MPC5746C_1N06M

Mask Set Errata

5 June 2024 Errata

1 Mask Set Errata for Mask 1N06M

1.1 Revision History

This report applies to mask 1N06M for these products:

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

Table 1. Mask Specific Information

jtag_id	0x1988_401D

Table 2. Revision History

Revision	Date	Significant Changes
10	3/2024	The following errata were added. • ERR052152
9	8/2022	The following errata were revised. • ERR011235 • ERR010590
8	3/2022	The following errata were revised. • ERR050575
7	8/2021	The following errata were removed. • ERR010117 The following errata were added. • ERR050575 • ERR011306 • ERR010590 • ERR010542 The following errata were revised. • ERR010763 • ERR011235 • ERR010963 • ERR050246
6	3/2021	The following errata were removed. • ERR009030 • ERR010141 • ERR008951

Table 2. Revision History*...continued*

Revision	Date	Significant Changes
		• ERR010327 The following errata were added. • ERR011321 • ERR010385 • ERR050130 • ERR050090 • ERR050195 • ERR011235 • ERR050572 • ERR050196 • ERR010963 • ERR050467 • ERR050144 • ERR050119 • ERR011287 • ERR011295 • ERR011294 • ERR011293 • ERR050154 • ERR050246 The following errata were revised. • ERR010340
5	1/2018	The following errata were added. • ERR010763 • ERR010723 • ERR011096 • ERR010200 • ERR010603 • ERR009030 • ERR010789 • ERR010810 • ERR011150 • ERR010491 • ERR010620 • ERR010595 • ERR010609 • ERR010549 • ERR010413 The following errata were revised. • ERR010340 • ERR010058
4 July 2016	7/2016	The following errata were removed. • ERR009994 The following errata were added. • ERR010447 • ERR010362 • ERR010368 • ERR010340 • ERR010323

Table 2. Revision History*...continued*

Revision	Date	Significant Changes
		• ERR010361 • ERR010327 The following errata were revised. • ERR010132
3.1 04/16	4/2016	The following errata were added. • ERR008759
3	4/2016	The following errata were removed. • ERR009992 The following errata were added. • ERR010132 • ERR010141 • ERR010117
2	2/2016	The following errata were added. • ERR009993 • ERR009978 • ERR009994 • ERR009656 • ERR009996 • ERR010058 • ERR009995 • ERR009992 The following errata were revised. • ERR008933
1	8/2015	The following errata were revised. • ERR008951
0	4/2015	Initial Revision

1.2 Errata and Information Summary

Table 3. Errata and Information Summary

Erratum ID	Erratum Title
ERR007274	LINFlexD: Consecutive headers received by LIN Slave triggers the LIN FSM to an unexpected state
ERR007991	FLASH: Rapid Program or Erase Suspend fail status
ERR008180	HSM: e200z0 Nexus interface DQTAG implemented as variable length field in DQM message
ERR008759	FlexCAN: FD frame format not compliant to the new ISO/CD 11898-1: 2014-12-11
ERR008770	FlexRAY: Missing TX frames on Channel B when in dual channel mode and Channel A is disabled
ERR008933	LINFlexD: Inconsistent sync field may cause an incorrect baud rate and the Sync Field Error Flag may not be set
ERR008970	LINFlexD: Spurious bit error in extended frame mode may cause an incorrect Idle State
ERR009656	DSPI: Frame transfer does not restart in case of SPI parity error in master mode
ERR009978	eMIOS: Unexpected channel flag assertion during GPIO to MCB mode transition

