NXP Semiconductors

Errata

Document identifier: MPC5643L_2N89D/0N18H

Rev. JUL2023, 7/2023

MPC5643L_2N89D/0N18H

Mask Set Errata

Mask Set Errata for Mask 2N89D/0N18H

Revision History

This report applies to mask 2N89D/0N18H for these products:

- MPC5643L

Table 1. Mask Specific Information

major_mask_rev_num	0x1/0x1
minor_mask_rev_num	0x2/0x3
jtag_id	0x2AEA_301D/0x3AEA_301D

Table 2. Revision History

Revision	Date	Significant Changes
JUL2023	7/2023	The following errata were added. • ERR051698 The following errata were revised. • ERR007394
SEP2022	8/2022	The following errata were revised. • ERR007394
AUG2021	9/2021	The following errata were added. • ERR050782 • ERR051064 • ERR050807 The following errata were revised. • ERR007394
September 2018	8/2018	The following errata were added. • ERR010755
October 2015	5/2018	The following errata have been added: 7589, 8080, 8933, 8970 Initial Revision

Errata and Information Summary

Table 3. Errata and Information Summary

Erratum ID	Erratum Title
ERR003320	Flash: single bit correction status is not available in the Error Correction Status Module (ECSM) and in the Fault Collection and Control Unit (FCCU).

Table continues on the next page...

Table 3. Errata and Information Summary (continued)

Erratum ID	Erratum Title
ERR003511	FlexPWM : Incorrect PWM operation when IPOL is set.
ERR003697	e200z: Exceptions generated on speculative prefetch
ERR004016	ADC: Presampling on channels 9, 10, 15 leads to incorrect results
ERR004166	CTU: FIFO full and concurrent push and pop operations
ERR004168	ADC: Abort switch aborts the ongoing injected channel as well as the upcoming normal channel
ERR004186	ADC: Do not trigger ABORT or ABORTCHAIN prior to the start of CTU triggered ADC conversions and do not trigger ABORTCHAIN prior to the start of INJECTED triggered ADC conversions.
ERR004334	MC_RGM: Device stays in reset state on external reset assertion.
ERR004340	LINFlexD: Buffer overrun can not be detected in UART Rx FIFO mode
ERR006481	NZ4C3/NZ7C3: Erroneous Resource Full Message under certain conditions
ERR006726	NPC: MCKO clock may be gated one clock period early when MCKO frequency is programmed as SYS_CLK/8.and gating is enabled
ERR006967	eDMA: Possible misbehavior of a preempted channel when using continuous link mode
ERR007120	NZxC3: DQTAG implemented as variable length field in DQM message
ERR007274	LINFlexD: Consecutive headers received by LIN Slave triggers the LIN FSM to an unexpected state
ERR007322	FlexCAN: Bus Off Interrupt bit is erroneously asserted when soft reset is performed while FlexCAN is in Bus Off state
ERR007352	DSPI: reserved bits in slave CTAR are writable
ERR007394	MC_ME: Incorrect mode may be entered on low-power mode exit.
ERR007589	LINFlexD: Spurious timeout error when switching from UART to LIN mode or when resetting LINTCSR[MODE] bit in LIN mode
ERR007877	FlexPWM: do not enable the fault filter
ERR008070	SWG: GPIO[55] functionality is not available unless the SWG is powered down
ERR008080	LINFlexD: TX pin gets set to High-Z when in IDLE state
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
ERR009682	SPI: Inconsistent loading of shift register data into the receive FIFO following an overflow event
ERR009849	BAM: Serial boot not supported when flash is programmed with Power Architecture instruction set software
ERR009928	FlexPWM: Half cycle automatic fault clearing does not work in PWM submodule 0 under some conditions
ERR009976	DSPI: Incorrect data received by master with Modified transfer format enabled when using Continuous serial communication clock mode

Table continues on the next page...

