Document identifier: IMX8XLB0_0P79F
Rev. 1, 11/2022

IMX8XLB0_0P79F

Mask Set Errata

Mask Set Errata for Mask 0P79F

Revision History

This report applies to mask 0P79F for these products:

- MIMX8DL2AVNFZAB
- MIMX8SL1AVNFZAB
- MIMX8DL1AVNFZAB
- MIMX8SL2AVNFZAB
- MIMX8DL3AVNFZAB
- MIMX8SL1CVNFZAB
- MIMX8DL1CVNFZAB
- MIMX8SL3AVNFZAB

Table 1. Revision History

Revision	Date	Significant Changes
1	11/2022	The following errata were added. • ERR051393
C	8/2022	The following errata were added. • ERR051198
B 4/2022	4/2022	No changes to errata with this revision
A 9/2021	2/2022	Initial Revision

Errata and Information Summary

Table 2. Errata and Information Summary

Erratum ID	Erratum Title
ERR050827	ADC0 generates phantom ADMA subsystem interrupts to SCU
ERR011077	ADC: Inside range compare function does not work if CVH=0.
ERR050949	ADC: Differential ADC mode is not supported
ERR050819	ADC: Higher Priority Trigger Interrupt is not supported.
ERR050821	ADC: Loop With automatic channel Increment (LWI) is not supported.
ERR050822	ADC: No support for pending trigger when same trigger is actively converting.
ERR050820	ADC: Repeat channel acquisition until true is not supported
ERR011184	ADC: Setting CFG[PUDLY] = 0 may result in an incorrect trigger delay time
ERR011189	ADC: When CFG[TPRICTRL] = 1, the first conversion of the higher priority trigger source will be performed twice

Table continues on the next page...

Table 2. Errata and Information Summary (continued)

Erratum ID	Erratum Title
ERR051393	Arm/Cortex-A core memory corruption
ERR006941	Core: Asynchronous sampling of SWDIOTMS might cause incorrect operation of SerialWire/ JTAG Debug Port
ERR050832	Core: ATS12NSOPR instruction might incorrectly translate when the HCR.TGE bit is set
ERR050877	Core: Fused MAC instructions give incorrect results for rare data combinations
ERR006939	Core: Interrupted loads to SP can cause erroneous behavior
ERR009004	Core: ITM can deadlock when global timestamping is enabled
ERR050831	Core: Mismatch between EDPRSR.SR and EDPRSR.R
ERR050830	Core: PMU counter might be inaccurate when monitoring BUS_ACCESS and BUS_ACCESS_ST
ERR050878	Core: Processor reset asserted asynchronously could corrupt FPB comparator registers and remap to wrong address
ERR050833	Core: Some AT instructions executed from EL3 might incorrectly report a domain fault
ERR050829	Core: Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation
ERR009005	Core: Store immediate overlapping exception return operation might vector to incorrect interrupt
ERR006940	Core: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used
ERR050451	DDR: MR register cannot be read if DDR DQ[7:0] are not connected to byte lane 0
ERR050793	Debug software must check the power status of each individual core before it interacts with it.
ERR050703	DEBUG: Trace function cannot be used on A35 cores
ERR051063	DEBUG: Tracing over low-power modes may produce incorrect results
ERR050272	DLL lock status bit not accurate due to timing issue
ERR010947	DRAM: DQS/DQSN glitch suppression resistors must be enabled during read-leveling
ERR010944	DRAM: In LPDDR4 mode, tMPCWR timing violation in incremental DQS2DQ Training
ERR010946	DRAM: In LPDDR4 mode: Auto refresh must be disabled during DQS2DQ training
ERR050341	DRAM: LPDDR4 VREF training may result in a non-optimal value
ERR010945	DRAM: PUB does not program LPDDR4 DRAM DDRPHY_MR22 prior to running DRAM ZQ calibration
ERR010948	DRAM: Timing Violation from Read/Write to MRW in LPDDR4 mode
ERR050533	DRC performance counters are not counting during DRC clock gating.
ERR050314	eDMA: High priority channel cannot preempt a low priority channel
ERR051078	ENET_AVB: ENETx_RGMII_TXC clock output may glitch when switching modes.
ERR051029	ENET_QOS: Gate Control List Switching is Incorrect for Intermediate Cycles When CTR is Less Than GCL Execution Time

Table continues on the next page...