Table 3. Errata and Information Summary*...continued*

Erratum ID	Erratum Title
ERR009993	[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.
ERR009995	DSPI0 and DSPI1: Frame transfer does not restart in case of DSI parity error in master mode
ERR009996	DSPI0 and DSPI1: Frame transfer does not restart after DSI frame matches preprogrammed value
ERR010058	CMP_0: At exit from STANDBY the output of Comparator 0 is configured to high-impedance
ERR010132	LPU: Mode transition to LPU_STOP or LPU_STANDBY may not complete
ERR010200	STM: Reading the System Timer Module Count register may return an incorrect value
ERR010323	MC_ME: The transition from DRUN/RUN mode to STANDBY will not complete if a wake-up is triggered in a 50nS window.
ERR010340	NZxC3: ICNT and HIST fields of a Nexus message are not properly reset following a device reset
ERR010361	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.
ERR010362	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
ERR010368	FlexCAN: Transition of the CAN FD operation enable bit may lead FlexCAN logic to an inconsistent state.
ERR010385	e200z4: Incorrect branch displacement at 16K memory boundaries
ERR010413	HSM: HSM RAM initialization clock out of specification when Accelerated Low Power Exit is enabled with FMPLL = 160MHz
ERR010447	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.
ERR010491	eDMA: When master ID replication is enabled, the stored ID and privilege level will change if read by another master
ERR010542	DSPI: Transmit, Command, and Receive FIFO fill flags in status register is not cleared when DMA is improperly configured
ERR010549	PMC: A wakeup event from STANDBY/ LPU mode to DRUN may result in a POR assertion if the internal ballast is used
ERR010590	PLL: Might remain unlocked when enabled after Standby or power on
ERR010595	FlexCAN: FLEXCAN1-7 modules will not work unless the Fast External Oscillator (FXOSC) clock source is enabled
ERR010603	NPC: Nexus Port Controller (NPC) must be enabled to allow mode changes during debug
ERR010609	MC_CGM: CLKOUT_0 and CLKOUT_1 may stop if the clock selection is changed when configured for divide by 2
ERR010620	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
ERR010723	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
ERR010763	PRAMC: Possibility of bus error when read burst optimization is enabled and SMPU is configured for cache inhibit
ERR010789	PFLASH: EEPROM ECC error suppression is not supported on 16KB and 32KB flash blocks in the address range 0x00F90000-0x00FBFFFF

Table 3. Errata and Information Summary*...continued*

Erratum ID	Erratum Title
ERR010810	STCU: If the Auxilliary Clock 9 Select Control Register CGM_AC9_SC[SELCTL] = 1, offline-BIST will fail leading to an STCU watchdog timeout
ERR010963	Flash: Memory accesses may be corrupted when flash is operating between 33MHz and 75MHz
ERR011096	SAI: Internal bit clock is not generated when RCR2[BCI]=1 or TCR2[BCI]=1
ERR011150	SAI: Internally generated receive or transmit BCLK cannot be re-enabled if it is first disabled when RCR2[DIV] or TCR2[DIV] > 0
ERR011235	EMIOS: Any Unified Channel running in OPWMB or OPWMCB mode may function improperly if the source counter bus is generated by Unified channel in MC mode
ERR011287	CMU: Sudden loss of clock does not signal the Fault Collection and Control Unit
ERR011293	EMIOS: For any UC operating in OPWFMB mode the Channel Count register should not be written with a value greater than Channel B Data register value
ERR011294	EMIOS: OPWFMB and MCB mode counter rollover resets the counter to 0x0 instead of 0x1 as mentioned in the specification
ERR011295	EMIOS: In OPWFMB mode, A1/B1 registers do not get reloaded with A2/B2 register values if counter value returns 0x1 after counter wrap condition
ERR011306	SAR ADC: Incorrect value of ADC power down exit delay evaluated by the formula given in PDEDR [PDED] field description
ERR011321	PIT_RTI: Generates false RTI interrupt on re-enabling
ERR050090	DSPI/SPI: Incorrect data may be transmitted in slave mode
ERR050119	FlexRay: Disabling of FlexRay Message Buffer during the STARTUP Protocol State takes longer than expected three Slots
ERR050130	PIT: Temporary incorrect value reported in LMTR64H register in lifetimer mode
ERR050144	SAI: Setting FCONT=1 when TMR>0 may not function correctly
ERR050154	Clocking: Device operation is impacted with a particular MC_CGM system divider configuration.
ERR050195	MEMU_0: MEMU_0 operation is impacted with a particular MC_CGM system divider configuration.
ERR050196	STANDBY EXIT: Device may not come out of reset if a short functional reset comes immediately after wakeup event.
ERR050246	FlexCAN: Receive Message Buffers may have its Code Field corrupted if the Receive FIFO function is used
ERR050467	PLL: Possible loss of lock when using the PLL bypass calibration mode
ERR050572	SIRC: Clock output may contain extra clock pulses
ERR050575	eMIOS: Any Unified Channel running in OPWMCB mode may function improperly if the lead or trail dead time insertion features is used and its timebase is generated by Unified channel in MCB mode
ERR052152	ENEN/ENET 1G: The hardware acceleration feature for ICMP checksum generation and checking does not work for IPv6.

