

Mask Set Errata for Mask 1N06M

This report applies to mask 1N06M for these products:

- MPC5746C
- MPC5746B
- MPC5745B
- MPC5745C
- MPC5744B
- MPC5744C

Mask Specific Information

JTAG identifier	0x1988_401D
-----------------	-------------

Table 1. Errata and Information Summary

Erratum ID	Erratum Title
e10327	ADC: Incorrect channel under measure is indicated after sampling phase of conversion until conversion completes
e10058	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
e9656	DSPI: Frame transfer does not restart in case of SPI parity error in master mode
e10491	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
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

Table continues on the next page...



Table 1. Errata and Information Summary (continued)

Erratum ID	Erratum Title
e8180	HSM: e200z0 Nexus interface DQTAG implemented as variable length field in DQM message
e10117	HSM: The True Random Number Generator (TRNG) can only use the 16MHz internal RC oscillator (FIRC) as a clock source
e8951	I2C: Attempting a start cycle while the bus is busy may generate a short clock pulse
e7274	LINFlexD: Consecutive headers received by LIN Slave triggers the LIN FSM to an unexpected state
e8933	LINFlexD: Inconsistent sync field may cause an incorrect baud rate and the Sync Field Error Flag may not be set
e8970	LINFlexD: Spurious bit error in extended frame mode may cause an incorrect Idle State
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
e10447	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
e10323	MC_ME: The transition from DRUN/RUN mode to STANDBY will not complete if a wake-up is triggered in a 50nS window.
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
e10789	PFLASH: EEPROM ECC error suppression is not supported on 16KB and 32KB flash blocks in the address range 0x00F90000-0x00FBFFFF
e10549	PMC: A wakeup event from STANDBY/ LPU mode to DRUN may result in a POR assertion if the internal ballast is used
e10763	PRAMC: Possibility of bus error when read burst optimization is enabled and SMPU is configured for cache inhibit
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
e10810	STCU: If the Auxilliary Clock 9 Select Control Register CGM_AC9_SC[SELCTL] = 1, offline-BIST will fail leading to an STCU watchdog timeout
e10200	STM: Reading the System Timer Module Count register may return an incorrect value
e9030	XBAR: End to End ECC may cause masters to stall or fetch data incorrectly
e9993	[STCU]: If PLL is the source of STCU for online or offline self test execution and it suffers an unrecoverable loss of lock, the device would hang.

Table 2. Revision History

Revision	Changes
0	Initial revision
1	The following errata were revised. <ul style="list-style-type: none"> • e8951
2	The following errata were added. <ul style="list-style-type: none"> • e9978 • e9996 • e9994 • e9995 • e9993 • e9992 • e9656 • e10058 The following errata were revised. <ul style="list-style-type: none"> • e8933
3	The following errata were removed. <ul style="list-style-type: none"> • e9992 The following errata were added. <ul style="list-style-type: none"> • e10132 • e10141 • e10117
3.1 04/16	The following errata were added. <ul style="list-style-type: none"> • e8759
4 July 2016	The following errata were removed. <ul style="list-style-type: none"> • e9994 The following errata were added. <ul style="list-style-type: none"> • e10327 • e10323 • e10447 • e10340 • e10362 • e10361 • e10368 The following errata were revised. <ul style="list-style-type: none"> • e10132
5, January 2018	The following errata were added. <ul style="list-style-type: none"> • e10595 • e10609 • e11096 • e10723 • e10620 • e10603

Table 2. Revision History

Revision	Changes
	<ul style="list-style-type: none">• e10200• e10413• e10810• e10763• e10789• e11150• e10549• e9030• e10491 <p>The following errata were revised.</p> <ul style="list-style-type: none">• e10340• e10058

e10327: ADC: Incorrect channel under measure is indicated after sampling phase of conversion until conversion completes

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

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

e10058: 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).

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

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

1. Master/slave mode select bit (MSTR) 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).

e10491: 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 feature of a DMA channel is enabled via the Channel n Master ID Register (DMA_DCHMIDn) by setting the Enable Master ID replication (EMI) bit (DMA_DCHMIDn[EMI]=1), the DMA_DCHMIDn[PAL] and DMA_DCHMIDn[MID] fields should reflect the privileged access level (PAL) and master ID (MID) respectively of the master that wrote the Transfer Control Descriptor (TCD) Control and Status register (DMA_TCDn_WORD_7) least significant byte (DMA_TCDn_WORD_7[DONE,ACTIVE, MAJOR_E_LINK, E_SG, DREQ, INT_HALF, INT_MAJOR, START] byte). However, if a different master reads the DMA_TCDn_WORD_7 least significant byte, the MID and PAL of DMA_DCHMIDn will incorrectly change to this read access master's MID and PAL.