Table 2. Errata and Information Summary (continued)

Erratum ID	Erratum Title
ERR011543	FlexCAN: Nominal Phase SJW incorrectly applied at CRC Delimiter
ERR050537	FlexSPI: Read timing sequence mismatches with several existing SPI NOR devices in dual, quad, and octal modes
ERR050948	FSM does not transition to reset state from IDLE when reg_pwren isn't set
ERR003777	GPT: Possibility of additional pulse on src_clk when switching between clock sources
ERR050513	GPV access end in bus fault
ERR051032	KS1 overconsumption in some subsystems
ERR010858	LPCG: IP clock gating register synchronization logic is sensitive to root clock gating
ERR010752	LPI2C: Slave Busy Flag may incorrectly read as zero when 10-bit address mode enabled in slave mode.
ERR011211	LPI2C: Slave Transmit Data Flag may incorrectly read as one when LPI2C_SCFGR1[TXCFG] is zero
ERR051041	LPIT: CVAL cannot be read correctly during timer running
ERR050606	LPSPI: TCR value does not get resampled when polling the register
ERR050607	LPSPI: TCR[FRAMSZ] can be ignored when TCR[TXMSK]=1b1
ERR050498	LPUART: Setting and immediately clearing SBK bit can result in transmission of two break characters
ERR010930	PCIE: EOM single point sample error/valid result is not correct
ERR011370	PCIE: EP, PM_PME: L1 Exit Does Not Occur when PME Service Timeout Mechanism Expires
ERR011194	PCIE: Plesiochronous loopback is not functional in PCIe Gen3
ERR051198	PWM: PWM output may not function correctly if the FIFO is empty when a new SAR value is programmed
ERR011150	SAI: Internally generated receive or transmit BCLK cannot be re-enabled if it is first disabled when RCR2[DIV] or TCR2[DIV] > 0
ERR050144	SAI: Setting FCONT=1 when TMR>0 may not function correctly
ERR050812	System Timer and TimeStamp Counter can stop under some slow clock conditions.
ERR010629	USDHC: EMMC HS200/SD SDR104 tuning process may fail based on delay cell number
ERR050873	USDHC: External Pull-up needed on pin ENET0_RGMII_TX_CTL to boot from USDHC1

Known Errata

ERR050451: DDR: MR register cannot be read if DDR DQ[7:0] are not connected to byte lane 0

Description

The DRAM controller performs MR reads and expects the LPDDR4 MR data to be on byte lane 0. There is no support for byte lane swapping. Hence when byte lanes are swapped on the PCB, the MR register on LPDDR4 memories cannot be read.

DDR3L memories use a different MR read method and are no affected.

Workaround

Connect byte lane 0 (DDR_DQ0 - DDR_DQ7) to DQ0 - DQ7 of the DRAM and byte lane 1 (DDR_DQ8 - DDR_DQ15) to DQ8-DQ15 of the DRAM.

ERR050820: ADC: Repeat channel acquisition until true is not supported

Description

Repeat channel acquisition until true (compare until true) can store invalid data in the FIFO when the command is not the last one in the sequence AND a data value of zero evaluates as TRUE.

Example:

ADC command buffer is programmed with a sequence CMD1, CMD2, CMD3, END.

- CMD3 is the last command in the sequence and is not affected by this issue.

- Assume CMD2 is configured for less than and CVL > 0; a data value of '0' will evaluate as TRUE and will trigger the issue.

- Assume CMD3 is configured for "outside"; when a data value of 0 is outside the CVL-CVH range, the data value of '0'will trigger the issue

.

