Mask Set Errata for Mask REVA

Introduction

This report applies to mask REVA for these products:

- MPC5561

ID before 15 MAY 2008	ID from 15 May 2008 to 30 JUNE 2010	ID after 1 JULY 2010	Errata Title
N/A	12488	2279	BAM: Pull RXD_A high during CAN serial boot mode
2297	7062	1722	BAM: Serial download unavailable to last 16 bytes (4 words) of System RAM
6049	5028	224	BAM: VLE added to the LENGTH field during serial boot message
1123	6934	575	DSPI: Changing CTARs between frames in continuous PCS mode causes error
4022	11100	1154	DSPI: DSPI_B pins split to separate supply, VDDEH10
N/A	10483	1103	DSPI: PCS Continuous Selection Format limitation
3397	5742	1812	DSPI: Serialization (DSI) features not implemented
2264	11097	1147	DSPI: Using DSPI in DSI mode with MTO may cause data corruption
N/A	9739	1082	DSPI: set up enough ASC time when MTFE=1 and CPHA=1
1874	6869	1531	EBI: Additional Address lines available
2823	7049	1708	EBI: Do not access external resources when the EBI is disabled
3111	11099	1151	EBI: Dual controller mode cannot be guaranteed under all conditions
3839	5762	1844	EBI: Timed out accesses (external TA only) may generate spurious TS_B pulse
3819	5715	1741	EQADC : 25% calibration channel sampling requires at least 64 sampling cycles
1742	6968	652	EQADC: 50% reference channels reads 20 mv low
2371	7439	2075	FLASH: Large blocks limited to 1,000 Program/erase cycles
2419	6946	605	FLASH: Minimum Programming Frequency is 25 MHz

Table continues on the next page...

ID before 15 MAY 2008	ID from 15 May 2008 to 30 JUNE 2010	ID after 1 JULY 2010	Errata Title
1745	7444	2083	FLASH: The ADR register may get loaded with a flash address even through no ECC error has occurred
715	6764	1111	FMPLL: LOLF can be set on MFD change
N/A	8014	2181	FMPLL: Non-zero pre-divider values can cause PLL lock failure
4527	5874	507	FMPLL: Oscillator Gain Increased
N/A	18087	1232	FMPLL: Reset may not be negated if an external reset occurs during a software initiated PLL relock sequence
		6531	FMPLL: Selecting GPIO mode on RSTCFG/PLLCFG[0:1] may cause PLL failure after a reset
		3407	FlexCAN: CAN Transmitter Stall in case of no Remote Frame in response to Tx packet with RTR=1
4414	5952	1557	FlexCAN: Corrupt ID may be sent in early-SOF condition
N/A	23304	2685	FlexCAN: Module Disable Mode functionality not described correctly
3617	5729	1794	FlexCAN: New Feature - Individual Message Mask Registers
3567	5804	352	FlexCAN: New feature - Individual RX matching and Message Queuing
3566	6413	1654	FlexCAN: New feature - Self reception disable
3138	5790	1895	FlexCAN: New feature - Transmit (TX)/Receive (RX) Warning Interrupts
N/A	34749	2424	FlexCAN: switching CAN protocol interface (CPI) to system clock has very small chance of causing the CPI to enter an indeterminate state
3657	5740	1808	FlexRay: A boundary violation frame followed by a valid startup frame during the startup phase may cause an abort of the startup
N/A	12733	2302	FlexRay: Message Buffer can not be disabled and not locked after CHI command FREEZE
N/A	13967	2344	FlexRay: Repeated wakeup after leading coldstart has too short WAKEUP_LISTEN phase
4128	5945	1545	FlexRay: Slot Status of Double Transmit Message Buffers updated incorrectly
5891	6036	1958	FlexRay: System memory access immediately timed out if SYMATOR[TIMEOUT] set to 0x1F
5890	6415	1657	FlexRay: System memory overwritten or invalid data transmitted after timeout of system memory read access
N/A	36635	1429	Flexray : No reception in slot in dynamic segment
6034	6266	789	MPC5561: SIU_MIDR[PARTNUM] is 5561, [MASKNUM] is 0x01, DID[PIN]=0x161
4387	5892	547	MPC5561: VSTBY pin current is higher than expected
N/A	40092	6726	NPC: MCKO clock may be gated one clock period early when MCKO frequency is programmed as SYS_CLK/8.and gating is enabled
1800	7216	1957	NPC: MCKO_DIV can be set to 0x0 (1X MCKO)
2273	7451	2096	NZ6C3: No sync message generated after 255 direct branch messages in history mode
2706	7456	2103	NZ6C3: Data Trace of stmw instructions may cause overruns

Table continues on the next page...

ID before 15 MAY 2008	ID from 15 May 2008 to 30 JUNE 2010	ID after 1 JULY 2010	Errata Title
108	6899	1618	NZ6C3: No indication of an exception causing a Nexus Program Trace (PT) message as opposed to a retired branch instruction causing a PT message.
3545	5836	424	Pad Ring: FlexRay pins changed to MH type
63	6700	488	Pad Ring: Possible poor system clock just after POR negation.
64	7224	1969	Pad Ring: RSTOUT is 3-stated during the power-on sequence.
3644	5814	375	Pad Ring: SCI_A available on CAN_A pins (GPIO[83:84])
		3377	Pad Ring:Nexus pins may drive an unknown value immediately after power up but before the 1st clock edge
3685	5724	1779	SIU: CRSE bit added to the SIU Configuration Register
2854	6961	637	SIU: ECCR[ENGDIV] field extended to eight bits
3276	5707	1721	SIU: ENGCLK alternate source select bit added to SIU_ECCR
N/A	35704	1419	SIU: Reverting ENGCLK source to the system clock has a very small chance of causing the ENGCLK generator to enter an indeterminate state
N/A	8913	1031	e200z6: Cache Invalidate/Clear Aborts due to masked core Interrupt
507	7419	2044	e200z6: Core renamed from e500z6
2312	7068	1729	e200z6: MMU has 32 Table Entries
5256	6262	779	e200z6: New SPE Instructions Added
4075	5971	1581	e200z6: VLE added to core
3565	7356	860	e200z6: Way Access Mode bit added to the Cache
		3421	e200z: Cache returns value, even when disabled
5093	11091	1136	eDMA: BWC setting may be ignored between 1st and 2nd transfers and after the last write of each minor loop.
		6966	eDMA: Possible misbehavior of a preempted channel when using continuous link mode
4535	5920	617	eDMA: Some new features do not function as specified
2305	7147	1834	eMIOS: OPMWC unable to produce close to 100% duty cycle signal
4050	5953	1558	eQADC: Leakage if any other supply/pin is greater than 0.6V above VDDA
2878	7114	1773	eQADC: conversions of muxed digital/analog channels close to the rail
N/A	8632	1013	eSCI : Automatic reset of the LIN state machine cause incorrect transmission
N/A	27326	1381	eSCI: LIN Wakeup flag set after aborted LIN frame transmission
N/A	17048	1221	eSCI: LIN bit error indicated at start of transmission after LIN reset
N/A	8635	1017	eSCI: LIN fast bit error detection causes incorrect LIN reception
N/A	8173	2190	eSCI: LIN slave timeout flag STO not asserted if CRC is received too late
7064	4968	1614	eSCI: Low pulse on LIN RX line may prevent assertion of transmit data ready flag ESCI_SR[TXRDY]

e2279: BAM: Pull RXD_A high during CAN serial boot mode

Errata type: Errata

Description: The Boot Assist Module (BAM) disables the internal pull up resistor on serial port A receive data pin (eSCI RXD_A) during serial boot mode operation. If the pin is not actively driven by system, its input level may drift below low threshold and be recognized as a valid eSCI boot operation thus preventing CAN boot mode selection.

Workaround: Always drive the eSCI RXD_A pin high during CAN serial boot mode. A external 10K pullup resistor can be used for this purpose.

e1722: BAM: Serial download unavailable to last 16 bytes (4 words) of System RAM

Errata type: Information

Description: When using the BAM Serial boot download feature, the BAM initializes an additional 4 32-bit words after the end of the downloaded records. This is done to insure that if the core fetches the last instruction of the downloaded code from the internal SRAM while executing the code, it will not prefetch instructions from memory locations that have not been initialized.

Note: if the download image has the exact same size as the internal SRAM, the 20 bytes at the beginning of the SRAM will be written with zero value due to incomplete memory decoding.