Workaround: Only allow the intended master ID replication core to access the DMA_TCDn_WORD_7 least significant byte (including accessing the full TCD word).

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.

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 described 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.

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.

e10117: HSM: The True Random Number Generator (TRNG) can only use the 16MHz internal RC oscillator (FIRC) as a clock source

Description: The user must not select the Fast External Oscillator (FXOSC) as a clock source for the HSM TRNG. The HSM TRNG clock source is selected at the Auxilliary Clock 3 Select Control Register, selection control bit (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).

e8951: I2C: Attempting a start cycle while the bus is busy may generate a short clock pulse

Description: When the I2C (Inter-Integrated Circuit) is operating in a multi-master network and a start cycle is attempted by the I2C device when the bus is busy, the attempting master will lose arbitration as expected but a short extra clock cycle is generated in the bus. After losing arbitration, the master switches to slave mode but it does not detect the short clock pulse. The acknowledge signal is expected at the ninth clock by the current bus master but it is not sent as expected due to the undetected short clock pulse.

Workaround: Software must ensure that the I2C BUS is idle by checking the bus busy bit in the I2C Bus Status Register (I2C_IBSR.IBB) before switching to master mode and attempting a Start cycle.

e7274: LINFlexD: Consecutive headers received by LIN Slave triggers the LIN FSM to an unexpected state

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

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

Workaround: The following three steps should be followed -

- 1) Configure slave to Set the MODE bit in the LIN Time-Out Control Status Register (LINTCSR[MODE]) to '0'.
- 2) Configure slave to Set Idle on Timeout in the LINTCSR[IOT] register to '1'. This causes the LIN Slave to go to an IDLE state before the next header arrives, which will be accepted without any framing error.
- 3) Configure master to wait for Frame maximum time (T Frame_Maximum as per LIN specifications) before sending the next header.

Note:

$$T_{Header_Nominal} = 34 * T_{Bit}$$

$$T_{Response_Nominal} = 10 * (N_{Data} + 1) * T_{Bit}$$

$$T_{Header_Maximum} = 1.4 * T_{Header_Nominal}$$

$$T_{Response_Maximum} = 1.4 * T_{Response_Nominal}$$

$$T_{Frame_Maximum} = T_{Header_Maximum} + T_{Response_Maximum}$$

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

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 (LINC1[MME] = 0b0)
2. Auto synchronization is enabled by setting LIN Auto Synchronization Enable (LINC1[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.

e8970: LINFlexD: Spurious bit error in extended frame mode may cause an incorrect Idle State

Description: The LINFlexD module may set a spurious Bit Error Flag (BEF) in the LIN Error Status Register (LINESR), when the LINFlexD module is configured as follows:

- Data Size greater than eight data bytes (extended frames) by configuring the Data Field Length (DFL) bitfield in the Buffer Identifier Register (BIDR) with a value greater than seven (eight data bytes)
- Bit error is able to reset the LIN state machine by setting Idle on Bit Error (IOBE) bit in the LIN Control Register 2 (LINCR2)

As consequence, the state machine may go to the Idle State when the LINFlexD module tries the transmission of the next eight bytes, after the first ones have been successfully transmitted and Data Buffer Empty Flag (DBEF) was set in the LIN Status Register (LINSR).

Workaround: Do not use the extended frame mode by configuring Data Field Length (DFL) bit-field with a value less than eight in the Buffer Identifier Register (BIDR) (BIDR[DFL] < 8)

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.

e10447: 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]

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]

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.

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.

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.

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.

e10549: PMC: A wakeup event from STANDBY/ LPU mode to DRUN may result in a POR assertion if the internal ballast is used

Description: If the device is operating using Internal regulation with internal ballast, a wakeup from STANDBY or any of the LPU modes to DRUN, may result in a POR (Power On Reset) event. The POR event sources that can lead to this behavior are LVD_LV_PD2_hot, and/or LVD_LV_PD1_hot, LVD_LV_PD0_hot and POR_LV.