Workaround

Do not use the repeat channel acquisition (CMPEN==11) on commands in a sequence unless it is the last command or a data value of '0'evaluates as false

ERR051063: DEBUG: Tracing over low-power modes may produce incorrect results

Description

When tracing software execution, the trace logic and the trace control logic must remain active. Entering into low-power modes can disable trace logic before the trace actions are completed which may result in corrupt data in the trace buffer. Low power modes may also power-down the trace buffer which will cause corruption or loss of trace data

Workaround

Do not trace over low-power modes that cause trace components to become inactive. Disable entering low-power modes when trace is enabled.

ERR050877: Core: Fused MAC instructions give incorrect results for rare data combinations

Description

Arm errata 839676

Affects: Cortex-M4F

Fault Type: Programmer Category B Rare

Fault Status: Present in: r0p0, r0p1. Open.

The Cortex-M4 processor includes optional floating-point logic which supports the fused MAC instructions (VFNMA, VFNMS, VFMA, VFMS). This erratum causes fused MAC operations on certain combinations of operands to result in either or both of the following:

• A result being generated which is one Unit of Least Precision (ULP) greater than the correct result.

• Inexact or underflow flags written to the Floating-point Status Control Register, FPSCR, incorrectly.

Configurations Affected:

Cortex-M4F configured with floating-point.

Conditions:

The conditions for this erratum are all of the following:

1. Cortex-M4 is configured with floating-point and it is enabled in the Coprocessor Access Control Register, CPACR.

2. Flush-to-Zero (FZ) mode is not enabled, that is, FPSCR.FZ is clear.

3. A fused MAC instruction (VFNMA, VFNMS, VFMA, or VFMS) is executed with all of the following properties:

• The addition part of the operation is Unlike Signed Addition (USA). This implies that the combination of the instruction used and the signs of the operands means that a subtraction is being performed.

• The result of the instruction, before rounding, is subnormal. This implies that the result is smaller in magnitude than 2-126.

• The significance of the product and the addend operand are the same or differ by one.

Implications:

If this erratum occurs, then the processor either produces an incorrect result to a computation or fails to run a floating-point exception routine when it should.

Note:

It is expected that most algorithms only use normalised numbers because:

• Subnormal results have low precision.

• It is easier to avoid underflow.

This erratum does not affect algorithms which only use normalized numbers.

Workaround

Enable FZ mode in the FPSCR.

ERR050832: Core: ATS12NSOPR instruction might incorrectly translate when the HCR.TGE bit is set

Description

Arm errata 801757

Affects: Cortex-A35

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2, r1p0. Open.

An ATS12NSOPR address translation instruction executed from EL3 might report an incorrect result in the PAR when both the HCR.TGE bit is set and the SCTLR.M is set.

Configurations Affected:

This erratum affects all configurations of the processor.

Conditions:

1. The core is executing at Exception level 3 in AArch32.

2. SCR.NS = 0

3. HCR.TGE = 1

4. SCTLR(ns).M = 1

5. The core executes an ATS12NSOPR instruction

Implications:

The PAR register will incorrectly report the translation as if the stage 1 MMU was enabled.

Note that the combination of both HCR.TGE and SCTLR.M bits being set was UNPREDICTABLE in earlier versions of the architecture, and was only given a defined behavior to reduce the UNPREDICTABLE space. It is not expected to be a useful combination for software.

Workaround

Secure software can clear the SCTLR(ns).M bit before executing the address translation instruction, if the HCR.TGE bit is set. It should restore the previous SCTLR(ns).M value before returning to a lower Exception level.

ERR050537: FlexSPI: Read timing sequence mismatches with several existing SPI NOR devices in dual, quad, and octal modes

Description

The FlexSPI controller expects every read command has at least one latency cycle between address phase and data phase to account for turnaround time on the IO bus. In multiple IO modes such as dual, quad, and octal modes, the FlexSPI controller inserts one additional clock cycle following the address (or command modifier) phase in order to prevent contention on bidirectional IO pins.

It will cause drive conflict if the SPI NOR device’s timing sequence does not contain dummy cycles after the command/address cycles. Such drive conflict might result in reading wrong data value. The problem usually happens when reading a SPI slave's register space.