Table 3. Errata and Information Summary (continued)

Erratum ID	Erratum Title
ERR010755	DSPI: Transmit and Receive FIFO fill flags in status register is not cleared when DMA is improperly configured
ERR050782	e200: Time Base TBU register contains wrong value during TBL rollover
ERR050807	FLASH: Flash array access may be incorrect after power up
ERR051064	CRC: AutoSAR 4.0 8-bit CRC8 0x2F is not supported in hardware
ERR051698	CTU : Double buffer reload mechanism is blocked when master reload pulse is not generated by Software

Known Errata

ERR003320: Flash: single bit correction status is not available in the Error Correction Status Module (ECSM) and in the Fault Collection and Control Unit (FCCU).

Description

A single bit error correction by the Flash is not passed to the Error Correction Status Module (ECSM) and to the Fault Collection and Control Unit (FCCU). The single bit error correction is only flagged by the SBC bit of the Flash Module Configuration Register (MCR).

Workaround

Poll the SBC bit (Single Bit Correction Status) of the Flash Module Configuration Register (MCR) to detect a single bit error correction event.

ERR003511: FlexPWM : Incorrect PWM operation when IPOL is set.

Description

When IPOL bit is set in complementary mode, the source of the PWMi waveform is supposed to be switched from the VAL2 and VAL3 registers to the VAL4 and VAL5 registers. This switch is not happening.

Workaround

Instead of setting IPOL bt, the application can swap the VAL2/3 values with the VAL4/5 values.

ERR003697: e200z: Exceptions generated on speculative prefetch

Description

The e200z4 core can prefetch up to 2 cache lines (64 bytes total) beyond the current instruction execution point. If a bus error occurs when reading any of these prefetch locations, the machine check exception will be taken. For example, executing code within the last 64 bytes of a memory region such as internal SRAM or FLASH, may cause a bus error when the core prefetches past the end of memory. An ECC exception can occur if the core prefetches locations that are valid, but not yet initialized for ECC.

Workaround

Do not place code to be executed within the last 64 bytes of a memory region. When executing code from internal ECC SRAM, initialize memory beyond the end of the code until the next 32-byte aligned address and then an additional 64 bytes to ensure that prefetches cannot land in uninitialized SRAM.

ERR004016: ADC: Presampling on channels 9, 10, 15 leads to incorrect results

Description

On ADC channels 9 (for factory test only) , 10 (VREG_1.2V), 15 (TSENS) when performing presampling using VSS_HV_ADR (PREVAL0=01) and bypassing the sampling (PRECONV=1) results in an incorrect converted presampled value.

Workaround

ADC Conversion Timing Register 1 (CTR1) and Presampling Control Register (PSCR), field PREVAL1(bits 27:28) can be programmed to select the conversion durations and reference voltages for ADC channels 9, 10, 15.

ERR004166: CTU: FIFO full and concurrent push and pop operations

Description

With a full FIFO, if concurrent FIFO read and write operations occur, then the order of the FIFO is not correct.

For example, FIFO is full and contains data A,B,C,D. Then there are POP and a PUSH requests in the same clock cycle. After the PUSH and POP operations instead of correct data B,C,D,E, the FIFO contains the data B,C,E,D. Data A is pushed out correctly, but data E and D are swapped.

Application software can detect the swap between E and D by reading the CTU.FLx.ADC and CTU.FLx.N_CH fields, unless E and D refer to the same ADC and same channel.

Workaround

To reduce the risk of this issue 2 suggestions are given:

1) lower the FIFO threshold to less than the size of the FIFO (probability is reduced, but can't be fully excluded).

2) forbid 2 consecutive conversions from the same ADC and channel source to allow swap detection by reading the CTU.FLx.ADC and CTU.FLx.N_CH fields.

ERR004168: ADC: Abort switch aborts the ongoing injected channel as well as the upcoming normal channel

Description

If an Injected chain (jch1,jch2,jch3) is injected over a Normal chain (nch1,nch2,nch3,nch4) the Abort switch does not behave as expected.

