2:

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

e8080: LINFlexD: TX pin gets set to High-Z when in IDLE state

Description: LINFlex drives the buffer enable signal for it's transmit pin output (TX) to be '0' after transmitting the LIN frame. This causes the TX line to go to High-Z which will be an issue if the associated LIN transceiver has an internal "pull down".

Issue will also occur when module is configured in UART mode with the TX output pin becoming High-Z when idle.

Workaround: When operating in LIN mode, use a LIN transceiver with internal "pull up". If the transceiver has an internal "pull down", add an external "pull up".

When operating in UART mode, the issue can be worked around by enabling the internal pull up on the TX pin using the corresponding SIU_MSCR register.

e8939: LINFlexD: Tx through DMA can be re-triggered after abort in LIN/UART modes or can prematurely end on the event of bit error with LINCR2[IOBE] bit being set in LIN mode (applicable only for LIN1)

Description: The LINFlexD module is driven by two different clocks. The transmit/reception logic is controlled by the module clock (LIN_CLK) and register accesses are controlled by the peripheral bus clock (PBRIDGEx_CLK). Due to possible synchronization issue between the two clock domains, there is a possibility that DMA transmission get stuck due to DMA Finite State Machine doesn't go into idle. This may occur in one of the following conditions:

- if an abort request is triggered (LINCR2[ABRQ]=1) in LIN or UART modes
- if idle on bit error feature is enabled (LINCR2[IOBE]=1) in LIN mode, and a bit error occurs.

DMA state machine will not generate any transaction, waiting for data transmission flag LINSR[DTF] to be set which will never occur.

Workaround: If DMA is used:

- Bit error interrupt should be enabled through LINEIER[BEIE]. When an bit error interrupt is triggered, the interrupt service routine must either reset the DMA Tx channel enable (DMATXE) and the DMA Rx channel enable (DMARXE) registers
- if an abort is requested (LINCR2[ABRQ]=1) in LIN/UART mode, either reset DMATXE/ DMARXE of LINFlexD after writing LINCR2 [ABRQ]

e10141: LPU: LPU_RUN mode system clock must be preconfigured for undivided FIRC prior to LPU_STANDBY entry

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

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

Workaround: Prior to entering LPU_STANDBY select undivided FIRC as the LPU System Clock by configuring LPU_RUN_CF[SYS_CLK_SEL] = 0 and FIRC_CTL[FIRCDIV] = 5'b0.

e10132: LPU: Mode transition to LPU_STOP or LPU_STANDBY may not complete

Description: A mode transition from LPU_RUN to LPU_STOP or LPU_RUN to LPU_STANDBY may not complete if a wake-up or interrupt is received in a 5 FIRC clock window after the mode transition is requested. This is only applicable if the FIRC is disabled in LPU_STOP and LPU_STANDBY. In this scenario, the z2 core is stopped and if the System Watchdog Timer (SWT) is enabled, the SWT continues to run, the SWT will timeout and a SWT destructive reset will be triggered.

Workaround: The user can select one of the following workarounds:

1. Enable the SWT during LPU modes to enable recovery through destructive reset
2. Enable the FIRC in LPU_STOP and LPU_STANDBY

e10609: MC_CGM: CLKOUT_0 and CLKOUT_1 may stop if the clock selection is changed when configured for divide by 2

Description: If the clock out functionality is enabled on either CLKOUT_0 and/or CLKOUT_1 and is configured for divide by 2 (via MC_CGM_AC6_DC0[DE] and/or MC_CGM_CLKOUT1_DC0[DE] = 0b1), then if the clock selection for CLKOUT_0/CLKOUT_1 is changed via MC_CGM_AC6_SC[SELCTL]/MC_CGM_CLKOUT1_SC[SELCTL] register respectively or a Destructive, Functional (long/short) reset occurs then the clock out may stop. The following clock sources when selected are affected:

- FXOSC
- FXOSC divided
- FXOSC ANA Clk
- SXOSC
- SXOSC divided
- SIRC
- SIRC divided

- PLL_CLKOUT1
- PLL_CLKOUT2
- RTC_CLK
- CAN0 CHI clk (when driven by FXOSC, not affected when driven by FS80)
- CAN0 PE clk (when driven by FXOSC, not affected when driven by F40)

Workaround: Changing CLKOUT_0/CLKOUT_1 clock source selection value via software, resets all its corresponding dividers and recovers them.

Apply the following sequence after each reset for enabled CLKOUT_0/CLKOUT_1 clock dividers that are to be configured to divide by 2 for the application.