Workaround

For FlexSPI memory device that supports multi IO Read command with zero latency cycle between address phase and data phase, use single line mode for read command, or use different data line to issue commands and read data.

The official NXP BSP release uses a signal line (1S-1S-1S) mode, but not multiple IO modes when access FlexSPI device registers.

ERR006940: Core: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used

Description

Arm Errata 776924: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used

Affects: Cortex-M4F

Fault Type: Programmer Category B

Fault Status: Present in: r0p0, r0p1 Open.

On Cortex-M4 with FPU, the VDIV and VSQRT instructions take 14 cycles to execute. When an interrupt is taken a VDIV or VSQRT instruction is not terminated, and completes its execution while the interrupt stacking occurs. If lazy context save of floating point state is enabled then the automatic stacking of the floating point context does not occur until a floating point instruction is executed inside the interrupt service routine.

Lazy context save is enabled by default. When it is enabled, the minimum time for the first instruction in the interrupt service routine to start executing is 12 cycles. In certain timing conditions, and if there is only one or two instructions inside the interrupt service routine, then the VDIV or VSQRT instruction might not write its result to the register bank or to the FPSCR.

Workaround

A workaround is only required if the floating point unit is present and enabled. A workaround is not required if the memory system inserts one or more wait states to every stack transaction.

There are two workarounds:

1) Disable lazy context save of floating point state by clearing LSPEN to 0 (bit 30 of the FPCCR at address 0xE000EF34).

2) Ensure that every interrupt service routine contains more than 2 instructions in addition to the exception return instruction.

ERR050703: DEBUG: Trace function cannot be used on A35 cores

Description

The A35 cores inside the i.MX 8XL are not accessible through the debug Trace interface. It is therefore not possible to capture a trace of the CA35 program execution.

Workaround

No workaround available

ERR050833: Core: Some AT instructions executed from EL3 might incorrectly report a domain fault

Description

Arm errata 799764

Affects: Cortex-A35

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r0p1, r0p2, r1p0. Open.

Address translation instructions executed from EL3 and targeting EL1 or EL0 might report an incorrect result in the PAR when the HCR_EL2.DC bit is set.

Configurations Affected:

This erratum affects all configurations of the processor.

Conditions:

1. The core is executing at Exception level 3 in AArch64.

2. The core executes one of the following address translation instructions:

• AT S1E0R, AT S1E0W

• AT S1E1R, AT S1E1W

• AT S12E0R, AT S12E0W

• AT S12E1R, AT S12E1W

3. The Exception level targeted by the address translation instruction is Non-secure and AArch32.

4. HCR_EL2.DC is set to 1.

5. The DACR is programmed so that domain 0 would cause a domain fault if the HCR_EL2.DC bit had not been set. Note that this is the default value out of reset.

Implications:

The PAR register will incorrectly report that a domain fault occurred.

Workaround

Secure software can set the DACR[1:0] to 0b01 before executing the address translation instruction. It should restore the previous DACR value before returning to a lower Exception level.

ERR050878: Core: Processor reset asserted asynchronously could corrupt FPB comparator registers and remap to wrong address

Description

Arm errata 1299509

Affects: Cortex-M4, Cortex-M4F

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1. Open.

Normally, the debugger uses the Flash Patch and Breakpoint (FPB) unit for breakpoints or for patching ROM code during debugging. However, you can also use the FPB functionality as a method of in-the-field ROM code patching. On the Cortex-M4 processor, the processor reset can be asynchronously asserted and this could potentially cause problems if the processor is in the middle of writing to debug components (which are not reset by the processor reset) such as the FPB unit.

Configurations Affected:

A Cortex-M4 processor that is configured with debug. Conditions

1) The processor is programming the FPB to remap an address in ROM to fetch a replacement opcode or vector from an area in RAM.

2) The processor is asynchronously reset during the programming of the FPB.

3) The data is incorrectly written to the FPB register due to metastability.

Implications:

In the unlikely event of this occurring it may cause incorrect functional operation of the processor to occur. If the FPB is programmed with incorrect data it may cause the processor to unintentionally patch instructions.