2 Known Errata

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

2.1.1 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.

2.1.2 Workaround

The following three steps should be followed -

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

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

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

Note:

THeader_Nominal = 34 * TBit

TResponse_Nominal = 10 * (NData + 1) * TBit

THeader_Maximum = 1.4 * THeader_Nominal

TResponse_Maximum = 1.4 * TResponse_Nominal

TFrame_Maximum = THeader_Maximum + TResponse_Maximum

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

2.2 ERR007991: FLASH: Rapid Program or Erase Suspend fail status

2.2.1 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.

2.2.2 Workaround

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

2.3 ERR008180: HSM: e200z0 Nexus interface DQTAG implemented as variable length field in DQM message

2.3.1 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.

2.3.2 Workaround

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

2.4 ERR008759: FlexCAN: FD frame format not compliant to the new ISO/CD 11898-1: 2014-12-11

2.4.1 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.

2.4.2 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.

2.5 ERR008770: FlexRAY: Missing TX frames on Channel B when in dual channel mode and Channel A is disabled

2.5.1 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.

2.5.2 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.

2.6 ERR008933: LINFlexD: Inconsistent sync field may cause an incorrect baud rate and the Sync Field Error Flag may not be set

2.6.1 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.

2.6.2 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.

2.7 ERR008970: LINFlexD: Spurious bit error in extended frame mode may cause an incorrect Idle State

2.7.1 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).

2.7.2 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)

2.8 ERR009656: DSPI: Frame transfer does not restart in case of SPI parity error in master mode

2.8.1 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.

2.8.2 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).

2.9 ERR009978: eMIOS: Unexpected channel flag assertion during GPIO to MCB mode transition

2.9.1 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.

2.9.2 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.

2.10 ERR009993: [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.

2.10.1 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.

2.10.2 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.

2.11 ERR009995: DSPI0 and DSPI1: Frame transfer does not restart in case of DSI parity error in master mode

2.11.1 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.

2.11.2 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).

2.12 ERR009996: DSPI0 and DSPI1: Frame transfer does not restart after DSI frame matches preprogrammed value

2.12.1 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.

2.12.2 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)

2.13 ERR010058: CMP_0: At exit from STANDBY the output of Comparator 0 is configured to high-impedance

2.13.1 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.

2.13.2 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.

2.14 ERR010132: LPU: Mode transition to LPU_STOP or LPU_STANDBY may not complete

2.14.1 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.

2.14.2 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

2.15 ERR010200: STM: Reading the System Timer Module Count register may return an incorrect value

2.15.1 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.

2.15.2 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]

2.16 ERR010323: MC_ME: The transition from DRUN/RUN mode to STANDBY will not complete if a wake-up is triggered in a 50nS window.

2.16.1 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.

2.16.2 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.

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

2.17.1 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, if Branch History trace is enabled, the ICNT and the Branch History (HIST) fields 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.

2.17.2 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), Boot Assist Module (BAM), or BootROM after reset, perform an indirect branch instruction shortly after reset to reset the ICNT (and HIST if Branch History mode is enabled). Also, 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 boot from boot ROM firmware, the BAF or BAM, an indirect branch will occur during the boot ROM/BAF/BAM execution and the tool trace will be re-synchronized prior to the execution of user code.

2.18 ERR010361: 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.

2.18.1 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.

2.18.2 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.

2.19 ERR010362: 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

2.19.1 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.

2.19.2 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.

2.20 ERR010368: FlexCAN: Transition of the CAN FD operation enable bit may lead FlexCAN logic to an inconsistent state.

2.20.1 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.