Expected behavior:

Correct Case (without SW Abort on jch3): Nch1- Nch2(aborted) -Jch1 - Jch2 - Jch3 - Nch2(restored) - Nch3 - Nch4

Correct Case(with SW Abort on jch3): Nch1 - Nch2(aborted) -Jch1 - Jch2 - Jch3(aborted) - Nch2(restored) - Nch3 - Nch4

Observed unexpected behavior:

Fault1 (without SW abort on jch3): Nch1 - Nch2(aborted) - Jch1 - Jch2 - Jch3 - Nch3 - Nch4 (Nch2 not restored)

Fault2 (with SW abort on jch3): Nch1- Nch2 (aborted) - Jch1 - Jch2 - Jch3(aborted) - Nch4 (Nch2 not restored & Nch3 conversion skipped)

Workaround

It is possible to detect the unexpected behavior by using the CEOCFRx register. The CEOCFRx fields will not be set for a not restored or skipped channel, which indicates this issue has occurred. The CEOCFRx fields need to be checked before the next Normal chain execution (in scan mode). The CEOCFRx fields should be read by every ECH interrupt at the end of every chain execution.

ERR004186: ADC: Do not trigger ABORT or ABORTCHAIN prior to the start of CTU triggered ADC conversions and do not trigger ABORTCHAIN prior to the start of INJECTED triggered ADC conversions.

Description

When ADC_MCR[ABORT] or ADC_MCR[ABORTCHAIN] is set prior to the ADC receiving a CTU trigger, the next CTU triggered ADC conversion will not be performed and further CTU triggered ADC conversions will be blocked.

When ADC_MCR[ABORTCHAIN] is set prior to the ADC receiving an INJECTED trigger, the next INJECTED ADC conversion will not be performed. Following the ABORTCHAIN command the MCU behaviour does not meet the specification as ADC_ISR[JECH] is not set and ADC_MCR[ABORTCHAIN] is not cleared.

Workaround

Do not program ADC_MCR[ABORT] or ADC_MCR[ABORTCHAIN] before the start of ADC conversions.

The case when CTU triggered ADC conversions are blocked should be avoided however it is possible to reactivate CTU conversions by clearing and setting ADC_MCR[CTUEN].

ERR004334: MC_RGM: Device stays in reset state on external reset assertion.

Description

On an external reset that is configured to be 'long' the device may remain in reset if the system clock is configured to be sourced by a clock source other than the 16 MHz Internal RC Oscillator (IRCOSC). Recovery from the reset in this case can only be achieved via a power-down and power-up cycle.

The failure condition can only be seen with the following Reset Generation Module (MC_RGM) settings for Functional Event Short Sequence register, External Reset field (RGM_FESS[SS_EXR]) and Functional Bidirectional Reset Enable register, External Reset field (RGM_FBRE[BE_EXR]):

- RGM_FESS[SS_EXR] = 0b0 (long external reset)

- RGM_FBRE[BE_EXR] = 0b0 (asserted on external reset event)

Note 1: This condition can only occur if the cause of the device reset was the external reset assertion. It does not occur if, for example, the device reset was due to a power-on.

Note 2: RGM_FESS[SS_EXR] = 0b0 and RGM_FBRE[BE_EXR] = 0b0 are the default settings out of power-on reset (POR).

Workaround

There are two possible workarounds. In both, the workaround takes effect only after software has reconfigured the MC_RGM. Therefore, in order to ensure that the issue cannot occur after POR exit and before the software has executed the workaround, the system clock must not be re-configured in the Mode Entry module (MC_ME) to be sourced by a clock source other than the IRCOSC until after the workaround has been executed.

Workaround #1:

Always configure the external reset event to prevent the external reset output to be driven by the MC_RGM by writing 0b1 to RGM_FBRE[BE_EXR].

If the external reset has been configured to be long (RGM_FESS[SS_EXR] = 0b0) and self testing has been enabled via the flash option, the external reset pin will still be asserted from the time of external assertion until reset sequence exit after start-up self test execution.