Workaround

The problem does not arise if the FPB is not used to patch instructions. If the FPB is used to patch instructions, a software workaround is to ensure that the FPB is globally disabled before programming any comparators. If code exists which programs the FPB other than at reset, the software workaround should include: 1. Disable the FPB. 2. Program the individual comparators required. 3. Explicitly disable the individual comparators not required. 4. Re-enable the FPB. This sequence prevents any corruptly programmed FPB comparators from being activated.

ERR051041: LPIT: CVAL cannot be read correctly during timer running

Description

The LPIT implements a functional clock domain for the counter and a bus clock domain for the register interface. The CVAL register increments on each clock cycle of the functional clock domain. Reading the register value happens on the bus clock domain. As these clock domains are not synchronous, there is a possibility that the register is read whilst the counter value is updating. This can lead to reading incorrect values as some bits of the counter may have settled whilst others are still transitioning to the new state.

Workaround

There should be no need to read the timer value since the timer is normally used to generate a periodic interrupt. However, if the timer value needs to be read, software can read the register more than once until the value matches the previous value. This ensures that the read operation did not coincide with the timer update and the value read is the actual timer value.

ERR050949: ADC: Differential ADC mode is not supported

Description

Two separate issues:

1) Sign-extend data is not passed to accumulator for averaging functionality, affecting differential mode.

2) Comparator sign issue, affecting only differential mode.

Workaround

Use single-ended ADC mode

ERR010946: DRAM: In LPDDR4 mode: Auto refresh must be disabled during DQS2DQ training

Description

If auto refresh is enabled during DQS2DQ training, a JEDEC Specification violation may occur. Auto refresh must be disabled during DQS2DQ training, which is performed during initial power-up (cold boot).

Workaround

For initial power-up (cold boot), disable auto refresh during DQS2DQ training. For self-refresh (warm boot), do not run DQS2DQ training. Restore the saved register values prior to self-refresh entry. Scripts provided by NXP’s DRAM RPA (Register Programming Aid) implement the required workaround.

ERR010944: DRAM: In LPDDR4 mode, tMPCWR timing violation in incremental DQS2DQ Training

Description

In LPDDR4 mode with incremental DQS2DQ Training enabled and speed grade > 2133 Mbps, the hardware incremental dqs2dq training routine performs Power Down (PD) Entry-Exit Cycle to reset the MPC WR-RD FIFO pointers in the DRAM. The PUB sends the MPC WRFIFO command after waiting for tXP from PD Exit. However, JEDEC specification requires waiting an additional tMPCWR after tXP timing. This extra tMPCWR timing is not handled by the PUB Training algorithm, resulting in a violation of tMPCWR JEDEC parameter.

Workaround

Do not run incremental DQS2DQ Training of the PHY in LPDDR4 mode.

ERR003777: GPT: Possibility of additional pulse on src_clk when switching between clock sources

Description

There is a possibility of an extra pulse on SCLK in the GPT when switching between the clock sources.

Workaround

Changing the clock source should only be done when the GPT is disabled. A way to accomplished this is as follows:

Disable GPT—Write 1'b0 to EN bit of GPTCR

Disable interrupts—Write 6'b000000 in Bits [5:0] of GPTIR

Configure Output Mode to unconnected/ disconnected—Write zeros in OM3, OM2, OM1 in GPTCR

Disable Input Capture Modes—Write zeros in IM1,IM2 in GPTCR

Change clock source CLKSRC in GPTCR

Clear Status register—Write 003F in GPTSR

Set ENMOD in GPTCR

ENABLE GPT—Write 1'b1 to EN bit of GPTCR. The GPTSR should not be read immediately after changing the clock source (a wait of at least one SCLK is required).

ERR050341: DRAM: LPDDR4 VREF training may result in a non-optimal value

Description

During LPDDR4 initialization, when performing LPDDR4 VREF training (through DDRPHY_PIR[VREF]), a discrepancy may be observed between the training-generated DDRPHY_MR14 value and an actual signal-measured VREF_DQ value such that the “trained” DDRPHY_MR14 value may not result in the most optimal VREF_DQ setting. However, the discrepancy is minimal such that no DRAM data failures have occurred due to this.