Workaround: When using the Serial download feature of the BAM, make sure that the maximum address of the downloaded code does not exceed the end address of the SRAM minus 16 bytes or the last address of the Memory Management Unit (MMU) entry minus 16 bytes (for devices with MMU and the SRAM MMU setting less than the full SRAM size), whichever is smaller.

e224: BAM: VLE added to the LENGTH field during serial boot message

Errata type: Information

Description: In serial boot mode, tools download 3 pieces of information: a 64-bit password, followed by a 32-bit start address, and then a 32-bit download length (LENGTH). On devices that support the Variable Length Encoded (VLE) instruction set, the 32-bit LENGTH field has been changed to a 1-bit VLE bit followed by a 31-bit LENGTH. The VLE bit replaces what was the MSB of the LENGTH. Setting the VLE bit to 1 indicates that the downloaded code should be run in VLE mode. Leaving VLE a 0 indicates that the downloaded code should be run in BookE/classic Power Architecture instruction set mode. When the VLE bit is set to 1 the BAM programs EBI, RAM and Flash MMU TLB entries (# 1,2 and 3) with the VLE attribute.

Workaround: Set the VLE bit (MSB of the 32-bit LENGTH) in the serial boot download data if the code being downloaded uses (was written in) VLE instructions.

e575: DSPI: Changing CTARs between frames in continuous PCS mode causes error

Errata type: Errata

Description: Erroneous data could be transmitted if multiple Clock and Transfer Attribute Registers (CTAR) are used while using the Continuous Peripheral Chip Select mode (DSPIx_PUSHR[CONT=1]). The conditions that can generate an error are:

1) If DSPIx_CTARn[CPHA]=1 and DSPIx_MCR[CONT_SCKE = 0] and DSPIx_CTARn[CPOL, CPHA, PCSSCK or PBR] change between frames.

2) If DSPIx_CTARn[CPHA]=0 or DSPIx_MCR[CONT_SCKE = 1] and any bit field of DSPIx_CTARn changes between frames except DSPIx_CTARn[PBR].

Workaround: When generating DSPI bit frames in continuous PCS mode, adhere to the aforementioned conditions when changing DSPIx_CTARn bit fields between frames.

e1154: DSPI: DSPI_B pins split to separate supply, VDDEH10

Errata type: Information

Description: The DSPI_B SINB, SOUTB, SCKB, PCS_B[0:2] were separated from the VDDEH6 and are now powered by the new power supply pin VDDEH10. Ball J23 on the 416 package was changed from being a duplicate VDDEH6 pin to being a separate VDDEH10 supply pin. 324 pin package drawings show the VDDE10 ball placement. VDDEH6 and VDDEH10 are combined/shorted internally on 208 packages.

Workaround: For compatibility to the MPC5554, always power VDDEH6 and VDDEH10 from the same power supply (3.0 to 5.25 volts). If compatibility is not required to the MPC5554, VDDEH10 and VDDEH6 can be supplied by different voltage supplies. This allows one DSPI to operate at a different voltage than the other DSPI modules (3.3 and 5 volts, for example).

e1103: DSPI: PCS Continuous Selection Format limitation

Errata type: Errata

Description: When the DSPI module has more than one entry in the TX FIFO and only one entry is written and that entry has the CONT bit set, and continuous SCK clock selected the PCS levels may change between transfer complete and write of the next data to the DSPI_PUSHR register.

For example:

If the CONT bit is set with the first PUSHR write, the PCS de-asserts after the transfer because the configuration data for the next frame has already been fetched from the next (empty) fifo entry. This behavior continues till the buffer is filled once and all CONT bits are one.

Workaround: To insure PCS stability during data transmission in Continious Selection Format and Continious SCK clock enabled make sure that the data with reset CONT bit is written to DSPI_PUSHR register before previous data sub-frame (with CONT bit set) transfer is over.

e1812: DSPI: Serialization (DSI) features not implemented

Errata type: Information

Description: The DSPI serialization/deserialization features (for serializing slow eMIOS and eTPU signals) are not implemented.

Workaround: Do not set the DSPI Configuration bits in the DSPI Module Configuration register (DSPIx_MCR) to use the Deserial Serial Interface (DSI) features of the DSPI by setting the DCONF to 0b01 (DSI mode) or 0b10 (combined SPI and DSI mode). Always set DCONF to 0b00, selecting normal SPI configuration.

e1147: DSPI: Using DSPI in DSI mode with MTO may cause data corruption

Errata type: Errata

Description: Using the DSPI in Deserial Serial Interface (DSI) Configuration (DSPIx_MCR[DCONF]=0b01]) with multiple transfer operation (DSPIx_DSICR[MTOE=1]) enabled, may cause corruption of data transmitted out on the DSPI master if the clock Phase is set for leading edge capture DSPIx_CTARn[CPHA]=0. The first bit shifted out of the master DSPI into the slave DSPI module will be corrupted and will convert a '0' to read as a '1'.

Workaround: There are three possible workarounds for this issue.

1) Select CPHA=1 if suitable for external slave devices.

2) Set first bit to '1', or ignore first bit. This may not be a workable solution if this bit is required.

3) Connect SOUT from the master to SIN of the first slave externally instead of using internal signals. This is achieved by setting the DSPI Input Select Register (SIU_DISR) to set the SINSELx field of the first slave DSPI to '00' and configuring this slave's SIN pin and master SOUT pin as DSPI SIN/SOUT functions respectively. This workaround is suitable only if these two signals are available to be connected externally to each other.

e1082: DSPI: set up enough ASC time when MTFE=1 and CPHA=1

Errata type: Information

Description: When the DSPI is being used in the Modified Transfer Format mode (DSPI_MCR[MTFE]=1) with the clock phase set for Data changing on the leading edge of the clock and captured on the following edge in the DSPI Clock and Transfer Attributes Register (DSPI_CTARn[CPHA]=1), if the After SCK delay scaler (ASC) time is set to less than 1/2 SCK clock period the DSPI may not complete the transaction - the TCF flag will not be set, serial data will not received, and last transmitted bit can be truncated.

Workaround: If the Modified Transfer Format mode is required DSPI_MCR[MTFE]=1 with the clock phase set for serial data changing on the leading edge of the clock and captured on the following edge in the SCK clock (Transfer Attributes Register (DSPI_CTARn[CPHA]=1) make sure that the ASC time is set to be longer than half SCK clock period.

e1531: EBI: Additional Address lines available

Errata type: Information

Description: Two additional address lines (ADDR6 and ADDR7) have been added to the External Bus Interface (EBI). These extra address lines are multiplexed with ADDR30 and ADDR31 as alternate functions and can be selected by the Pin Assignment (PA) field of SIU_PCR[26] and SIU_PCR[27].

Workaround: Customers should be aware that not all members of the MPC5500 family with an external address bus have these extra 2 address lines.

Note external masters still use ADDR30 and ADDR31 for internal accesses and always use ADDR8 though ADDR31 in this case.

e1708: EBI: Do not access external resources when the EBI is disabled

Errata type: Information

Description: When the external bus is disabled in the External Bus Interface Module Control Register (EBI_MCR[MDIS] = 1), accesses through the EBI will not terminate and the master requesting the access will not request another one.

Workaround: Do not disable the EBI or do not allow accesses to the external bus through Memory Management Unit (MMU) settings in the core. Other internal bus masters (such as DMA) bypasses the MMU and therefore these accesses will hang the external bus if the destination is in the external bus address map.

e1151: EBI: Dual controller mode cannot be guaranteed under all conditions

Errata type: Errata

Description: In dual controller mode, the specification for the phase relationship between EXTAL and CLKOUT is +/- 1 ns, however this does not allow adequate set up and hold times to guarantee successful operation of the external bus to a second MCU.

Workaround: Do not use in Dual Controller mode.

e1844: EBI: Timed out accesses (external TA only) may generate spurious TS_B pulse

Errata type: Errata

Description: When an external Transfer Acknowledge (TA) access times out, there is a boundary case where the External Bus Interface (EBI) asserts a Transfer Start (TS) pulse as if starting another access, even if no other internal request is pending. The boundary case is when the access is part of a "small access" set (sequence of external accesses to satisfy 1 internal request), and when the external TA arrives around the same cycle (+/- 1 clkout cycle) as the bus monitor timeout (BMT).