If the external reset has been configured to be long (RGM_FESS[SS_EXR] = 0b0) and self testing has been disabled via the flash option, the external reset pin will still be asserted from the time of external assertion until the chip configuration is loaded from the flash during reset PHASE3.

If the external reset has been configured to be short (RGM_FESS[SS_EXR] = 0b1), the external reset pin will still be released as soon as it is no longer asserted from off-chip.

Workaround #2:

Always configure the external reset as 'short' by writing 0b1 to RGM_FESS[SS_EXR]. In addition, use software to trigger a long 'functional' or 'destructive' reset via the Mode Entry module (MC_ME) if flash initialization or start-up self test is required.

The impact of this workaround is the additional time that the device is in reset (due to the short reset sequence triggered by the external reset) and the overhead required for software to check the reset status and request a software reset.

ERR004340: LINFlexD: Buffer overrun can not be detected in UART Rx FIFO mode

Description

When the LINFlexD is configured in UART Receive (Rx) FIFO mode, the Buffer Overrun Flag (BOF) bit of the UART Mode Status Register (UARTSR) register is cleared in the subsequent clock cycle after being asserted.

User software can not poll the BOF to detect an overflow.

The LINFlexD Error Combined Interrupt can still be triggered by the buffer overrun. This interrupt is enabled by setting the Buffer Overrun Error Interrupt Enable (BOIE) bit in the LIN Interrupt enable register (LINIER). However, the BOF bit will be cleared when the interrupt routine is entered, preventing the user from identifying the source of error.

Workaround

Buffer overrun errors in UART FIFO mode can be detected by enabling only the Buffer Overrun Interrupt Enable (BOIE) in the LIN interrupt enable register (LINIER).

ERR006481: NZ4C3/NZ7C3: Erroneous Resource Full Message under certain conditions

Description

The e200zx core Nexus interface may transmit an erroneous Resource Full Message (RFM) following a Program Trace history message with a full history buffer. The History information for both of the messages are the same and the RFM should have not been transmitted. This occurs when the instruction following the indirect branch instruction (which triggers the Program Trace History message) would have incremented the History field. The instructions must be executed in back to back cycles for this problem to occur. This is the only case that cases this condition.

Workaround

There are three possible workarounds for this issue.

(1) Tools can check to see if the Program Trace History message and proceeding Resource Full Message (RFM) have the same history information. If the history is the same, then the RFM is extraneous and can be ignored.

(2) Code can be rewritten to avoid the History Resource Full Messages at certain parts of the code. Insert 2 NOP instructions between problematic code. Or inserting an "isync" or a indirect branch some where in the code sequence to breakup/change the flow.

(3) If possible, use Traditional Program Trace mode can be used to avoid the issue completely. However, depending on other conditions (Nexus port width, Nexus Port speed, and other enabled trace types), overflows of the port could occur.

ERR006726: NPC: MCKO clock may be gated one clock period early when MCKO frequency is programmed as SYS_CLK/8.and gating is enabled

Description

The Nexus auxiliary message clock (MCKO) may be gated one clock period early when the MCKO frequency is programmed as SYS_CLK/8 in the Nexus Port Controller Port Configuration Register (NPC_PCR[MCKO_DIV]=111) and the MCKO gating function is enabled (NPC_PCR[MCKO_GT]=1). In this case, the last MCKO received by the tool prior to the gating will correspond to the END_MESSAGE state. The tool will not receive an MCKO to indicate the transition to the IDLE state, even though the NPC will transition to the IDLE state internally. Upon re-enabling of MCKO, the first MCKO edge will drive the Message Start/End Output (MSEO=11) and move the tool's state to IDLE.

Workaround

Expect to receive the MCKO edge corresponding to the IDLE state upon re-enabling of MCKO after MCKO has been gated.

ERR006967: eDMA: Possible misbehavior of a preempted channel when using continuous link mode

Description

When using Direct Memory Access (DMA) continuous link mode Control Register Continuous Link Mode (DMA_CR[CLM]) = 1) with a high priority channel linking to itself, if the high priority channel preempts a lower priority channel on the cycle before its last read/write sequence, the counters for the preempted channel (the lower priority channel) are corrupted. When the preempted channel is restored, it continues to transfer data past its "done" point (that is the byte transfer counter wraps past zero and it transfers more data than indicated by the byte transfer count (NBYTES)) instead of performing a single read/write sequence and retiring.