Workaround

A software workaround has been included in the SCU firmware (SCFW) since version 1.4.0. The software workaround has been included in the MX8QXP DDR Register Programming Aid (RPA) since RPA version 14.

ERR050793: Debug software must check the power status of each individual core before it interacts with it.

Description

The Cortex-A core complex provides separate power domains for the cluster-level logic and individual cores. Debug access to an unpowered core within a Cortex-A cluster that has been powered may fail or hang.

Improper isolation of the debug interface during core power up results in a transient gap which may cause unpredictable results of debug accesses to the respective core. Access to the shared cluster power domain which contains the EDPRSR register (for example) is not an issue.

Workaround

Debug software must check the power status of each individual core before it interacts with it (technically the power domain of the specific processor) in order to ensure that the core is already powered up.

The status can be found by looking at the EDPRSR.PU bit for the particular core

ERR050606: LPSPI: TCR value does not get resampled when polling the register

Description

Reading the Transmit Command Register will return the current state of the command register.

Following a write to the TCR (Transmit Command register), if the user continuously reads the TCR (polls the register), then the read content no longer represents the contents of the Transmit Command register if it updates due to internal logic following the first read. The same value shall continue to be read.

Workaround

After reading the Transmit Command Register must always access a different register in between subsequent reads from TCR.

ERR011189: ADC: When CFG[TPRICTRL] = 1, the first conversion of the higher priority trigger source will be performed twice

Description

When CFG[TPRICTRL] = 1, the first conversion of the higher priority trigger source will be performed twice and this will result in the first conversion result being incorrect.

Workaround

If possible, use CFG[TPRICTRL] = 0 and immediately abort the present conversion when a higher priority trigger occurs.

If it is required to allow the present conversion to complete first, and have CFG[TPRICTRL] = 1, then the first command executed by the higher priority trigger must be a dummy conversion. The dummy conversion can be performed by chaining commands. This is achieved by configuring the command register of the corresponding trigger to perform a single conversion and set the CMDHx[NEXT] field in this command register to the command buffer that will perform the desired conversion(s).

The dummy conversion should use the compare function to “store on true” and be configured so that the comparison on the result will always be false, to prevent the dummy result being placed in the result FIFO. This is done by setting,

CMDHx[CMPEN] = 2’b10 (enable the compare function to store on true)

CVx[CVL] = CVx[CVH] = 1 (configure comparison condition to always be false)

ERR011077: ADC: Inside range compare function does not work if CVH=0.

Description

When the ADC compare function is enabled and configured for inside range operation (CVL > CVH), the compare will always return true if CVH = 0.

Workaround

When using the inside range compare operation, the following condition must be met:

CVL > CVH > 0.

ERR051029: ENET_QOS: Gate Control List Switching is Incorrect for Intermediate Cycles When CTR is Less Than GCL Execution Time

Description

Impacted Configurations:

DWC_ether_qos configurations in which you select Enable Enhancements to Scheduling Traffic (EST) feature.

Parameter:

DWC_EQOS_AV_EST == 1

Versions Affected: 5.00a and later

Defect Summary:

The EST (Enhancements to Scheduled Traffic) scheduler switches to the next Gate Control List (GCL) after executing the current Gate-Control List (GCL) regardless of the difference between the Cycle Time Register (CTR) value, and sum of the Base Time Register (BTR) and Time Intervals (TI) of the GCL rows whose execution is complete. If the GCL execution takes longer than the cycle time, the GCL is truncated at the cycle time, and the subsequent loop begins at

BTR + N * Cycle Time, where N represents the iteration number, an integer.

However due to the defect, in the following situations, the GCL incorrectly updates the internal BTR twice. As a result, the GCL skips the execution of the next GCL loop:

CTR value is less or greater than GCL loop execution time.

The difference between the CTR and the sum of the BTR and Time Intervals of completely executed GCL rows is less than 8 PTP clock periods expressed in ns.