Most EBI signals will stay negated during this erroneous transfer (CS, OE, WE, BDIP). However, along with TS assertion, RD_WR may also assert (for 1 cycle only, during this phantom TS), if the prior access that timed out was a write. This condition can generate an erroneous write transfer (with CS negated). The address (ADDR pins) will be incremented to the address of the next small access transfer that would have been performed, and the value driven by the EBI on the DATA bus (if a write) may change. Busy Busy (BB) may be asserted along with the phantom TS (if external master modes is enabled in the EBI Module configuration Register, SIU_MCR[EXTM]=1), and the Transfer Size (TSIZ) value may change.

Internally, the EBI terminates the timeout access, and the internal state machine goes to IDLE after the timeout access. So the EBI will not be "hung" after the spurious TS, and the EBI does respond properly to future internal or external requests.

However, the side effect of the spurious TS is that it may cause an external non-chip-select device to think an access is being performed to it, resulting in 1 of 2 bad effects (depending on RD_WR value during spurious TS):

1) RD_WR high (read): ext. device may drive back read data some number of cycles later, possibly conflicting with a future real access (e.g. write) that might have started by that time.

2) RD_WR low (write): ext. device may get an erroneous write performed to it

Note that the soonest possible TS for a real transfer (after the timeout transfer), is 2 cycles after the spurious TS (so 1 cycle gap), meaning this Bug will never result in a 2-cycle TS pulse.

Workaround: Do not enable bus monitor in the EBI Bus Monitor Control Register (keep SIU_BMCR[BME]=0), unless at least 1 of the following 3 conditions can be met:

1) The external TA will never be asserted from external device within 1 cycle of when the access would be timing out (see NOTE below)

2) No internal requests greater than external bus size will be performed (e.g. doing data-only fetches of 32 bits or less on 32-bit data bus or 16 bits or less on a 16 bit bus only, so a "small access" could never occur).

3) The side effect of this TS pulse driven to non-CS device is judged to be tolerable in system after a timeout error occurs; depends on spec of external device and user requirements for data coherency after a timeout error occurs.

NOTE: Of the 3 above, #1 is easiest to achieve in most systems. If the maximum possible TA latency of the external device is known, the user just needs to set the BMT period more than (external device maximum latency + 2), and this condition will not occur.

e1741: EQADC : 25% calibration channel sampling requires at least 64 sampling cycles

Errata type: Information

Description: The 25%*(VRH-VRL) calibration channel (ADC channel 44) will not convert to specification with an ADC sample time less than 64 cycles.

Workaround: For accurate calibration, the 25% calibration channel should be converted using the Long Sample Time (LST) setting for either 64 or 128 ADC sample cycles in the ADC Conversion Command Message (LST = 0b10 or 0b11).

e652: EQADC: 50% reference channels reads 20 mv low

Errata type: Information

Description: The equation given for the definition of the 50% reference channel (channel 42) of the Enhanced Queued Analog to Digital Converter (eQADC) is not correct. The 50% reference point will actually return approximately 20mV (after calibration) lower than the expected 50% of difference between the High Reference Voltage (VRH) and the Low Reference Voltage (VRL).

Workaround: Do not use the 50% point to calibrate the ADC. Only use the 25% and 75% points for calibration.

After calibration, software should expect that the 50% Reference will read 20 mV low.

e2075: FLASH: Large blocks limited to 1,000 Program/erase cycles

Errata type: Errata

Description: The electrical specification for Program/Erase cycling on large Flash blocks (all 128K blocks - Middle Address Space [MAS] blocks M0 and M1, plus High Address Space [HAS] blocks H0 to H3/H7/H11/H19 [depending on total flash size]) has been changed to 1,000 PE cycles minimum. The small blocks (16K, 48K, and 64K - Low Address Space [LAS] blocks L0-L5) are still specified as 100,000 PE cycles minimum.

The data retention specification all blocks is still 20 years for blocks cycled less than 1000 times and 5 years for blocks cycled 1001 to 100,000 cycles (1,000 for large blocks).

Workaround: Only use the small blocks for EEPROM emulation (LAS L0-L5). Do not use blocks MAS M0/M1 or HAS H0 to H3/H7/H11/H19 (depending on total flash size) for EEPROM emulation requiring greater than 1,000 Program/Erase cycles. Refer to the latest device electrical specifications (Data Sheet) dated July 2007 or later.

e605: FLASH: Minimum Programming Frequency is 25 MHz

Errata type: Information

Description: Programming and erase operations of the internal flash could fail if the clock to the flash (usually the system clock) is less than 25 MHz.

Workaround: Do not program or erase the flash when the system operating frequency is below 25Mhz.

e2083: FLASH: The ADR register may get loaded with a flash address even through no ECC error has occurred

Errata type: Information

Description: The Flash Address Register (FLASH_AR) may be loaded with a flash address when no Error Correction Code (ECC) has occurred. When an ECC does occur, the FLASH_AR is properly set.

Workaround: Check the Flash Module Control Register ECC Event Error (FLASH_MCR[EER]=1) to check for an ECC error before examining the ADR register. If an error has occurred then the ADR register data is valid. If an error has not occurred then the FLASH_AR data could change on any flash access.

e1111: FMPLL: LOLF can be set on MFD change

Errata type: Errata

Description: Normally, the Loss of Lock Flag (FMPLL_SYNCR[LOLF]) would not be set if the loss of lock occurred due to changing of the Multiplication Factor Divider bits or PREDIV bits (FMPLL_SYNCR[MFD] or [PREDIV]) or enabling of Frequency Modulation (FMPLL_SYNCR[Depth]>0b00). However, if LOLF has been set previously (due to an unexpected loss of lock condition) and then cleared (by writing a 1), a change of the MFD, PREDIV or DEPTH fields can cause the LOLF to be set again which can trigger an interrupt request if LOLIRQ bit is set.

In addition, changing the RATE bit will also set the LOLF regardless of previous conditions.

Workaround: The Loss of Lock Interrupt Request enable in the Synthesizer Control Register (FMPLL_SYNCR[LOLIRQ]) should be cleared before any change to the multiplication factor (MFD), PREDIV, modulation depth (DEPTH), or modulation rate (RATE) to avoid unintentional interrupt requests. After the PLL has locked (LOCK=1), LOLF should be cleared (by writing a 1) and LOLIRQ may be set again if required.

e2181: FMPLL: Non-zero pre-divider values can cause PLL lock failure

Errata type: Errata

Description: When configuring the MPC55xx Frequency Modulated Phase Lock Loop (FMPLL) in crystal or external reference clock mode, the system clock frequency (SYSCLK) is determined by the values programmed into the Pre-Divider (PREDIV), Multiplication Factor Divider (MFD), and Reduced Frequency Divider (RFD) fields in the FMPLL Synthesizer Control Register (FMPLL_SYNCR). If the pre-divider is set to divide by 2, 3, 4, or 5 (FMPLL_SYNCR[PREDIV] = 1, 2, 3, or 4), a condition may occur in which the FMPLL will fail to lock. Odd predividers may result in the PLL stuck in a lock routine where it can not escape. Even predividers may result in the PLL VCO frequency not being able to reach target frequencies below 110MHz. To clear this condition when it occurs, the part must be powered down.

Workaround: If a pre-divider of 2, 3, 4, or 5 must be used in order to achieve the desired system clock frequency, any write that causes a relock of the FMPLL (changing either the PREDIV or MFD) with a FMPLL_SYNCR[PREDIV] = 1, 2, 3, or 4 must occur with the current system frequency set to one-half of the crystal frequency or less through setting of the PLL RFD prior to writing the MFD or PREDIV.

NOTE: When programming the FMPLL, care must also be taken not to violate the maximum system clock frequency of the device, or the maximum and minimum frequency specifications of the FMPLL.

e507: FMPLL: Oscillator Gain Increased

Errata type: Information

Description: The gain of the oscillator was increased to handle a 40 MHz crystal on some devices. The 40 MHz crystal, however, is not supported on all devices.

Workaround: A resistor may need to be added between the XTAL pin and the crystal. Consult the crystal manufacturer for the recommended crystal configuration, however, the maximum value recommended should be kept to 1.0K ohms or lower for an 8 MHz crystal and a 470 ohms for a 40 MHz crystal.

e1232: FMPLL: Reset may not be negated if an external reset occurs during a software initiated PLL relock sequence

Errata type: Errata

Description: During a software initiated change in frequency of the Phase Lock Loop (PLL) to a frequency greater than 112 MHz, if reset is externally asserted, it is possible that reset will never be exited.