2.20.2 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.

2.21 ERR010385: e200z4: Incorrect branch displacement at 16K memory boundaries

2.21.1 Description

The branch target address will be incorrectly calculated in the e200z4 core under the following conditions (all conditions must be matched):

- The first full instruction in a 16 Kbyte section/page of code is a 32-bit long branch with a branch displacement value with the lower 14 bits of the displacement exactly 0x3FFE
- And this branch instruction is located at byte offset 0x0002 in the section/page
- And the preceding instruction is a 32-bit length instruction which is misaligned across the 16K boundary
- And both instructions are dual-issued

Under these conditions, the branch target address will be too small by 32Kbytes.

2.21.2 Workaround

After software is compiled and linked, code should be checked to ensure that there are no branch instructions located at address 0x2 of any 16K memory boundary with the lower 14 bits of the displacement equal to 0x3FFE if preceded a 32-bit instruction that crosses the 16K memory boundary. If this sequence occurs, add a NOP instruction or otherwise force a change to the instruction addresses to remove the condition.

A tool is available on nxp.com that can be run to examine code for this condition, search for branch_ displacement_erratum_10385_checker.

2.22 ERR010413: HSM: HSM RAM initialization clock out of specification when Accelerated Low Power Exit is enabled with FMPLL = 160MHz

2.22.1 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.

2.22.2 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.

2.23 ERR010447: 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.

2.23.1 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.

2.23.2 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]

2.24 ERR010491: eDMA: When master ID replication is enabled, the stored ID and privilege level will change if read by another master

2.24.1 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.

2.24.2 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).

2.25 ERR010542: DSPI: Transmit, Command, and Receive FIFO fill flags in status register is not cleared when DMA is improperly configured

2.25.1 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).

2.25.2 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.

2.26 ERR010549: PMC: A wakeup event from STANDBY/ LPU mode to DRUN may result in a POR assertion if the internal ballast is used

2.26.1 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.

2.26.2 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).

2.27 ERR010590: PLL: Might remain unlocked when enabled after Standby or power on

2.27.1 Description

When enabled after an event where the following conditions are met, the PLL might fail to lock:

1.- VDD_LV (1.2 V) was disabled

2.- A Pre-divider of 1 (PLLDIG_PLLDV[PREDIV] = 0 or 1) is used

3.- The PLL is clocked from the fast external oscillator (FXOSC)

Common situations in which VDD_LV might be disabled are Standby and power-off of the device.

2.27.2 Workaround

There are three ways to avoid this:

1.- Avoid using a pre-divider of 1 (PLLDIG_PLLDV[PREDIV] = 0 or 1) and instead use a value of 2.

2.- During Standby leave the supply to VDD_LV enabled.

3.- Initialize the PLL using the fast internal oscillator (FIRC) as source clock and then switch to use the fast external oscillator (FXOSC).

2.28 ERR010595: FlexCAN: FLEXCAN1-7 modules will not work unless the Fast External Oscillator (FXOSC) clock source is enabled

2.28.1 Description

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

2.28.2 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).

2.29 ERR010603: NPC: Nexus Port Controller (NPC) must be enabled to allow mode changes during debug

2.29.1 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.

2.29.2 Workaround

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

2.30 ERR010609: MC_CGM: CLKOUT_0 and CLKOUT_1 may stop if the clock selection is changed when configured for divide by 2

2.30.1 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)

2.30.2 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.

2.31 ERR010620: 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

2.31.1 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.

2.31.2 Workaround

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

2.32 ERR010723: 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

2.32.1 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.

2.32.2 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.

2.33 ERR010763: PRAMC: Possibility of bus error when read burst optimization is enabled and SMPU is configured for cache inhibit

2.33.1 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 note that the reset value of PRAMCx_PRCR1 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.

2.33.2 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_PRCR1[Py_BO_DIS]. This workaround will have a minimal impact on MCU performance.

2.34 ERR010789: PFLASH: EEPROM ECC error suppression is not supported on 16KB and 32KB flash blocks in the address range 0x00F90000-0x00FBFFFF

2.34.1 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.