1. Disable the CLKOUT_0 and/or CLKOUT_1 clock divider by writing to MC_CGM_AC6_DC0[DE] and/or MC_CGM_CLKOUT1_DC0[DE] = 0b0

2. Change the CLKOUT_0 and/or CLKOUT_1 clock source selection to FIRC (MC_CGM_AC6_SC[SELCTL] = 0b0001 and/or MC_CGM_CLKOUT1_SC[SELCTL] = 0b1001).

3. Select the desired clock source as the CLKOUT_0 and/or CLKOUT_1 clock source (e.g. for FXOSC: MC_CGM_AC6_SC[SELCTL] = 0b0000 and/or MC_CGM_CLKOUT1_SC[SELCTL] = 0b1000).

4. Configure and enable the corresponding CLKOUT_0 and/or CLKOUT_1 clock divider by writing to MC_CGM_AC6_DC0[DE] and/or MC_CGM_CLKOUT1_DC0[DE] = 0b1.

e10440: MC_ME & LPU: The transition between DRUN/RUN mode to STANDBY or DRUN/RUN mode to LPU_RUN may not complete if a reset is asserted.

Description: The following reset sources can cause the errata condition but all can either be disabled via a control register or avoided as they are triggered by software.

Destructive Resets indicated at MC_RGM_DES:

• Software Watchdog Timer 0, MC_RGM_DES[F_SWT0_RES]

• Software Watchdog Timer 1, MC_RGM_DES[F_SWT1_RES]

• Software Watchdog Timer 2, MC_RGM_DES[F_SWT2_RES]

Functional Resets indicated at MC_RGM_FES:

• Non Maskable Interrupt from Wakeup Unit, MC_RGM_FES[F_NMI_WKPU]

• Clock Monitor Unit FXOSC less than FIRC, MC_RGM_FES[F_CMU_OLR]

• Fault Collection and Control Unit Long Functional Reset, MC_RGM_FES[F_FCCU_LONG]

• Fault Collection and Control Unit Short Functional Reset, MC_RGM_FES[F_FCCU_SHORT]

• VDD_HV_A Low Voltage Detect, MC_RGM_FES[F_LVD_IO_A_HI]

• High Voltage Detect, MC_RGM_FES[F_HVD_LV_cold]

• Power Domain 2 Low Voltage Detect, MC_RGM_FES[F_LVD_LV_PD2_cold]

At the DRUN/RUNx to STANDBY transition there are 2 windows (each 50nS typical) at which time if any of the listed resets are asserted, the mode transition will not complete. The 2 windows occur in the STANDBY entry transition period (20uS typical) - this period is from the mode transition request at the MC_ME_MCTL register to a toggle of the EXTREGC (External Regulator Control) pin. The EXTREG pin signals the low power transition is complete.

At the DRUN/RUN to LPU_RUN transition there are 3 windows:

1. A single window, 686nS (typical) for DRUN-FIRC or 236nS (typical) for DRUN-FMPLL160MHZ.

2. Plus 2 other windows each 50nS typical.

If any of the listed resets are asserted in any of the 3 windows the mode transition will not complete. The 3 windows occur in the LPU_RUN entry transition period (20uS typical) - this period is from the mode transition request at the MC_ME_MCTL register to a toggle of the EXTREGC pin.

For the case the mode transition does not complete the MCU will be stuck in reset or stuck in STANDBY and will only recover via a power-cycle of VDD_HVA.

Workaround: Prior to transitioning to STANDBY or LPU_RUN mode the application should:

Configure the following reset sources so they cannot be triggered in the window of susceptibility:

• Software Watchdog Timer 0, MC_RGM_DES[F_SWT0_RES]

• Software Watchdog Timer 1, MC_RGM_DES[F_SWT1_RES]

• Software Watchdog Timer 2, MC_RGM_DES[F_SWT2_RES]

Disable the following reset sources:

• Functional Reset Escalation, MC_RGM_FES[F_FUNC_ESC]

• Non Maskable Interrupt from Wakeup Unit, MC_RGM_FES[F_NMI_WKPU]

• Clock Monitor Unit FXOSC less than FIRC, MC_RGM_FES[F_CMU_OLR]

• Fault Collection and Control Unit Long Functional Reset, MC_RGM_FES[F_FCCU_LONG]

• Fault Collection and Control Unit Short Functional Reset, MC_RGM_FES[F_FCCU_SHORT]

• VDD_HV_A Low Voltage Detect, MC_RGM_FES[F_LVD_IO_A_HI]

• High Voltage Detect, MC_RGM_FES[F_HVD_LV_cold]

• Power Domain 2 Low Voltage Detect, MC_RGM_FES[F_LVD_LV_PD2_cold]

e10361: MC_ME & LPU: The transition between DRUN/RUN mode to STANDBY or DRUN/RUN mode to LPU_RUN may not complete if EXR is asserted.