Workaround: Do not allow external resets to be applied during a software initiated PLL frequency change that requires a relock of the PLL. A possible way to prevent this from occurring is to use a GPIO signal to gate (hold off) the RESET input to the device from the external power supply. The following sequence would have to be used:

1) drive GPIO to hold RESET negated,

2) Start the PLL lock sequence,

3) wait for PLL lock sequence to complete,

4)Toggle the GPIO pin to allow the external reset to assert the RESET pin.

e6531: FMPLL: Selecting GPIO mode on RSTCFG/PLLCFG[0:1] may cause PLL failure after a reset

Errata type: Errata

Description: If the General Purpose Input/Output (GPIO) mode is selected in software for the PLLCFG[0:1] and/or the RSTCFG pins AND the default clock reference mode (crystal reference) is not used, any RESET assertion of the device (internal or external) may corrupt the operating mode of the PLL and the microcontroller may not exit reset (RSTOUT will not negate).

Workaround: Do not enable the GPIO mode in software for the PLLCFG[0:1]/RSTCFG pins when over-riding the default (crystal) PLL clock mode (RSTCFG=0 AND PLLCFG[0:1] != 0b10).

e3407: FlexCAN: CAN Transmitter Stall in case of no Remote Frame in response to Tx packet with RTR=1

Errata type: Errata

Description: FlexCAN does not transmit an expected message when the same node detects an incoming Remote Request message asking for any remote answer.

The issue happens when two specific conditions occur:

1) The Message Buffer (MB) configured for remote answer (with code "a") is the last MB. The last MB is specified by Maximum MB field in the Module Configuration Register (MCR[MAXMB]).

2) The incoming Remote Request message does not match its ID against the last MB ID.

While an incoming Remote Request message is being received, the FlexCAN also scans the transmit (Tx) MBs to select the one with the higher priority for the next bus arbitration. It is expected that by the Intermission field it ends up with a selected candidate (winner). The coincidence of conditions (1) and (2) above creates an internal corner case that cancels the Tx winner and therefore no message will be selected for transmission in the next frame. This gives the appearance that the FlexCAN transmitter is stalled or "stops transmitting".

The problem can be detectable only if the message traffic ceases and the CAN bus enters into Idle state after the described sequence of events.

There is NO ISSUE if any of the conditions below holds:

a) The incoming message matches the remote answer MB with code "a".

b) The MB configured as remote answer with code "a" is not the last one.

c) Any MB (despite of being Tx or Rx) is reconfigured (by writing its CS field) just after the Intermission field.

d) A new incoming message sent by any external node starts just after the Intermission field.

Workaround: Do not configure the last MB as a Remote Answer (with code "a").

e1557: FlexCAN: Corrupt ID may be sent in early-SOF condition

Errata type: Errata

Description: This erratum is not relevant in a typical CAN network, with oscillator tolerances inside the specified limits, because an early start of frame condition (early-SOF) should not occur.

An early-SOF may only be a problem if the oscillators in the network operate at opposite ends of the tolerance range (maximum 1.58%), which could lead to a cumulated phase error after 10 bit-times larger than phase segment 2.

A corrupt ID will be sent out if a transmit message buffer is identified for transmission during INTERMISSION, and an early-SOF condition is entered due to a dominant bit being sampled during bit 3 of INTERMISSION.

The message sent will be taken from the newly set up transmit buffer (Tx MB), with the exception of the 1st 8 ID bits, which are taken from the previously selected Tx MB.

The CRC is correctly calculated on the resulting bit stream so that receiving nodes will validate the message.

The early-SOF condition is detailed in the Bosch CAN Specification Version 2.0 Part B, Section 3.2.5 INTERFRAME SPACING - INTERMISSION.

Workaround: 1) Configure Tx MBs during FREEZE mode, or

2) Out of FREEZE mode, configure Tx MBs during bus idle:

- For networks with low traffic, determine Bus Idle status by reading the Idle bit of the Error and Status register (CANx_ESR[IDLE]).

- For networks with high traffic, configure Tx MBs after the 3rd bit of intermission, and before the third bit of the CRC field from the next transmission.

e2685: FlexCAN: Module Disable Mode functionality not described correctly

Errata type: Errata

Description: Module Disable Mode functionality is described as the FlexCAN block is directly responsible for shutting down the clocks for both CAN Protocol Interface (CPI) and Message Buffer Management (MBM) sub-modules. In fact, FlexCAN requests this action to an external logic.

Workaround: In FlexCAN documentation chapter:

Section "Modes of Operation", bullet "Module Disable Mode":

Where is written:

"This low power mode is entered when the MDIS bit in the MCR Register is asserted. When disabled, the module shuts down the clocks to the CAN Protocol Interface and Message Buffer Management sub-modules.."

The correct description is:

"This low power mode is entered when the MDIS bit in the MCR Register is asserted by the CPU. When disabled, the module requests to disable the clocks to the CAN Protocol Interface and Message Buffer Management sub-modules."

Section "Modes of Operation Details", Sub-section "Module Disable Mode":

Where is written:

"This low power mode is entered when the MDIS bit in the MCR Register is asserted. If the module is disabled during Freeze Mode, it shuts down the clocks to the CPI and MBM sub-modules, sets the LPM_ACK bit and negates the FRZ_ACK bit.."

The correct description is:

"This low power mode is entered when the MDIS bit in the MCR Register is asserted. If the module is disabled during Freeze Mode, it requests to disable the clocks to the CAN Protocol Interface (CPI) and Message Buffer Management (MBM) sub-modules, sets the LPM_ACK bit and negates the FRZ_ACK bit."

e1794: FlexCAN: New Feature - Individual Message Mask Registers

Errata type: Information

Description: When the FlexCAN Message Buffer Filter Enable control bit in the FlexCAN Module Configuration Register, CANx_MCR[MBFEN] (bit 15), is set, additional filtering is provided by the RXIMR0 to RXIMR63 Individual Mask Registers which replace RXGMASK, RX14MASK and RX15MASK.

Workaround: This feature may not exist on all parts, and for software compatibility with devices that do not include Individual message mask registers, do not use the additional message mask registers. They can be used if backwards software compatibility is not required.

e352: FlexCAN: New feature - Individual RX matching and Message Queuing

Errata type: Information

Description: The FlexCAN allows reception of the same message ID in multiple message buffers by setting the new Message Buffer Filter Enable control bit in the FlexCAN Module Configuration Register, CANx_MCR[MBFEN] (bit 15). By programming more than one Message Buffer with the same ID or using a mask, received messages will be queued into the Message Buffers.

Workaround: For backwards software compatibility with the MPC5554, MPC5553, and the initial versions of the MPC5534, do not use this new feature or insure that the feature exists prior to their use.

e1654: FlexCAN: New feature - Self reception disable

Errata type: Information

Description: The FlexCAN can now be configured to disallow reception of frames transmitted by itself by setting the Self Reception Disable bit in the FlexCAN Module Configuration Register (CANx_MCR[SRXDIS]=0b1, bit 14).

Workaround: For backwards software compatibility with the MPC5554, MPC5553, and the initial versions of the MPC5534, do not use these new features or insure that the features exist prior to their use.

e1895: FlexCAN: New feature - Transmit (TX)/Receive (RX) Warning Interrupts

Errata type: Information

Description: The Warning Interrupt bit has been added in the FlexCAN Module Configuration Register, CANx_MCR[WRNEN] (bit 10). In addition two bits have been added in the FlexCAN Control Register, Transmit Warning Interrupt Mask, CANx_CR[TWRNMSK] (bit 20)and the Receive Warning Mask, CANx_CTRL[RWRNMSK] (bit 21) allow applications to enable monitoring for Transmit and Receive error counters and generate an interrupt for either if the error count reaches 96 errors or more. Consequently, two status bits have been added in the FlexCAN

Error and Status register to signal interrupts for these additional interrupt causes, the Transmit Warning Interrupt bit (CANx_ESR[TWRNINT], bit 14) and the Receive Warning Interrupt bit (CANx_ESR[RWRNINT], bit 15). Both of these status bits are cleared by writing a 1 to the bit.