2.34.2 Workaround

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

2.35 ERR010810: STCU: If the Auxilliary Clock 9 Select Control Register CGM_AC9_SC[SELCTL] = 1, offline-BIST will fail leading to an STCU watchdog timeout

2.35.1 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.

2.35.2 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 (0xAA00000000080660). 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 (0x7F0000000008000C).

2.36 ERR010963: Flash: Memory accesses may be corrupted when flash is operating between 33MHz and 75MHz

2.36.1 Description

Error Correction Code (ECC) errors may be generated in the flash due to corrupted data when all of the following conditions are met:

- The flash operating frequency is greater than 33MHz and less than or equal to 75MHz

- Read Wait State Control (PFLASH_PFCR1[RWSC]) = 2 and Address Pipeline Control (PFLASH_PFCR1[APC]) = 1

- Pipelined reads which are executed between the UTEST flash block and any other flash block. Pipelined reads are overlapping reads, where before the previous read is completed a new read is requested.

Device has UTEST memory space and remaining flash area is the main array memory space. This issue condition occurs if pipelined reads are executed between UTEST and the main array space. If pipelined reads are executed between UTEST and UTEST memory space or between main array and main array memory space, this issue condition will not be seen.

2.36.2 Workaround

When the flash clock frequency is greater than 33MHz and less than or equal to 75MHz configure PFLASH_PFCR1[RWSC] = 2 and PFLASH_PFCR1[APC] = 0. The configuration for all other flash clock frequencies is specified in the MCU datasheet.

2.37 ERR011096: SAI: Internal bit clock is not generated when RCR2[BCI]=1 or TCR2[BCI]=1

2.37.1 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

2.37.2 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

2.38 ERR011150: SAI: Internally generated receive or transmit BCLK cannot be re-enabled if it is first disabled when RCR2[DIV] or TCR2[DIV] > 0

2.38.1 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.

2.38.2 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.

2.39 ERR011235: EMIOS: Any Unified Channel running in OPWMB or OPWMCB mode may function improperly if the source counter bus is generated by Unified channel in MC mode

2.39.1 Description

The Unified channel (UC) configured in Center Aligned Output Pulse Width Modulation Buffered (OPWMCB) or Output Pulse Width Modulation Buffered (OPWMB) modes is not working properly when it is sourced from the UC configured in Modulus Counter (MC) mode by setting the channel control register MODE bitfield to 0x10 or 0x11 and any of its pre-scalers (internal or global) divider ratio is higher than 1.

2.39.2 Workaround

When a counter bus is generated by the UC set in the MC mode with any pre-scaler (internal or global) divider ration higher than 1, don’t use this counter bus for the UC set in OPWMCB or OPWMB mode.

2.40 ERR011287: CMU: Sudden loss of clock does not signal the Fault Collection and Control Unit

2.40.1 Description

The Clock Monitor Unit (CMU) detects when a monitored clock frequency drops below a programmed threshold through the Frequency Less than Low Threshold (FLL) signal. This FLL signal is routed to the Fault Collection and Control Unit (FCCU) providing a mechanism to react to the clock fault. Due to it's implementation, the FLL signal will not be triggered when the monitored clock source suddenly stops.

2.40.2 Workaround

The CMU has an internal signal which is designed to give an indication that the monitored clock has dropped below 1/4 of the reference clock CLKMT0_RMN. This provides an alternative means to detect the sudden loss of clock, however since this internal signal is not routed to the FCCU, the user software must periodically poll bitfield [3] of CMU_ISR register to detect a sudden loss of clock. Write '0b1' to clear the bitfield [3] of CMU_ISR after enabling CMU.

2.41 ERR011293: EMIOS: For any UC operating in OPWFMB mode the Channel Count register should not be written with a value greater than Channel B Data register value

2.41.1 Description

For any Unified Channel (UC) running in Output Pulse-Width and Frequency Modulation Buffered (OPWFMB) mode, Channel Control Register MODE bitfield = 7'h1011000 or 7'h1011010, the internal counter runs from 0x1 to Channel B Data register value.