Description: At the DRUN/RUNx to STANDBY transition there are 2 windows (each 50nS typical) at which time if the External Reset (EXR) is asserted, the mode transition will not complete. The 2 windows occur in the STANDBY entry transition period (20uS typical) - this period is from the mode transition request at the MC_ME_MCTL register to a toggle of the EXTREGC (External Regulator Control) pin. The EXTREG pin signals the low power transition is complete.

At the DRUN/RUN to LPU_RUN transition there are 3 windows:

1. A single window, 686nS (typical) for DRUN-FIRC or 236nS (typical) for DRUN-FMPLL160MHZ.

2. Plus 2 other windows each 50nS typical.

If EXR is asserted in any of the 3 windows the mode transition will not complete. The 3 windows occur in the LPU_RUN entry transition period (20uS typical) - this period is from the mode transition request at the MC_ME_MCTL register to a toggle of the EXTREGC pin.

For the case the mode transition does not complete the MCU will be stuck in reset or stuck in STANDBY and will only recover via a power-cycle of VDD_HVA.

Workaround: 1. Ensure that External Reset (EXR) is not triggered during the windows of susceptibility at the entry to STANDBY mode or at the entry to LPU_RUN mode.

2. Alternatively, use the unaffected low power mode STOP.

e10362: MC_ME & LPU: The transition between DRUN/RUN mode to STANDBY or DRUN/RUN mode to LPU_RUN may not complete if any LVD is asserted or PORST goes low

Description: At power-up to DRUN there is a window (50nS typical) at which time if any of the Low Voltage Detects (LVDs) are asserted or PORST goes low, the mode transition will not complete.

At the DRUN/RUNx to STANDBY transition or DRUN/RUN to LPU_RUN there are 2 windows (each 50nS typical) at which time if any of the LVDs are asserted, the mode transition will not complete. The 2 windows occur in the STANDBY/LPU entry transition period (20uS typical) - this period is from the mode transition request at the MC_ME_MCTL register to a toggle of the EXTREGC (External Regulator Control) pin. The EXTREG pin signals the low power transition is complete.

For the case the mode transition does not complete the MCU will be stuck in reset and will only recover via a power-cycle of VDD_HVA.

Upon wake-up from STANDBY or LPU_RUN modes there is a single window (50nS typical) at which time if any of the LVDs are asserted or PORST is asserted, the mode transition will not complete. This window occurs in the STANDBY/LPU exit transition period (12uS typical) immediately after assertion of the wake-up signal. For this case when the mode transition does not complete the MCU will be stuck in reset and recover via a power-cycle of VDD_HVA.

Workaround: 1. Ensure that no Low Voltage Detect (LVD) is triggered or PORST goes low during the windows of susceptibility at Power-up, at the entry to STANDBY mode, at the exit of STANDBY, at the exit of LPU modes, or at the entry to LPU_RUN mode.

2. Alternatively, use the unaffected low power mode STOP.

e9200: MC_ME and LPU: JTAG TCK pin must be configured to ensure successful exit from STANDBY and LPU modes

Description: If the JTAG Test Clock Input (TCK) pin is not driven, it is possible at the exit from STANDBY or LPU (Low Power Unit) modes the core clock(s) may not start. If this state occurs, the MCU can be reset via a power cycle or PORST pin. MCU will also recover if TCK pin is driven low externally.

Workaround: There are 2 options for the workaround depending on whether a debugger is attached:

1. Debugger is detached:

- Prior to entering STANDBY or LPU modes, the input buffer of the TCK pad must be disabled in the Multiplexed Signal Configuration Register (SIUL2_MSCR[IBE] = 0).
- To re-enable debug functionality at exit from STANDBY or LPU modes, the input buffer of the TCK pad must be enabled (SIUL2_MSCR[IBE]=0b1).

Note for the 324MPABGA package TCK can be configured for pad 121 and pad 197 and is selected by JTAG_SELECT.

2. Debugger is attached:

- When the debugger is connected it will drive TCK, hence there is no impact.

e10323: MC_ME: The transition from DRUN/RUN mode to STANDBY will not complete if a wake-up is triggered in a 50nS window.