Workaround: Do not use the Internal ballast in applications using any of the LPU modes or STANDBY mode. Instead use an off chip NPN transistor as a pass device as described in the power management section of the reference manual and ensure that the device is set up to use external ballast (INT_BAL_SELECT pin tied to ground).

e10763: PRAMC: Possibility of bus error when read burst optimization is enabled and SMPU is configured for cache inhibit

Description: It is possible that a bus master receives a bus error when accessing a RAM location that is configured by the SMPU RGD (System Memory Protection Unit Region Descriptor) with cache inhibit enabled at SMPUx_RGDn_WRD3[CI]. The erratum is only valid when the PRAMC (Platform RAM Controller) is configured with port read burst optimization enabled. It should be noted that the reset value of PRAMCx_PFCR1 configures port read burst optimization to be enabled by default. For the case a core access to the RAM triggers the bus error, the core would experience an exception.

Workaround: The PRAMC port read burst optimization should be disabled for the case when a SMPU RGD spans the RAM and has cache-inhibit enabled. The PRAMC port read burst optimization can be disabled by setting PRAMCx_PFCR1[Py_BO_DIS]. This workaround will have a minimal impact on MCU performance.

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.

e10810: STCU: If the Auxilliary Clock 9 Select Control Register CGM_AC9_SC[SELCTL] = 1, offline-BIST will fail leading to an STCU watchdog timeout

Description: When STCU (Self Test Control Unit) offline-BIST (Built In Self Test) is enabled (default) and the Auxilliary Clock 9 Select Control Register CGM_AC9_SC[SELCTL] = 1 (default = 0) the offline-BIST will not complete leading to an STCU watchdog timeout. STCU watchdog timeout duration depends on STCU_WDG DCF record programmed before the STCU_RUN DCF record.

Workaround: To avoid the STCU offline-BIST failure, leading to the STCU watchdog timeout the user must select 1 of the following workarounds:

1. If there is a requirement for CGM_AC9_SC[SELCTL] = 1 (FXOSC) the user can avoid STCU offline-BIST failure by disabling FlexCAN_0 MBIST by programming STCU_MB_CTRL24 DCF (0xAA0000000080660). This will reduce BIST coverage as FlexCAN_0 MBIST is excluded.
2. If STCU offline-BIST is enabled for 100% coverage the user must select CGM_AC9_SC[SELCTL] = 0 (FS80).
3. If there is a requirement for CGM_AC9_SC[SELCTL] = 1 (FXOSC) the user can disable STCU offline-BIST by programming STCU_CFG DCF (0x7F000000008000C).

e10200: STM: Reading the System Timer Module Count register may return an incorrect value

Description: The erratum may only occur when the STM is configured to use the FXOSC clock source selected at STM_CR[CSL]. For the case application software reads the STM Count register (STM_CNT) the value returned may be incorrect. However, the user should be assured that STM interrupts will continue to be triggered at the expected STM_CNT value.

Note the default clock source for the STM is the FS80 (divided system clock) and this configuration is not impacted by this erratum.

Workaround: To avoid the erratum condition the user should select the FS80 clock source for the STM. However, for the case the FXOSC is required to clock the STM and the STM Count register is to be read, the following sequence must be executed:

1. Disable the STM via STM_CR[TEN]
2. Read STM Counter register STM_CNT
3. Re-enable the STM via STM_CR[TEN]

e9030: XBAR: End to End ECC may cause masters to stall or fetch data incorrectly

Description: Default programming of bus traffic optimization for a subset of masters (DMA, FlexRay and ENET) can cause those masters to stall, receive wrong read data, or get a spurious read access when uncorrectable ECC errors are received from slaves. For a read transaction following an uncorrectable ECC error, these masters can stall or receive the wrong data.

Workaround: There is a software workaround where code can disable the Pending Read Enable (PRE*) for each affected pathway. The software workaround will incur a small performance degradation for the pathways of the affected masters on case to case basis.

1. DMA: Disable PRE_DMA bit in register PCM_IAHB_BE0. Setting the DMA channel TCD SSIZE equal to DSIZE will eliminate performance loss.
2. FlexRay: Disable PRE_FLX bit in register PCM_IAHB_BE0.
3. ENET: Disable PRE_ENET bit in register PCM_IAHB_BE0.

e9993: [STCU]: If PLL is the source of STCU for online or offline self test execution and it suffers an unrecoverable loss of lock, the device would hang.

Description: During online or offline self-test execution, a PLL loss of lock event is non recoverable resulting in the device stuck in reset. This could potentially happen if the PLL is using the external oscillator as it's clock source and the external crystal oscillator becomes non functional.

Workaround: Use an alternative clock source for offline or online self test execution. This could be either the FIRC or PLL with FIRC as reference clock.

How to Reach Us:**Home Page:**nxp.com**Web Support:**nxp.com/support

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

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

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and μ Vision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2018 NXP B.V.