The internal counter can be overwritten by software using the Chanel Count register during 'freeze' operation.

If a counter wrap occurs due to overwriting of the counter with a value greater than its expiry value (B Data Register value); than the output signal behavior cannot be guaranteed.

2.41.2 Workaround

For any UC operating in OPWFMB mode the Channel Count register should not be written with a value greater than Channel B Data register value.

2.42 ERR011294: EMIOS: OPWFMB and MCB mode counter rollover resets the counter to 0x0 instead of 0x1 as mentioned in the specification

2.42.1 Description

When the enhanced Modular Input/Output System (eMIOS) is used in Output Pulse-Width and Frequency Modulation Buffered (OPWFMB) or Modulus Counter Buffered (MCB) modes, when the counter rolls over, the counter returns to 0x0 instead of 0x1 as specified in the reference manual.

2.42.2 Workaround

In order to avoid the counter wrap condition:

1. Make sure internal counter value is within the 0x1 to (B1 register) value range when the OPWFMB mode is entered.

2. Overwrite of Channel Count register by forcing 'freeze' in OPWFMB mode should not be outside the range of 0x1 to (B register) value.

2.43 ERR011295: EMIOS: In OPWFMB mode, A1/B1 registers do not get reloaded with A2/B2 register values if counter value returns 0x1 after counter wrap condition

2.43.1 Description

In Output Pulse-Width and Frequency Modulation Buffered (OPWFMB) mode, A1/B1 registers do not get reloaded with A2/B2 register values if counter value returns 0x1 after counter wrap condition.

In order to avoid the counter wrap condition make sure internal counter value is within the 0x1 to B1 register value range when the OPWFMB mode is entered. Also overwriting of Channel Count register by forcing 'freeze' in OPWFMB mode should not take internal counter outside 0x1 to B register value.

2.43.2 Workaround

In order to avoid the counter wrap condition:

1. Make sure internal counter value is within the 0x1 to (B1 register) value range when the OPWFMB mode is entered.

2. Overwrite of Channel Count register by forcing 'freeze' in OPWFMB mode should not be outside the range of 0x1 to (B register) value.

2.44 ERR011306: SAR ADC: Incorrect value of ADC power down exit delay evaluated by the formula given in PDEDR [PDED] field description

2.44.1 Description

The formula in the register field ADC_PDEDR [PDED] provides the delay between the power down bit reset and start of conversion value in number of clock cycles of the ADC module clock, however the given formula of PDED x 1/[ADC_clock_frequency] is incorrect. This gives a calculated value that is too short by 1 cycle of ADC Bus clock and 1 cycle of ADC clock (AD_clk).

2.44.2 Workaround

The correct formula that should be used to calculate the value for the ADC_PDEDR[PDED] register is -

(1/ADC Bus clock) + ((PDED+1) x 1/[ADC_clock_frequency])

Where:

ADC_clock_frequency = Frequency of ADC clock (AD_clk)

ADC Bus clock= Module interface clock for register access (ADC_CLK)

2.45 ERR011321: PIT_RTI: Generates false RTI interrupt on re-enabling

2.45.1 Description

A false Real-Time Interrupt (RTI) may be observed when the RTI module is re-enabled if, after servicing an RTI interrupt (by clearing TFLGn[TIF]), the clocks to the RTI module are disabled.

This occurs only if the RTI module clock is disabled within four RTI clock cycles of an RTI Interrupt being cleared.

2.45.2 Workaround

Option 1: The user should check the RTI interrupt flag,TFLGn[TIF] before servicing the interrupt, this flag won't be set for the false/spurious interrupts.

Option 2: Ensure that the module clock to the RTI module is not disabled within four RTI clock cycles after servicing an RTI interrupt. Consult the chip-specific documentation to determine the clock period of the RTI module and implement a time delay of at least five times this period before disabling the RTI module clock.

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

2.46.1 Description

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

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

4 X IPG_CLOCK_PERIOD + 4 X DSPI_CLOCK_PERIOD + 0.5 x SCK_CLOCK_PERIOD

Where:

IPG_CLOCK is the internal bus clock ("system" clock)

DSPI_CLOCK is the protocol clock.