The preempting channel (the higher priority channel) will execute as expected.

Workaround

Disable continuous link mode (DMA_CR[CLM]=0) if a high priority channel is using minor loop channel linking to itself and preemption is enabled. The second activation of the preempting channel will experience the normal startup latency (one read/write sequence + startup) instead of the shortened latency (startup only) provided by continuous link mode.

ERR007120: NZxC3: DQTAG implemented as variable length field in DQM message

Description

The e200zx core 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.

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

Description

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

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

Workaround

The following three steps should be followed -

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

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

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

Note:

THeader_Nominal = 34 * TBit

TResponse_Nominal = 10 * (NData + 1) * TBit

THeader_Maximum = 1.4 * THeader_Nominal

TResponse_Maximum = 1.4 * TResponse_Nominal

TFrame_Maximum = THeader_Maximum + TResponse_Maximum

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

ERR007322: FlexCAN: Bus Off Interrupt bit is erroneously asserted when soft reset is performed while FlexCAN is in Bus Off state

Description

Under normal operation, when FlexCAN enters in Bus Off state, a Bus Off Interrupt is issued to the CPU if the Bus Off Mask bit (CTRL[BOFF_MSK]) in the Control Register is set. In consequence, the CPU services the interrupt and clears the ESR[BOFF_INT] flag in the Error and Status Register to turn off the Bus Off Interrupt.

In continuation, if the CPU performs a soft reset after servicing the bus off interrupt request, by either requesting a global soft reset or by asserting the MCR[SOFT_RST] bit in the Module Configuration Register, once MCR[SOFT_RST] bit transitions from 1 to 0 to acknowledge the soft reset completion, the ESR[BOFF_INT] flag (and therefore the Bus Off Interrupt) is re-asserted.

The defect under consideration is the erroneous value of Bus Off flag after soft reset under the scenario described in the previous paragraph.

The Fault Confinement State (ESR[FLT_CONF] bit field in the Error and Status Register) changes from 0b11 to 0b00 by the soft reset, but gets back to 0b11 again for a short period, resuming after certain time to the expected Error Active state (0b00). However, this late correct state does not reflect the correct ESR[BOFF_INT] flag which stays in a wrong value and in consequence may trigger a new interrupt service.

Workaround

To prevent the occurrence of the erroneous Bus Off flag (and eventual Bus Off Interrupt) the following soft reset procedure must be used:

1. Clear CTRL[BOFF_MSK] bit in the Control Register (optional step in case the Bus Off Interrupt is enabled).

2. Set MCR[SOFT_RST] bit in the Module Configuration Register.

3. Poll MCR[SOFT_RST] bit in the Module Configuration Register until this bit is cleared.

4. Wait for 4 peripheral clocks.

5. Poll ESR[FLTCONF] bit in the Error and Status Register until this field is equal to 0b00.

6. Write "1" to clear the ESR[BOFF_INT] bit in the Error and Status Register.

7. Set CTRL[BOFF_MSK] bit in the Control Register (optional step in case the Bus Off Interrupt is enabled).

ERR007352: DSPI: reserved bits in slave CTAR are writable

Description

When the Deserial/Serial Peripheral Interface (DSPI) module is operating in slave mode (the Master [MSTR] bit of the DSPI Module Configuration Register [DSPIx_MCR] is cleared), bits 10 to 31 (31 = least significant bit) of the Clock and Transfer Attributes Registers (DSPIx_CTARx) should be read only (and always read 0). However, these bits are writable, but setting any of these bits to a 1 does not change the operation of the module.

Workaround

There are two possible workarounds.

Workaround 1: Always write zeros to the reserved bits of the DSPIx_CTARn_SLAVE (when operating in slave mode).