Workaround: For backwards software compatibility with the MPC5554, MPC5553, and the initial versions of the MPC5534, do not use this new feature or insure that the feature exists prior to their use.

e2424: FlexCAN: switching CAN protocol interface (CPI) to system clock has very small chance of causing the CPI to enter an indeterminate state

Errata type: Errata

Description: The reset value for the clock source of the CAN protocol interface (CPI) is the oscillator clock. If the CPI clock source is switched to the system clock while the FlexCAN is not in freeze mode, then the CPI has a very small chance of entering an indeterminate state.

Workaround: Switch the clock source while the FlexCAN is in a halted state by setting HALT bit in the FlexCAN Module Configuration Register (CANx_MCR[HALT]=1). If the write to the CAN Control Register to change the clock source (CANx_CR[CLK_SRC]=1) is done in the same oscillator clock period as the write to CANx_MCR[HALT], then chance of the CPI entering an indeterminate state is extremely small. If those writes are done on different oscillator clock periods, then the corruption is impossible. Even if the writes happen back-to-back, as long as the system clock to oscillator clock frequency ratio is less than three, then the writes will happen on different oscillator clock periods.

e1808: FlexRay: A boundary violation frame followed by a valid startup frame during the startup phase may cause an abort of the startup

Errata type: Errata

Description: The FlexRay module may abort the startup due to a wrong deviation measurement if:

(a) The FlexRay module is in STARTUP state and

(b) the FlexRay module receives a startup frame that violates the boundary at the beginning of the slot followed by a valid startup frame in the same slot.

The following flags and fields may be affected:

(a) The PROTSTATE field of the Protocol Status Register FR_PSR0 may indicate INTEGRATION_LISTEN instead of NORMAL_ACTIVE after the startup.

(b) The OFFSETCORR field in the Offset Correction Value Register (FR_OFCORVR) may show a wrong value for the related communication cycle.

(c) The RATECORR field in the Rate Correction Value Register (FR_RTCORVR) may show a wrong value for the related communication cycle pair.

(d) The Clock Correction Limit Reached Interrupt Flag CCL_IF of the Protocol Interrupt Flag Register 0 (FR_PIFR0) may be set by the FlexRay module, indicating an EXCEED_BOUNDS condition due to the erroneous deviation measurement.

Workaround: There is no workaround for this erratum.

e2302: FlexRay: Message Buffer can not be disabled and not locked after CHI command FREEZE

Errata type: Errata

Description: If a complete message was transmitted from a transmit message buffer or received into a message buffer and the controller host interface (CHI) command FREEZE is issued by the application before the end of the current slot, then this message buffer can not be disabled and locked until the module has entered the protocol state normal active.

Consequently, this message buffer can not be disabled and locked by the application in the protocol config state, which prevents the application from clearing the commit bit CMT and the module from clearing the status bits. The configuration bits in the Message Buffer Configuration, Control, Status Registers (MBCCSRn) and the message buffer configuration registers MBCCFRn, MBFIDRn, and MBIDXRn are not affected.

At most one message buffer per channel is affected.

Workaround: There are two types of workaround.

1) The application should not send the CHI command FREEZE and use the CHI command HALT instead.

2) Before sending the CHI command FREEZE the application should repeatedly try to disable all message buffers until all message buffers are disabled. This maximum duration of this task is three static or three dynamic slots.

e2344: FlexRay: Repeated wakeup after leading coldstart has too short WAKEUP_LISTEN phase

Errata type: Errata

Description: When the FlexRay module is

a) configured for single-channel, and

b) configured as leading coldstarter, and

c) the FlexRay module has performed a startup as leading coldstarter, and

d) the application triggers another wakeup via the wakeup command (without performing a power-on reset before, which means that the previous configuration and internal state is preserved)

then

a) the WAKUP_LISTEN phase is truncated to 4 microticks which translates to 4 * 25ns @10Mbps, and

b) the FlexRay module then immediately transitions into WAKEUP_SEND and starts transmission of the wakeup-pattern on the network.

Workaround: Prior to executing the WAKEUP command the application should perform a hypothetical leading coldstart using the following steps:

1) Configure the FlexRay module as leading coldstarter via the ALLOW_COLDSTART command.

2) Perform the RUN command.

3) Wait until the FlexRay module has reached the COLDSTART_LISTEN state (PSR0[STARTUPSTATE] == 0011).

4) Abort the startup via READY command.

5) Wait until the FlexRay module has reached the READY state (PSR0[PROTSTATE] == 011).

6) Perform WAKEUP command.

e1545: FlexRay: Slot Status of Double Transmit Message Buffers updated incorrectly

Errata type: Errata

Description: This erratum only affects FlexRay modules that have at least two double transmit message buffers configured by the application.

After the transmission of a non-null frame from a double transmit message buffer, the FlexRay module

1) updates the slot status of the transmit side of this double transmit message buffer,

2) does not update the slot status of the commit side of this double transmit message buffer, and

3) updates the slot status of the commit side of the double transmit message buffer with the highest message buffer ID.

Due to internal commit operations on double transmit message buffers it can happen, that the slot status of the commit side and transmit side are exchanged after the update. As a result, the slot status of the double transmit message buffer is incorrect for a certain amount of time.

Workaround: If the application has configured at most one double transmit message buffer, the slot status of this message buffer is always correct.

If the application has configured more than one double transmit message buffers, it should ignore the slot status of the double transmit message buffers and instead should use the dedicated slot status reporting registers provided by the FlexRay module to get the slot status information. These registers are

- Channel A Status Error Counter Register (FR_CASERCR)

- Channel B Status Error Counter Register (FR_CBSERCR)

- Protocol Status Register 2 (FR_PSR2)

- Protocol Status Register 3 (FR_PSR3)

- Slot Status Registers (FR_SSR0 up to FR_SSR7)

- Slot Status Counter Registers (FR_SSCR0 up to FR_SSCR3)

e1958: FlexRay: System memory access immediately timed out if SYMATOR[TIMEOUT] set to 0x1F

Errata type: Errata

Description: If the FlexRay module performs a read or write operation on the system memory and the TIMEOUT field in the System Memory Timeout Register (SYMATOR) is set to its maximum value of 0x1F, the FlexRay module will immediately set the System Bus Communication Failure Error Flag SBCF_EF in the CHI Error Flag Register (CHIERFR). In case of an

read operation, the FlexRay module will proceed its operation and assume a read value of 0.

Workaround: The application should not write the value of 0x1F to the SYMATOR[TIMEOUT] field.

e1657: FlexRay: System memory overwritten or invalid data transmitted after timeout of system memory read access

Errata type: Errata

Description: When the FlexRay module performs a read operation from the system memory and the system memory subsystem fails to deliver the requested data within the number of system clock cycles configured by the TIMEOUT field in the System Memory Timeout Register (SYMATOR), the FlexRay module will proceed its operation with a read value of 0.

If the value to be read was in the Frame Header of the Message Buffer Header Field, an incorrect frame header will be transmitted.

If the value to be read was the Data Offset Field of the Message Buffer Header Field the FlexRay module will

a) in case of a transmit slot, fetch the payload data from the start of the 64 KByte system memory window starting at the address defined by the System Memory Base Address Registers (SYMBADHR,SYMBADLR), and consequently transmit incorrect payload data, or

b) in case of a receive slot, write the payload data to the start of the 64 KByte system memory window starting at the address defined by the System Memory Base Address Registers (SYMBADHR,SYMBADLR), and consequently overwrite several Message Buffer Header Fields.

If the value to be read was in the Message Buffer Data Field an incorrect payload word will be transmitted.

Workaround: To prevent the occurrence of an system memory timeout, the application should configure the priorities for the system memory bus masters in the Crossbar Switch properly along with an appropriate setting of the SYMATOR[TIMEOUT] field.

To prevent the overwrite of Message Buffer Header Fields, the application should reserve 254 bytes of unused memory at the start of the FlexRay Memory Window.

To prevent the transmission of incorrect payload data, the application should continuously observe or assign an interrupt service routine to the System Bus Communication Failure Error Flag SBCF_EF in the CHI Error Flag Register (CHIERFR). When this flag is set or the interrupt is triggered, the application should stop the FlexRay module by the protocol command FREEZE.

e1429: Flexray : No reception in slot in dynamic segment

Errata type: Errata

Description: If a FlexRay transmit message buffer and a receive message buffer are both assigned to dynamic slots, and the transmit message buffer has a greater message buffer number than the receive message buffer, then the receive message buffer will never be found by the message buffer search and no reception will happen.