Description: At the DRUN/RUNx to STANDBY mode transition there are 2 windows (each 50nS typical) at which time if a WKPU (wake-up) occurs the mode transition will not complete. The 2 windows occur in the STANDBY entry transition period (20uS typical) - this period is from the mode transition request at the MC_ME_MCTL register to a toggle of the EXTREGC (External Regulator Control) pin. The EXTREGC pin signals the low power transition is complete.

For the case the mode transition does not complete the MCU will be stuck in reset (window #1) or stuck in STANDBY (window #2) and will only recover via a power-cycle of VDD_HVA.

Workaround: Ensure wake-ups are not triggered in the STANDBY entry transition period during the DRUN to STANDBY mode transition, by adhering to all of the following:

- If the application cannot guarantee to avoid triggering an external wake-up during the STANDBY entry transition period, prior to entering STANDBY mode all external wake-ups must be disabled.
- Application SW should use a periodic wake-up (RTC-API) and poll WKPU_WISR. If a wake-up is recorded at WKPU_WISR this signals to the application SW that an external wake-up has occurred whilst in STANDBY mode.
- The RTC-API timer must not timeout during the STANDBY entry transition period.

Alternatively, use an unaffected mode

a) LPU_STANDBY(FIRC-on) rather than STANDBY. In LPU_STANDBY mode, external wake-ups can be enabled (see also e10132).

b) STOP mode.

e8871: ME: In STANDBY/LPU STANDBY modes, if FIRC is disabled all 256kB of STANDBY RAM is retained

Description: For the case when the FIRC is enabled in STANDBY/LPU STANDBY mode the amount of RAM retained in STANDBY mode is configurable at the PMCDIG RAM Domain Configuration Register (PMCDIG_RDCR). However, for the case when the FIRC is disabled in STANDBY mode the configuration at PMCDIG_RDCR is not applied and all 256K RAM is retained in STANDBY mode.

Workaround: When the FIRC is disabled in STANDBY/LPU STANDBY mode the user need not select the amount of RAM to be retained in STANDBY mode, controlled at PMCDIG_RDCR.

e8898: ME: For a supply voltage of greater than 5.3V the MCU may be reset during a LPU_STANDBY to LPU_RUN mode transition

Description: The MPC5748G datasheet specifies a maximum supply voltage (VDD_HV_A) of 5.5V when the MCU is configured for the 5.0V range. For a supply voltage of 5.3V and greater there is a possibility the MCU may trigger a power-on-reset at the LPU_STANDBY to LPU_RUN mode transition. Note the following STANDBY mode transitions are not affected

- STANDBY to DRUN
- LPU_STANDBY to DRUN

This issue is only present in a corner case of the silicon process.

Workaround: When the supply voltage is 5.3V and greater the user should exit LPU_STANDBY via DRUN and then re-enter LPU_RUN.

e8880: MEMU: After exit from LPU RUN mode, the MEMU_PROT registers are uninitialized which may impact the usage of MEMU

Description: The MEMU_PROT module corresponds to the register protection unit for MEMU.

Both MEMU and MEMU_PROT are present in power domain2.

The issue is that the MEMU PROT module does not get initialized after LPU RUN to DRUN mode transition.

This will lead to random register values in MEMU PROT including the LOCK bits which will hinder the usage of MEMU by preventing the user from re-writing MEMU registers.

Workaround: If the Lock bits get set, the only workaround is to issue a Reset. This could be done using soft reset from the user application code.

e8870: NEXUS: Mutli core tracing using NEXUS3 gives corrupted traces for the case when the cores are in the ratio of 2:1

Description: Mutli core tracing using NEXUS3 gives corrupted traces for the case when the cores are in the ratio of 2:1

This is applicable to both data and instruction trace.

There is no issue when multi core tracing only Z4A and Z4B as the frequency ratio is always 1:1.

Care has to be taken when multi core tracing of Z4A/Z4B and Z2 as the defualt frequency ratio is 2:1.

Workaround: There are two workarounds:

1. Enable timestamping during tracing.

2. Configure the Clock Generation Module (CGM) such that the Z4A/Z4B and Z2 core frequency ratio is 1:1.

e9135: NPC: LPM Debug handshake does not work for STOP Mode using all TCK frequencies

Description: If a debugger or other tool is connected to the microcontroller and enables the Low Power Mode handshake in the Nexus Port Controller 1 Port Configuration Register (NPC_1 PCR[LP_DBG] = 1), the MCU may not transition from STOP mode back to RUN mode if the JTAG Test Clock (TCK) is less than one-tenth (1/10) of the system frequency. Transitions to and from the normal STANDBY mode or transitions using the Low Power Subsystem (LPU), including into the LPU_STOP, are not impacted and operate correctly.