Workaround 2: Mask the reserved bits of DSPIx_CTARn_SLAVE when reading the register in slave mode.

ERR007394: MC_ME: Incorrect mode may be entered on low-power mode exit.

Description

For the case when the Mode Entry (MC_ME) module is transitioning from a run mode (RUN0/1/2/3) to a low power mode (HALT/STOP/STANDBY*) if a wake-up or interrupt is detected one clock cycle after the second write to the Mode Control (ME_MCTL) register, the MC_ME will exit to the mode previous to the run mode that initiated the low power mode transition.

Example correct operation DRUN->RUN1-> RUN3->STOP->RUN3

Example failing operation DRUN->RUN1-> RUN3->STOP->RUN1

Note STANDBY mode is not available on all MPC56xx microcontrollers

Workaround

To ensure the application software returns to the run mode (RUN0/1/2/3) prior to the low power mode (HALT/STOP/STANDBY*) it is required that the RUNx mode prior to the low power mode is entered twice.

The following example code shows RUN3 mode entry prior to a low power mode transition.

```
ME.MCTL.R = 0x70005AF0; /* Enter RUN3 Mode & Key */

ME.MCTL.R = 0x7000A50F; /* Enter RUN3 Mode & Inverted Key */

while (ME.GS.B.S_MTRANS) {} /* Wait for RUN3 mode transition to complete */

ME.MCTL.R = 0x70005AF0; /* Enter RUN3 Mode & Key */

ME.MCTL.R = 0x7000A50F; /* Enter RUN3 Mode & Inverted Key */

while (ME.GS.B.S_MTRANS) {} /* Wait for RUN3 mode transition to complete */

/* Now that run mode has been entered twice can enter low power mode */

/* (HALT/STOP/STANDBY*) when desired. */
```

ERR007589: LINFlexD: Spurious timeout error when switching from UART to LIN mode or when resetting LINTCSR[MODE] bit in LIN mode

Description

If the LINFlexD module is enabled in Universal Asynchronous Receiver/Transmitter (UART) mode and the value of the MODE bit of the LIN Timeout Control Status register (LINTCSR) is 0 (default value after reset), any activity on the transmit or receive pins will cause an unwanted change in the value of the 8-bit field Output Compare Value 2 (OC2) of the LIN Output Compare register (LINOCR).

If the LINFlexD module is enabled in LIN mode and the value of the MODE bit of the LIN Timeout Control Status register (LINTCSR) is changed from ‘1’ to ‘0’, then the old value of the Output Compare Value 1 (OC1) and Output Compare Value 2 (OC2) of the LIN Output Compare register (LINOCR) is retained.

As a consequence, if the module is reconfigured from UART to Local Interconnect Network (LIN) mode, or LINTCSR MODE bit is changed from ‘1’ to ‘0’, an incorrect timeout exception is generated when the LIN communication starts.

Workaround

If the LINFlexD module needs to be switched from UART mode to LIN mode, before writing UARTCR[UART] to 1, ensure that the LINTCSR[MODE] is first set to 1.

If the LINFlexD module is in LIN mode and LINTCSR[MODE] needs to be switched from 1 to 0 in between frames, the LINOCR must be set to 0xFFFF by software.

ERR007877: FlexPWM: do not enable the fault filter

Description

Operation of the fault pin filter of the Flexible Pulse Width Modulation (FLEX_PWM) may be inconsistent if the Fault Filter is enabled, by setting the Filter Period greater than zero in the Fault Filter register (FFILT[FILT_PER] > 0). The fault filter flag may be set even though the pulse is shorter than the filter time.

Workaround

Do not enable the PWM fault pin filters. Disable the fault pin filters by setting the Fault Filter Period to 0 in the Fault Filter Register (FFILT[FILT_PER] = 0).

ERR008070: SWG: GPIO[55] functionality is not available unless the SWG is powered down

Description

The General Purpose Input/Output 55 (GPIO[55]) functionality on port D[7] is disabled if the Sine Wave Generator module (SWG) is not in power down mode. The SWG will not enter power down mode if the SWG clock input is disabled.

