

Mask Set Errata for Mask 1N84S

This report applies to mask 1N84S for these products:

- MPC5746C
- MPC5746B
- MPC5746D
- MPC5745B
- MPC5745C
- MPC5745D

Mask Specific Information

JTAG identifier	0x1988_501D
-----------------	-------------

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
e10620	Clocking: Flexray doesnt work with FS80 clock source selection in linear DFS mode.
e10542	DSPI: Transmit, Command, and Receive FIFO fill flags in status register is not cleared when DMA is improperly configured
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
e10595	FlexCAN: FLEXCAN1-7 modules will not work unless the Fast External Oscillator (FXOSC) clock source is enabled
e8770	FlexRAY: Missing TX frames on Channel B when in dual channel mode and Channel A is disabled
e8180	HSM: e200z0 Nexus interface DQTAG implemented as variable length field in DQM message
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
e10609	MC_CGM: CLKOUT_0 and CLKOUT_1 dividers may become stuck if clock selection is changed while dividers with divide by 2 are operational

Table continues on the next page...



Table 1. Errata and Information Summary (continued)

Erratum ID	Erratum Title
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
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
e10103	STCU2: Unexpected STCU self-test timeout can occur when a short functional reset is triggered during execution of online self-test
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
e10577	STCU: Device may get stuck in reset if the Fast External Oscillator (FXOSC) is lost during self test
e10200	STM: System Timer Module (STM) Count register read corruption

Table 2. Revision History

Revision	Changes
1 March 2017	Initial revision

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).

e10620: Clocking: Flexray doesnt work with FS80 clock source selection in linear DFS mode.

Description: Flexray doesnt work with FS80 clock source selection in linear DFS mode. The reason is that the FlexRay needs 80 MHz PLL clock or 40 MHz XOSC but in the linear DFS mode the FS80 has value only 40MHz so only XOSC can be used as a clock source.

Workaround: Before entering linear DFS mode customer should select FXOSC as clock source for Flexray PE clock.

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

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

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

Use the DMA loop to transfer more data if needed.

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.

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).

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.

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.

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)

e10609: MC_CGM: CLKOUT_0 and CLKOUT_1 dividers may become stuck if clock selection is changed while dividers with divide by 2 are operational

Description: If clock out functionality is enabled on either CLKOUT_0 and/or CLKOUT_1 and there is a divide by 2 divider operational on these clocks (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 dividers may become stuck, causing no clock to be output from the divider.

This will not clear until a power on reset occurs.

This is only true if the divider is using divide by 2 (MC_CGM_AC6_DC0[DIV] and/or MC_CGM_CLKOUT1_DC0[DIV] = 0b1).

and one of the following clock source is not selected when a Destructive , Functional (long/short) reset occurs:

FIRC,

FIRC divided,

System clocks (F40,F80,FS80,F160,F20,S40,S80,S160) ,

Clocks derived from system clocks (LPU sys clk,Z2 core clk,Z4 clk,F80 Flexray,F40 CAN,HSM Core clk)

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.

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.

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.

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.

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).

e10577: STCU: Device may get stuck in reset if the Fast External Oscillator (FXOSC) is lost during self test

Description: If the fast external oscillator (FXOSC) is used as a reference clock for the PLL for online self test and the FXOSC is lost permanently while self test is in progress, the device will be stuck in the reset state.

Workaround: Use the Fast Internal Oscillator (FIRC) as the reference clock for the PLL when running online self test.

e10200: STM: System Timer Module (STM) Count register read corruption

Description: Reading of the System Timer Module (STM) Counter register, STMx_CNT(x denotes the instance of STM), can provide a corrupted value if enabled while reading. This applies to all instances of the STM on the device.

Workaround: 1. To ensure an accurate read of the STM Timer Counter STMx_CNT, the Timer must be disabled first through the TEN bit in the STMx_CR register. Then the counter timer will be stable and readable. Once read the STM can be immediately restarted if desired.

OR

2. STM should be operated on system clock frequency rather than FXOSC

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.

© 2017 NXP B.V.