Workaround: Use a TCK frequency greater than one-tenth of the System Clock frequency if using a tool and transitioning into and out of the normal STOP mode.

e10603: NPC: Nexus Port Controller (NPC) must be enabled to allow mode changes during debug

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

Workaround: Enable the NPC by enabling the Message Clock Output (MCKO_EN = 1) in the NPC Port Configuration Register (NPC_PCR).

e10723: NPC: Repeated Nexus3 Debug Status messages can be observed if more than one master (including a device core) is active and the core is subsequently disabled

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

Workaround: While transitioning to a low power mode(STOP, STANDBY, LPU_RUN), use the NPC Handshake by clearing NPC_1 PCR [LP1_SYNC] bit. The debugger can then disable the Nexus3 tracing of the core before it acknowledges that the transition into a low-power mode may proceed . For a non-low power mode transition (DRUN, RUNx) , do not disable device core but instead use the Power Architecture ‘wait' instruction to move the device core to the wait state.

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

e10340: 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.

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

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

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

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

e9873: PFLASH: Calibration remap to flash memory not supported on 16KB and 32KB flash blocks in address range 0x00F90000-0x00FBFFFF

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

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

e10789: PFLASH: EEPROM ECC error suppression is not supported on 16KB and 32KB flash blocks in the address range 0x00F90000-0x00FBFFFF

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

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

e11096: SAI: Internal bit clock is not generated when RCR2[BCI]=1 or TCR2[BCI]=1

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

a) SYNC = 00 and BCS = 0

b) SYNC = 01 and BCS = 1

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

a) SYNC = 01 and BCS = 0

b) SYNC = 00 and BCS = 1

e11150: SAI: Internally generated receive or transmit BCLK cannot be re-enabled if it is first disabled when RCR2[DIV] or TCR2[DIV] > 0

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

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

e8406: SMPU: Process Identifier region hit determination is not available in debug mode

Description: The Process Identifier (PID) feature can be used in the determination of whether the current access hits the SMPU region descriptor. When in debug mode the PID feature is not functional.

Workaround: The SMPU PID feature should be used when debug mode is disabled.

e10103: STCU2: Unexpected STCU self-test timeout can occur when a short functional 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 pulled low) and this reset is configured to cause a short functional reset, the self test does not issue a hardware abort but rather the STCU watchdog signals a time-out. This means that the STCU2 Error Register On-line Hardware Abort Flag (STCU2_ERR_STAT [ABORTHW]) will not be set, but the On-Line LOCK Error (STCU2_ERR_STAT[LOCKESW]) and On-Line Watchdog Time-out (STCU2_ERR_STAT [WDTOSW]) flags will be set after the reset. The duration for which the device waits for self-test to complete when this condition occurs is dependent on the watchdog time-out value set in (STCU2_WDG[WDGEOC])

Workaround: Do not configure functional reset sources as short functional reset when selftest is running.

e8902: STM: The STM Counter Register will not report count value when TEN is cleared

Description: For the case when the System Timer Module (STM) counter moves from the running state to the disabled state using the Timer Counter Enable (STM_CR[TEN]) the STM Count Register (STM_CNT) will report zero or the last counter load value, instead of the counter value. Note that when TEN is reasserted the counter will continue from the last count value.

Workaround: To read the current value of the STM Count Register, TEN must be enabled.

e9322: TDM: Erase of TDR flash block may be blocked by the TDM

Description: Erase of a flash block associated with a Tamper Detect Region (TDR) may be improperly blocked by the Tamper Detect Module (TDM) in a system if the Hardware Security Module (HSM) is also performing a program or erase operation. When this occurs, the erase operation will be shown as complete with Program/Erase Good (c5FMC_MCR.DONE=1 and C55FMC_MCR.PEG=1), but the block will not be erased.

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

e8868: WKPU: Wakeup Pads pull-ups cannot be enabled in Standby Mode

Description: During Standby Mode, all 30 external wakeup Pins cannot be configured in Pulled-Up state using WKPU register WIPUER.

If WIPUER[x] is programmed to value 0, the wakeup pins go to a High-Z state.

If WIPUER[x] is programmed to value 1, the wakeup pins gets configured to Pulled-down state.

In All Modes except Standby Mode, the pull state of wakeup Pins can be configured using its associated SIUL_MSCRx[PUE] and SIUL_MSCRx[PUS] register bits.

Workaround: Use external pull up on board if wakeup pins are required to be pulled up in standby mode.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support