Workaround

Ensure that the SWG clock input is enabled via the Aux Clock 0 Divider Configuration 1 register (CGM_AC0_DC1[DE1]=1) prior to putting the SWG in power down mode in the SWG control register (SWG_CTRL[PDS] = 1). This will allow GPIO[55] functionality on port D[7].

ERR008080: LINFlexD: TX pin gets set to High-Z when in IDLE state

Description

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

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

Workaround

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

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

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

Description

When the LINFlexD module is configured as follows:

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

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

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

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

Workaround

There are 2 possible workarounds.

Workaround 1:

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

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

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

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

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

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

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

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

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

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

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

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

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

Workaround 2:

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

ERR008970: 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)

ERR009682: SPI: Inconsistent loading of shift register data into the receive FIFO following an overflow event

Description

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

Workaround

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

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

ERR009849: BAM: Serial boot not supported when flash is programmed with Power Architecture instruction set software

Description

The Boot Assist Module (BAM) will not support serial boot when flash memory is programmed with Power Architecture instruction set (Book E) software. The functionality of the BAM is dependent on the value of the Variable Length Encoding (VLE) bit in the Reset Configuration Half Word (RCHW) of the bootable flash sector. If the VLE bit = 0, indicating that Book E software is programmed in flash memory, the BAM will not function correctly and serial boot will not be supported.

Workaround

Erase flash memory or program flash memory with VLE software prior to executing serial boot with Book E software.

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

Description

When

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

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

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

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

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

Workaround

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

ERR009976: DSPI: Incorrect data received by master with Modified transfer format enabled when using Continuous serial communication clock mode

Description

When the Deserial Serial Peripheral Interface (DSPI) module is configured as follows:

1. Master mode is enabled (Master/Slave Mode Select bit in Module Configuration Register is set (DSPI_MCR [MSTR] = 0b1))

2. Modified transfer format is enabled (Modified Transfer Format Enable bit in Module Configuration Register is set (DSPI_MCR [MTFE] = 0b1))

3. Continuous serial communication clock mode is enabled (Continuous SCK Enable bit in Module Configuration Register is set (DSPI_MCR [CONT_SCKE] = 0b1))

In this configuration if the frame size of the current frame is greater than the frame size of the next received frame, corrupt frames are received in two scenarios:

a) Continuous Peripheral Chip Select Enable bit in PUSH TX FIFO Register is set (DSPI_PUSHR [CONT] = 0b1)

b) DSPI_PUSHR [CONT] = 0b0 and lower significant bit of the frame is transferred first (LSB first bit in Clock and Transfer Attributes Register is set (DSPI_CTAR [LSBFE] =0b1))

Workaround

To receive correct frames:

a) When DSPI_PUSHR [CONT] = 0b1, configure the frame size of the current frame less than or equal to the frame size of the next frame (for all frames).

b) When DSPI_PUSHR [CONT] = 0b0, configure DSPI_CTAR [LSBFE] = 0b0. Alternatively, configure the frame size of the current frame less than or equal to the frame size of the next frame (for all frames).

Make sure that for all received frames, the bits are read equal to their respective frame sizes and any extra bits during POP operation are masked.

ERR010755: DSPI: Transmit and Receive FIFO fill flags in status register is not cleared when DMA is improperly configured

Description

The Deserial/Serial Peripheral Interface Transmit and Receive First In/First Out (FIFO) buffers can request additional information to be transferred via the Direct Memory Access (DMA) module when either the Transmit or Receive FIFO Fill/Drain Flags are set in the DSPI Status Register (SR[TFFF/RFDF]). However, the Transmit Fill Flag indicates that at least 1 location each (2 bytes each) in the Transmit and Command FIFOs 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 (2 bytes) 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/Drain Flags may not be properly cleared indicating that the FIFO is not full even when the FIFO is actually full (for Transmit FIFO) and not empty when the FIFO is actually empty (for Receive FIFO).