Workaround: All individual transmit message buffers should be located before all receive message buffers, that means, all transmit message buffers have message buffer numbers lower than any receive message buffer number.

e789: MPC5561: SIU_MIDR[PARTNUM] is 5561, [MASKNUM] is 0x01, DID[PIN]=0x161

Errata type: Information

Description: The part number field in the MCU Identification Register (SIU_MIDR[PARTNUM]) is 0x5561. The mask revision number (SIU_MIDR[MASKNUM]) is updated to 0x01. The Part Number Identification field in the Nexus Port Controller Device Identification Register/JTAGC Identification (NPC_DID[PIN]) is 0x161.

Workaround: Software should be aware that the SIU_MIDR[MASKNUM] field can change in the future. Tools should be aware of the JTAGC_ID/NPC_DID[PIN]. In addition, tools should be aware that the revision number in the JTAG and Nexus ID could change in the future (JTAGC_ID/NPC_DID[PRN]).

e547: MPC5561: VSTBY pin current is higher than expected

Errata type: Errata

Description: The VSTBY pin does not meet data sheet values for standby current at 25C or 85C.

Workaround: Expect that at least 50 uA of additional current will be required on the power supply for the VSTBY pin compared to specified values, Alternately if no standby functionality required, connect Vstby pin to ground.

e6726: NPC: MCKO clock may be gated one clock period early when MCKO frequency is programmed as SYS_CLK/8.and gating is enabled

Errata type: Errata

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.

e1957: NPC: MCKO_DIV can be set to 0x0 (1X MCKO)

Errata type: Information

Description: The Nexus Port Controller Port Configuration Register MCKO Divider bits (NPC_PCR[MCKO_DIV]) can be set to 0b000 to select a 1X clock rate as the Nexus Auxiliary output port frequency for the MCKO and MDO pins. Note: Depending on the system frequency, this may force the MCKO and MDO pins to switch at a frequency higher than can be supported by the pins. This maximum frequency is specified in the device electrical specification of the Nexus MCKO and MDO pins.

Workaround: Insure that the maximum operating frequency of the MDO and MCKO pins is not violated when setting the NPC_PCR[MCKO_DIV] values.

Note: tools may not support 1X mode. Check with your tool vendor.

e2096: NZ6C3: No sync message generated after 255 direct branch messages in history mode

Errata type: Information

Description: When using the branch history mode of direct branch program trace in the e200z6 core, a synchronization message is not transmitted after 255 program trace messages in a row as defined in the IEEE-ISTO 5001-2003 standard. This will occur if resource full messages are sent and not counted for triggering a sync message indicating that the branch history fields are full. The resource full message is generated when more than 31 direct branches occur without an indirect branch or exception.

Workaround: Debuggers should account for the possibility that more than 255 messages could be received without a program trace synchronization message by keeping track of the last known program trace address prior to branch history resource full messages.

e2103: NZ6C3: Data Trace of stmw instructions may cause overruns

Errata type: Information

Description: If Nexus data trace is enabled on a section of memory that is loaded or stored with a store multiple word (stmw), or load multiple word (lmw for data read traces), an overrun condition could occur, even if the stall on overrun feature is enabled (NZ6C3_DC1[OVC]=0b011). Stalls can only occur on instruction boundaries. The stmw/lmw instructions can generate up to 16 Nexus trace messages with a single instruction. If there are not 16 queue locations available, an overflow will occur. Stall mode does not stall the core until there are only four locations available in the e200 Nexus message queue. Therefore if a stmw/lmw generates more than four messages or if additional Nexus messages are generated, the queue will overflow. The stmw/lmw instructions load or store two 32-bit registers at a time (64-bit stores/loads) if an even number of registers are selected.

Workaround: If stall mode is enabled (NZ6C3_DC1[OVC]=0b011), limiting store multiple word instruction in a data trace region to store/load 8 registers or less, will improve the chances that an overrun will not occur, but this is dependent on other messages that could be generated simultaneously with the data trace messages. If stall mode is disabled, or stmw instructions with more than 8 registers are stored, accept overruns in the data and program trace flow.

e1618: NZ6C3: No indication of an exception causing a Nexus Program Trace (PT) message as opposed to a retired branch instruction causing a PT message.

Errata type: Errata

Description: The e200z6 core Nexus (NZ6C3) transmits a Program Trace Indirect Branch message without indicating if the message was sent due to a taken branch or due to an exception. The instruction count for an exception is 1 less than a normal indirect branch. The result is that program trace reconstruction can be off by one instruction.

Workaround: Trace reconstruction tools should be aware that the I-CNT is different for Exceptions than for Indirect Branches. The tool may need to know (from the user or by parsing registers) the exception handler addresses from the Interrupt vector prefix register (IVPR) and the Interrupt vector offset registers (IVORxx). Users also should not jump directly to interrupt handler addresses. Tools can then differentiate between exceptions and indirect branches.

e424: Pad Ring: FlexRay pins changed to MH type

Errata type: Errata

Description: To meet the specified speed on the FlexRay interface for channel B pins FR_TX_B and FR_TX_B_EN, the pad type was changed to a MH type pad. Other MPC55XX parts that implement FlexRay may have other pad types or have a different pad used for these functions.

Workaround: Refer to the specific device reference manual and errata sheet for details on the specific implementation for any other MPC5500 device. Exact drop-in replacement may not be available for this feature.

e488: Pad Ring: Possible poor system clock just after POR negation.

Errata type: Information

Description: The pins RSTCFG_B and PLLCFG[0:1] select one of three PLL modes or allows a clock to be injected, bypassing the PLL. When Power On Reset (POR) negates, if the transitions on these pins selects the bypass mode, a poor clock on EXTAL can provide a poor clock to MCU logic no longer reset by POR. The state of that logic can be corrupted.

Workaround: If the default PLL and Boot configuration (external crystal reference and boot from internal flash) will be used, then negate the RSTCFG pin (=1). For any other configuration, depending on the final mode required, the pins must have the following values on the pins when the internal POR negates.

Final Mode RSTCFG_B PLLCFG[0] PLLCFG[1]

default 1 - -

external reference 0 1 1

external crystal - 1 -

dual controller - 1 -

After POR negates, the RSTCFG_B and PLLCFG[0:1] can be changed to their final value, but should avoid switching through the 0,0,0 state on these pins. See application note AN2613 "MPC5554 Minimum Board Configuration" for one example off the external configuration circuit.

e1969: Pad Ring: RSTOUT is 3-stated during the power-on sequence.

Errata type: Information

Description: RSTOUT_B is 3-stated during power on reset.

Workaround: Connect an external pull device to RSTOUT_B during power on reset. This should be pull-down unless an external reset configuration circuit is being used, in which case it should be pull-up. Refer to AN2613 'MPC5554 Minimum Board Configurations' for further information.

e375: Pad Ring: SCI_A available on CAN_A pins (GPIO[83:84])

Errata type: Information

Description: The Serial Interface (SCI_A) transmit (TX) and receive (RX) functions have been added to the pin multiplexer on the GPIO83/CAN_A_TX and GPIO84/CAN_A_RX pins and can be selected in the Pad Configuration Registers.

Workaround: Do not use the SCI_A TX and RX functions on the CAN_A pins if backwards hardware compatibility is required with previously defined devices (MPC5534, MPC5553, and MPC5554). The SCI_A TX and RX can be selected by setting the Pin Assignment bits of the Pad Configuration Register to 0b10 (PCR83 and PCR84).

e3377: Pad Ring:Nexus pins may drive an unknown value immediately after power up but before the 1st clock edge

Errata type: Errata

Description: The Nexus Output pins (Message Data outputs 0:15 [MDO] and Message Start/End outputs 0:1 [MSEO]) may drive an unknown value (high or low) immediately after power up but before the 1st clock edge propagates through the device (instead of being weakly pulled low). This may cause high currents if the pins are tied directly to a supply/ground or any low resistance driver (when used as a general purpose input [GPI] in the application).

Workaround: 1. Do not tie the Nexus output pins directly to ground or a power supply.

2. If these pins are used as GPI, limit the current to the ability of the regulator supply to guarantee correct start up of the power supply. Each pin may draw upwards of 150mA.

If not used, the pins may be left unconnected.

e1779: SIU: CRSE bit added to the SIU Configuration Register

Errata type: Information

Description: A new bit was added to the System Integration Unit to disable driving both the normal external bus and the calibration bus interface.