SCK_CLOCK is the Line-Side Serial Communication Clock.

2.46.2 Workaround

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

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

2.47 ERR050119: FlexRay: Disabling of FlexRay Message Buffer during the STARTUP Protocol State takes longer than expected three Slots

2.47.1 Description

Disabling of FlexRay Message Buffer takes longer than the expected three Slots. This is observed, when software application tries to disable the Message Buffer during the FlexRay STARTUP protocol state (vPOC! State = POC:startup) when vPOC!StartupState = “initialize schedule” or “integration consistency check”.

In this scenario, FlexRay Communication Controller keeps the specific Message buffer search results until the availability of next cycle start/segment start/slot start events and therefore prevent the disabling of Message Buffer.

Note:

1.All Message Buffers can be disabled immediately if FlexRay protocol state (vPOC!State) is in following States: “POC:default config”, “POC:config”, “POC:wakeup”, “POC:ready”, “POC:halt”, “POC:startup” and (vPOC! StartupState = "POC:integration listen” or "POC: ColdStart-Listen”).

2.All Message Buffers can be disabled within three slots, if FlexRay protocol state (vPOC!State) is in following states: ”POC: Normal-Active” or ”POC: Normal-Passive”.

2.47.2 Workaround

Do not disable Message Buffer, while FlexRay is in STARTUP protocol State

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

2.48.1 Description

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

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

2.48.2 Workaround

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

2.49 ERR050144: SAI: Setting FCONT=1 when TMR>0 may not function correctly

2.49.1 Description

When FCONT=1 the transmitter will recover after a FIFO error when the FIFO is no longer empty and starting again from the same word in the following frame where the error occurred.

Configuring TMR > 0 will configure one or more words in the frame to be masked (nothing transmitted during that slot). If anything other than the last word(s) in the frame are masked when FCONT=1 and a FIFO Error Flag is set, then the transmitter will not recover and will set FIFO Error Flag during each frame.

2.49.2 Workaround

To avoid this issue, set FCONT in TCR4 to be 0.

2.50 ERR050154: Clocking: Device operation is impacted with a particular MC_CGM system divider configuration.

2.50.1 Description

BAF issues SWT0 timeout destructive reset if the below conditions are met:

1) MC_CGM divider configuration was programmed in application such that ratio of MC_CGM_SC_DC0[DIV] and MC_CGM_SC_DC5[DIV] is 1 and

2) Short functional reset is issued.

2.50.2 Workaround

Do not configure any functional reset source in MC_RGM_FESS to generate short functional reset in this clocking configuration.

2.51 ERR050195: MEMU_0: MEMU_0 operation is impacted with a particular MC_CGM system divider configuration.

2.51.1 Description

Register read-write access on MEMU_0 may not happen correctly if the ratio of MC_CGM_SC_DC0[DIV] and MC_CGM_SC_DC5[DIV] is 1.

2.51.2 Workaround

Do not perform register read write access on MEMU_0 or disable MEMU_0 when this clocking configuration is used.

2.52 ERR050196: STANDBY EXIT: Device may not come out of reset if a short functional reset comes immediately after wakeup event.

2.52.1 Description

The device may get stuck in STANDBY exit sequence if a short functional reset is observed between wakeup event and low power mode exit.

A destructive or POR reset after short functional reset will recover the device out of the stuck condition.

2.52.2 Workaround

Do not configure any functional reset source in MC_RGM_FESS to generate short functional reset before entering low power mode.

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

2.53.1 Description

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

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

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

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

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

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

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

The problem does not occur in cases when only Rx FIFO or only a dedicated MB is used (i.e. either RX MB or Rx FIFO is used). The problem also does not occur when the Enhanced Rx FIFO and dedicated MB are used

in the same application. The problem only occurs if the FlexCAN is programmed to receive in the Legacy FIFO and dedicated MB at the same application.