Workaround

Properly configure the DMA to fill/drain only 2 bytes to Transmit, Command and Receive FIFOs. Use the DMA loop to transfer more data if needed.

ERR050782: e200: Time Base TBU register contains wrong value during TBL rollover

Description

The e200 Time Base (TB) facility is a 64-bit structure provided for maintaining the time of day and operating interval timers. The TB consists of two 32-bit registers - time base upper (TBU) and time base lower (TBL). TBU and TBL are concatenated to provide a long-period 64-bit counter. TBL increments until its value becomes 0xFFFF_FFFF. The intended behavior is that at the next increment when the TBL value becomes 0x0000_0000 that the TBU value is incremented. But the actual behavior is that after the TBL value becomes 0x0000_0000, the TBU value will not increment until the transition of the TBL value to 0x0000_0001.

Workaround

Software will need to take care about the wrong TBU value during TBL rollover. Use the following sequence for reading TBU and TBL values:

```
loop:
mfspr r12, TBL
mfspr r3, TBU
mfspr r4, TBL
cmpl r4,r12
se_blt loop
```

ERR050807: FLASH: Flash array access may be incorrect after power up

Description

After power up, a flash read may return incorrect data or a program/erase operation may fail. The condition will persist from power up until the next reset assertion.

Workaround

Flash read: Software Watchdog Timer (SWT) needs to be enabled at the time the micro exits its reset sequence after power up which provides a recovery mechanism if code execution fails.

Flash program/erase: After power up, the device must be reset before performing flash program/erase operations. However, through proper usage of software mechanisms such as EE emulation algorithms in addition to Error Correction Code (ECC), the probability of observing a failure due to ineffective programing or erasing will be greatly reduced.

ERR051064: CRC: AutoSAR 4.0 8-bit CRC8 0x2F is not supported in hardware

Description

The Cyclic Redundancy Check (CRC) module does not implement the 8-bit CRC-8-H2F required to support the Autosar 4.0 specification. The CRC-8-H2F uses a polynomial generator seed of 0x2F and an equation of $x^{5} + x^{3} + x^{2+x+1.}$

Workaround

The 8-bit CRC-8-H2F function must be written in software to support AutoSAR 4.0.

ERR051698: CTU : Double buffer reload mechanism is blocked when master reload pulse is not generated by Software

Description

CTU operates on two clock domains. The Bus Interface Clock (BIC) domain, used for SW communication and the Module Clock (MC) domain, used for its own functionality. The FGRE bit of CR register is set with first write into double buffered registers in BIC domain and the synchronization logic propagates this set bit into the MC domain. FGRE bit is cleared in both domains when the MR occurs and FGRE bit is equal to GRE bit in the MC domain. Double buffered registers are reloaded only when MR occurs and FGRE and GRE bit are set in the MC domain. But when the MR occurs at the same time as FGRE bit set, generated by first double buffered register write, is propagated into the MC domain the FGRE bits are cleared in both.

Two possible cases can happen:

1) In case of updating one double buffered register the reload doesn’t occur and the CR register is kept in 0x2 value which blocks further double buffered register update.

2) In case of updating more than one double buffered register the reload doesn’t occur and the CR register is kept in 0xA when the delay between first and second double buffered register write is less than the synchronization propagation time which blocks further register update.

Workaround

1) Generate the Master reload pulse by SW (by setting the bit CR[MRS_SG] after setting GRE at the end of update sequence)

2) Master reload pulse is not generated by SW and one double buffered register to be updated:

The register needs to be written twice with the delay between the writes longer than synchronization propagation time.

3) Master reload pulse is not generated by SW and more than one double buffered register to be updated:

The delay between write to first and second double buffered register needs to be longer than synchronization propagation time.

Synchronization propagation time = (6*BIC) cycles + (5*MC) cycles, where BIC is Bus Interface Clock period and MC is Module Clock period.

Legal information

Trademarks

Date of release: 7/2023

Document identifier: MPC5643L_2N89D/0N18H