The Calibration Reflection Suppression Enable (SIU_CCR[CRSE]) bit enables the suppression of reflections from the External Bus Interface's calibration bus onto the non-calibration bus. The EBI drives some outputs to both the calibration and non-calibration busses. When CRSE is asserted (0b1), the values driven onto the calibration bus pins will not be reflected onto the non-calibration bus pins. When CRSE is negated (0b0), the values driven onto the calibration bus pins will be reflected onto the non-calibration bus pins. CRSE only enables reflection suppression for non-calibration bus pins that do not have a negated state to which the pins return at the end of the access. CRSE does not enable reflection suppression for the non-calibration bus pins that have a negated state to which the pins return at the end of an access. Those reflections always are suppressed. Furthermore, the suppression of reflections from the non-calibration bus onto the calibration bus is not enabled by CRSE. Those reflections also always are suppressed.

Workaround: Set the CRSE bit in the SIU_CCR to prevent signals on the calibration bus from being reflected onto the normal external bus interface.

e637: SIU: ECCR[ENGDIV] field extended to eight bits

Errata type: Information

Description: Two additional bits (bits 15:16) have been added to the divider (ENGDIV) for the Engineering Clock (ENGCLK) in the External Clock Control register (SIU_ECCR).

Note: not all devices that support these additional bits actually implement the Engineering Clock pin.

Workaround: For backwards compatibility to other devices either do not use the extra two bits, otherwise software will only be compatible on devices that include these two extra bits for the divider.

e1721: SIU: ENGCLK alternate source select bit added to SIU_ECCR

Errata type: Information

Description: An alternate source selection bit (Engineering Clock Source Select) has been added to the System Integration Unit External Clock Control Register (SIU_ECCR[ECSS], bit 24) to select the clock source that drives the Engineering Clock (ENGCLK) pin through the engineering clock divider from either the system clock or the EXTAL frequency.

Note: not all devices that support this additional bit actually implement the Engineering Clock pin.

Workaround: Set the SIU_ECCR[ENGSSEL] to 0b1 to enable the select the crystal (or external clock on EXTAL) as the source of the ENGCLK divider. Setting the bit to 0b0 will select the system clock to the engineering clock divider (backwards compatible).

e1419: SIU: Reverting ENGCLK source to the system clock has a very small chance of causing the ENGCLK generator to enter an indeterminate state

Errata type: Errata

Description: The reset value for the Engineering Clock (ENGCLK) source is the system clock. If the clock source is switched to the external clock (buffered crystal frequency or external clock input) by setting Engineering Clock Source Select bit in the External Clock Control Register (SIU_ECCR[ECSS]=1) and then reverted to the system clock, then the ENGCLK generator has a very small chance of entering an indeterminate state.

Workaround: Do not change the ENGCLK source back to the System clock after changing it to the external clock.

e1031: e200z6: Cache Invalidate/Clear Aborts due to masked core Interrupt

Errata type: Errata

Description: The e200z6 core will abort a "Cache Invalidate" (CINV) or "Cache Lock Bits Flash Clear" (CLFC) operation (in the Level 1 Cache Control and Status Register 0 (L1CSR0) if an external interrupt occurs during the operation, even if external interrupts to the core are disabled in the Machine Status Register (MSR[EE]=0). The effect of this issue is that disabling external interrupts alone will not guarantee completion of a CINV or CLFC operation. Software must check for the cache abort condition.

Workaround: After clearing the EE bit in MSR, perform the following steps:

(1) Save the interrupt controller's current priority, INTC_CPR[PRI].

(2) Raise the interrupt controller's current priority to maximum (INTC_CPR[PRI]=0xF).

(3) Synchronize by executing msync and isync instructions.

(4) Ensure critical interrupts are also disabled (MSR[CE]=0).

(5) Execute CINV or CLFC cache operation.

After the cache operation completes, check to see if the abort bit has been set, indicating an external interrupt has caused the operation to terminate before completion. If set, clear the abort bit and retry the operation.

To restore normal operation after the cache invalidation or clear, perform the following steps:

(1) Restore the saved priority setting to INTC_CPR[PRI].

(2) Restore MSR[CE] if it was altered.

(3) Restore MSR[EE].

e2044: e200z6: Core renamed from e500z6

Errata type: Information

Description: The name of the main processing core has been changed from the e500z6 to the e200z6.

Workaround: Expect the new name for the e200z6 core in documentation.

e1729: e200z6: MMU has 32 Table Entries

Errata type: Information

Description: Initial documentation for the MPC5554 stated that there would be only 24 table entries in the e200z6 core Memory Managment Unit (MMU). Actually, 32 entries were implemented and will remain in the future e200z6 devices.

Workaround: All 32 of the MMU table entries can be used.

e779: e200z6: New SPE Instructions Added

Errata type: Information

Description: e200z6 cores that support VLE also have additional instructions available in the traditional PowerPC instruction set.

The following SPE instructions have been added:

evfsmadd - Vector Floating-Point Single-Precision Multiply-Add

evfsnmadd - Vector Floating-Point Single-Precision Negative Multiply-Add

evfsmsub - Vector Floating-Point Single-Precision Multiply-Subtract

evfsnmsub - Vector Floating-Point Single-Precision Negative Multiply-Subtract

efsmadd - Floating-Point Single-Precision Multiply-Add efsnmadd - Floating-Point Single-Precision Negative Multiply-Add

efsmsub - Floating-Point Single-Precision Multiply-Subtract

efsnmsub - Floating-Point Single-Precision Negative Multiply-Subtract

In addition, the restriction of doubleword alignment (64-bit) for SPE load/store instructions has been removed.

Workaround: For backwards compatibility to devices that do not support VLE, do not use these new SPE instructions. Compilers may provide a switch to allow or to not allow use of these additional instructions in assembly code.

e1581: e200z6: VLE added to core

Errata type: Information

Description: An optional Variable Length Encoded (VLE) instruction set has been added to the e200z6 core. VLE is an alternate instruction set that includes both 16-bit and 32-bit instruction encodings. Additional bits have been added to several registers to support this mode of operation. See the "e200z6 with VLE" addendum to the "e200z6 Reference Manual" for complete details on the instruction set and register bits. The addendum is available at: http://www.freescale.com/files/32bit/doc/ref_manual/e200z6RMAD1.pdf

Workaround: Insure that the device in use implements VLE before executing code from a page defined as VLE in a Memory Management Unit's table entry.

e860: e200z6: Way Access Mode bit added to the Cache

Errata type: Information

Description: A new feature has been added to the e200 Cache to allow the cache ways to be completely disabled if not enabled specifically for either data or instruction use. Setting the Way Access Mode (WAM, bit 10) in the Level 1 Cache Control and Status register 0 (L1CSR0) completely disables look ups in ways that are not specifically disabled by the Additional Way Instruction Disable (AWID), Way Instruction Disable (WID), Additional Way Data Disable (AWDD), and Way Data disable (WDD) fields(L1CSR0[WAM]=0b1). Note the AWID and AWDD bits are not available on devices with 8K of cache.

Workaround: For future compatibility (with devices that support the WAM bit), set the WAM bit when writing to the L1CSR0. On devices that do not support WAM, writing this bit has no affect and will read back as cleared (0b0). This allows software to be written to take advantage of WAM capability on devices that will support it. Setting WAM will reduce the operating power consumption of the cache on devices that support WAM.

e3421: e200z: Cache returns value, even when disabled

Errata type: Errata

Description: The cache line fill buffer of the e200z core will still provide a data cache hit, even if the Way Data Disable (WDD), Additional Ways Data Disable (AWDD), and the Way Access Mode (WAM, if available) bits are set.

Workaround: To avoid cache hit to the Line Fill Buffer after the WDD, AWDD, and WAM bits are disabled, a Cache Miss must be forced to clear the fill buffer. This can be done by branching to an instruction address that has not been fetched before.

e1136: eDMA: BWC setting may be ignored between 1st and 2nd transfers and after the last write of each minor loop.

Errata type: Errata

Description: The eDMA Transfer Control Descriptor Bandwidth Control field setting may be ignored between 1st and 2nd transfers and after the last write of each minor loop. This will occur if the source and destination sizes are equal. This behavior is a side effect of measures designed to reduce start-up latency. Reference Manuals may fail to mention this behavior.

Workaround: There are 2 possible workarounds:

1) Adjust the Transfer Control Descriptor (TCD) to make the source size not equal to the destination sizes (i.e. ssize = 16 bit, dsize = 32 bit). This delays the write which allows BWC[0:1] arriving from the TCD to be considered in the execution pipeline during start-up.