2.53.2 Workaround

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

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

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

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

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

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

3. Read the contents of the mailbox.

4. Clear the proper flag in the IFLAG register.

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

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

2.54 ERR050467: PLL: Possible loss of lock when using the PLL bypass calibration mode

2.54.1 Description

A momentary PLL loss of lock may occur shortly after enabling the PLL if using the PLL bypass calibration mode (PLLDIG_PLLCAL1[BYPCAL] = 1). Note that the PLLDIG_PLLCAL1[BYPCAL] bit is set by default after exiting the low power STANDBY mode.

2.54.2 Workaround

Do not use the PLL bypass calibration mode and instead ensure PLLDIG_PLLCAL1[BYPCAL] is cleared before enabling the PLL.

2.55 ERR050572: SIRC: Clock output may contain extra clock pulses

2.55.1 Description

The 128 kHz Slow Internal RC (SIRC) oscillator may generate an extra clock pulse per clock period. The occurrence of the potential extra clock pulse may vary for each clock period ranging from no extra clock pulse to one extra clock pulse per period. Factors that affect the occurrence rate are part to part variations, temperature, and core voltage.

The SIRC output clock may be selected as the clock source for the Real Time Clock (RTC) via the RTC_RTCC[CLKSEL] register, as a clock to monitor in the Clock Monitor Unit (CMU) via the CMU_CSR[CLKSEL1] register, or as the clock source for the Software Watchdog Timer (SWT) timers (SWTx, HSM_SWT). The RTC, CMU, and SWT counters/timers may get an extra clock per SIRC clock period causing a higher than expected count (which may affect the RTC count/wakeup duration, the CMU measured SIRC frequency, or the SWT time-out period) or may cause a corrupted count value. The SIRC clock may also be selected to the CLKOUT_0 and CLKOUT_1 pins in which the extra pulse may or may not be observable for a divide by 1 configuration and for non-divide by 1 configurations may affect the divided clock duty cycle or may corrupt or stall the divided clock waveform.

2.55.2 Workaround

Select other clock sources for the RTC. The RTC supports other clock sources via the RTC_RTCC[CLKSEL] register which include selections for the 32 kHz Slow External Crystal Oscillator (SXOSC), the 16 MHz Fast Internal RC Oscillator (FIRC), or the 8-40 MHz Fast External Crystal Oscillator (FXOSC). If the RTC uses the SIRC clock, then the application needs to manage possible unexpected counter values such as count values that are double the expected value.

If the CMU monitors the SIRC clock, then the application needs to manage possible unexpected measured SIRC frequencies based on possible unexpected counter values such as frequencies that are double the expected value.

The application needs to account for the possibility of a quicker SWT timeout for any SWT that is enabled. Other timer sources such as a PIT timer may be used to compliment the SWT counts.

If the CLKOUTx pin selects the SIRC clock source, then the application needs to manage the possible unexpected CLKOUT waveforms.

2.56 ERR050575: eMIOS: Any Unified Channel running in OPWMCB mode may function improperly if the lead or trail dead time insertion features is used and its timebase is generated by Unified channel in MCB mode

2.56.1 Description

The Unified channel (UC) configured in Center Aligned Output Pulse Width Modulation Buffered (OPWMCB) mode is not working properly when:

1. It’s timebase is sourced from the UC configured in Modulus Counter Buffered (MCB) mode.

2. The lead or trail dead time insertion features is used.

3. Its channel prescaler is different than timebase channel prescaler.

2.56.2 Workaround

Channel configured in OPWMCB mode with lead or trail dead time insertion features enabled must have channel prescaler equal to the timebase channel prescaler configured in MCB mode.

2.57 ERR052152: ENEN/ENET 1G: The hardware acceleration feature for ICMP checksum generation and checking does not work for IPv6.

2.57.1 Description

The hardware acceleration feature for ICMP checksum generation and checking does not work for IPv6. It does work for IPv4.

2.57.2 Workaround

Use software for IPv6 ICMP checksum generation and checking.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Suitability for use in automotive applications — This NXP product has been qualified for use in automotive applications. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as "Critical Applications"), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Contents

For more information, please visit: https://www.nxp.com

Date of release: 5 June 2024
Document identifier: MPC5746C_1N06M_REV_10