2) Adjust the TCD so the channel executes a single read/write sequence and then retires. In addition, the channel can be configured to execute a minor loop link to itself which will restart the channel after arbitration and channel start-up latency. The total number of bytes transferred can be controlled by the major loop count.

e6966: eDMA: Possible misbehavior of a preempted channel when using continuous link mode

Errata type: Errata

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.

e617: eDMA: Some new features do not function as specified

Errata type: Errata

Description: The Cancel Transfer with Error bit (EDMA_CR[ECX]) does not function correctly. Reading of the Cancel Transfer bit (EDMA_CR[CX]) does not give status of when cancel has been honored. The NOP bits in Set Enable Request Register (EDMA_SERQR), Clear Enable Request Register (EDMA_CERQR), Set Enable Error Interrupt Register (EDMA_SEEIR), Clear Enable Error Interrupt Register (EDMA_CEEIR), Clear Interrupt Request Register (EDMA_CIRQR), Clear Error Register (EDMA_CER), Set Start Bit Register (EDMA_SSBR), and Clear Done Status Bit Register (EDMA_CDSBR) do not function correctly.

Workaround: EDMA_CR[ECX] should not be used and doing so can cause unexpected results. Software should poll the TCD.ACTIVE bit to determine when the cancel of the executing channel has been honored. Software should not use the NOP bits in SERQR, CERQR, SEEIR, CEEIR, CIRQR, CER, SSBR, and CDSBR and doing so can cause unexpected results.

e1834: eMIOS: OPMWC unable to produce close to 100% duty cycle signal

Errata type: Errata

Description: The Center Aligned Output Pulse Width Modulation with Dead-time Mode (OPWMC) of the eMIOS module does not function correctly if the trailing edge dead time is programmed to a value outside of the current cycle time. The OPWMC mode requires that matches occur in the specific order: A, A, and then B, where the first A must match on the up count of the modulus counter, the second A match occurs on the down count of the modulus counter, and the B match occurs on the internal counter. If the programmed B match value is greater than the time required for the modulus counter to count down from the second A match and then up to the first A match of the next cycle, the first A match of the next cycle will be missed and the mode will not function correctly from that point on.

Workaround: Configure the selected modulus counter time base and the internal counter of the channel in OPWMC mode to count at the same rate. Program the value of the B match (dead time) to a value less than 2 times the programmed A match value.

e1558: eQADC: Leakage if any other supply/pin is greater than 0.6V above VDDA

Errata type: Information

Description: Additional leakage could occur if any pin or power supply on the device (except VSTBY) is greater than 0.6 volts above the VDDA supply, including during power off conditions. The leakage will be to the VDDA supply. This leakage path is active only after the VDDA supply is powered. The leakage path remains active until internal POR is asserted.

Workaround: Expect higher leakages until the voltages are reduced to within 0.6 volts of the voltage on VDDA.

e1773: eQADC: conversions of muxed digital/analog channels close to the rail

Errata type: Information

Description: If the VDDEH9 and the VDDA power supplies are at different voltage levels, the input clamp diodes on the multiplexed digital and analog signals (AN12, AN13, AN14, and AN15) will clamp to the lower of the two supplies.

If VDDEH9 is lower than the VDDA, conversions on these channels will not obtain full scale readings if voltage is close the the VDDA voltage.

Workaround: When multiplexed digital/analog signals are used as analog inputs, connect VDDEH9 to VDDA and do not use any of the digital functions multiplexed on these pins.

e1013: eSCI : Automatic reset of the LIN state machine cause incorrect transmission

Errata type: Errata
Description: When the LIN FSM of the eSCI module performs an automatic reset due to a bus error condition, it is possible that the application is no longer synchronized to the LIN FSM. As a result, the eSCI may consider the payload data provided by the application via the LIN Transmit Register (eSCI_LTR) as a LIN frame header data and will generate an unexpected LIN frame.

Workaround: The application should disable the automatic LIN FSM reset functionality by setting LDBG bit in the LIN Control Register (eSCI_LCR) to 1.

e1381: eSCI: LIN Wakeup flag set after aborted LIN frame transmission

Errata type: Errata
Description: If the eSCI module is transmitting a LIN frame and the application sets and clears the LIN Finite State Machine Resync bit in the LIN Control Register 1 (eSCI_LCR1[LRES]) to abort the transmission, the LIN Wakeup Receive Flag in the LIN Status Register may be set (LWAKE=1).

Workaround: If the application has triggered LIN Protocol Engine Reset via the eSCI_LCR1[LRES], it should wait for the duration of a frame and clear the eSCI_IFSR2[LWAKE] flag before waiting for a wakeup.

e1221: eSCI: LIN bit error indicated at start of transmission after LIN reset

Errata type: Errata
Description: If the eSCI module is in LIN mode and is transmitting a LIN frame, and the application sets and subsequently clears the LIN reset bit (LRES) in the LIN Control register 1 (ESCI_LCR1), the next LIN frame transmission might incorrectly signal the occurrence of bit errors (ESCI_IFSR1[BERR]) and frame error (ESCI_IFSR1[FE]), and the transmitted frame might be incorrect.

Workaround: There is no generic work around. The implementation of a suitable workaround is highly dependent on the application and a workaround may not be possible for all applications.

e1017: eSCI: LIN fast bit error detection causes incorrect LIN reception

Errata type: Errata
Description: When using the eSCI module in LIN mode, if the fast bit error detection is enabled in the eSCI Control Register 2 (eSCIx_CR2[FBR]) and a bit distortion occurs on the RX line, the eSCI module may process the bit error signal incorrectly. This may cause unexpected transmission and reception behavior of the LIN state machine.

Workaround: The application should disable the fast bit error detection by setting the FBR bit to 0.

e2190: eSCI: LIN slave timeout flag STO not asserted if CRC is received too late

Errata type: Errata

Description: The eSCI module will not assert the Slave-Timeout Flag (STO) in the eSCI LIN Status Register 1 (LINSTAT1) if the checksum field of on RX frame is received later than the time given in the Timeout Value (TO) field of the LIN RX control header.

If the checksum field of on RX frame is not received at all, The STO flag will not be set and the LIN FSM will wait forever.

Workaround: The application should use a timer external to the eSCI module, that is started when a LIN RX frame transmission is started with an expiration time programmed to the expected duration of the reception. The timer is stopped, when the eSCI module signals the end of the LIN frame reception. If this timer expires, the eSCI has ran into the slave timeout condition. In this case, the application should clear the SCICR2[TE] and SCICR2[RE] bits to disable the transmitter and receiver, and should set the LINCTRL1[LRES] bit the reset the eSCI internal LIN slave and master tasks. To resume LIN frame data transmission the application should enable the transmitter and receiver and clear the LINCTRL1[LRES] bit to enable the LIN slave and master tasks.

e1614: eSCI: Low pulse on LIN RX line may prevent assertion of transmit data ready flag ESCI_SR[TXRDY]

Errata type: Errata

Description: If the eSCI module is in LIN mode and receives a low pulse on the RX line while transmitting a frame header or a stop bit, the eSCI internal LIN master and slave tasks may be stopped, and the transmit data ready flag ESCI_SR[TXRDY] will never be asserted again.

Each of the following scenarios of the RX low pulse may provoke this erroneous behavior.

a) The low pulse begins less than 12 bits before the start of the break character, and ends at least 21 bits and at most 30 bits after the start of the break character.

b) The low pulse begins at least 13 bits and at most 14 bits after the start of the break character, and ends at least 22 bits after the start of the break character.

c) The low pulse begins during the stop bit and has duration of at least 2 bits.

Workaround: The application should use a timer external to the eSCI module, that is started when a LIN frame transmission is started with an expiration time programmed to the expected duration of the transmission. The timer is stopped, when the eSCI module signals the end of the LIN frame transmission.

If this timer expires, the eSCI tasks have been stopped internally. In this case, the application should clear the Transmit Enable (SCICR2[TE]) and the Receive Enable (SCICR2[RE] bits to disable the transmitter and receiver, and should set the LIN FSM resync (LINCTRL1[LRES]) bit the reset the eSCI internal LIN slave and master tasks.

To resume LIN frame data transmission the application should enable the transmitter and receiver and clear the LINCTRL1[LRES] bit to enable the LIN slave and master tasks.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

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

Freescale, the Freescale logo, and Qorivva are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.