Impacted Use Cases:

The CTR value that you programmed is not equal to GCL execution time. GCL execution time is as follows:

BTR + N * sum of time intervals of valid GCL rows.

Workaround

Program the CTR, BTR, and Time Intervals of the GCL rows such that the difference between the CTR and the sum of the BTR and time intervals of fully executed GCL rows is greater than 8 PTP clock periods expressed in ns.

Alternatively, the CTR must be equal to the sum of the BTR and Time Intervals of the fully executed GCL rows.

ERR011194: PCIE: Plesiochronous loopback is not functional in PCIe Gen3

Description

Customers should be using mesochronous loopback when sending arbitrary bit streams.

Plesiochronous loopback: Is loopback from Rx back to Tx after the PCIe elastic buffer function in the PCS.

The intent of this reverse loopback scheme is to send arbitrary bit-streams through the elastic FIFO on the Rx side of the PCS and back through the Tx side of the PCS into the PMA. However, this does not work at Gen3 speed. This mode is not practical because the entire PCS PCIe pipeline is designed for protocol-dependent data, and requires many bypass paths to enable arbitrary bit streams through it. Moreover, there is no way to support elasticity when the bit-stream is protocol-agnostic, rendering the elastic FIFO useless.

Mesochronous (meso) loopback: Is loopback from Rx back to Tx before any elastic buffer, hence requiring 0ppm frequency difference between TxClk and RxClk, and requires TxClk and RxClk to be phase-adjusted using an automatic CDR skip-bit routine (as described in the PUG). Meso loopback assumes that the intersection set of the setup+hold margin for all 20 bits in the Rx to Tx STA path has a large open window. The SDC constraints were originally intended to contain max_delay and min_delay constraints to ensure this, but customers may not have optimized the window. Historically, mesochronous mode rarely worked at the highest protocol speeds due to this dependency on customer's timing optimization.

Workaround

Customers must use meso loopback when sending arbitrary bit streams.

ERR051393: Arm/Cortex-A core memory corruption

Description

A race condition in the Cortex-A CPU subsystem during initialization can cause memory setup values to be incorrectly applied to some Cortex-A cluster internal memories.

This may lead to memory corruption on some devices.

Workaround

An SCU firmware revision implementing SCF-838 modifies the Cortex-A CPU subsystem initialization sequence to avoid the race condition.

ERR006939: Core: Interrupted loads to SP can cause erroneous behavior

Description

Arm Errata 752770: Interrupted loads to SP can cause erroneous behavior

This issue is more prevalent for user code written to manipulate the stack. Most compilers will not be affected by this, but please confirm this with your compiler vendor. MQX™ and FreeRTOS™ are not affected by this issue.

Affects: Cortex-M4, Cortex-M4F

Fault Type: Programmer Category B

Fault Status: Present in: r0p0, r0p1 Open.

If an interrupt occurs during the data-phase of a single word load to the stack-pointer (SP/R13), erroneous behavior can occur. In all cases, returning from the interrupt will result in the load instruction being executed an additional time. For all instructions performing an update to the base register, the base register will be erroneously updated on each execution, resulting in the stack-pointer being loaded from an incorrect memory location.

The affected instructions that can result in the load transaction being repeated are:

1) LDR SP,[Rn],#imm

2) LDR SP,[Rn,#imm]!

3) LDR SP,[Rn,#imm]

4) LDR SP,[Rn]

5) LDR SP,[Rn,Rm]

The affected instructions that can result in the stack-pointer being loaded from an incorrect memory address are:

1) LDR SP,[Rn],#imm

2) LDR SP,[Rn,#imm]!

Conditions:

1) An LDR is executed, with SP/R13 as the destination.

2) The address for the LDR is successfully issued to the memory system.

3) An interrupt is taken before the data has been returned and written to the stack-pointer.

Implications:

Unless the load is being performed to Device or Strongly-Ordered memory, there should be no implications from the repetition of the load. In the unlikely event that the load is being performed to Device or Strongly-Ordered memory, the repeated read can result in the final stack-pointer value being different than had only a single load been performed.

Interruption of the two write-back forms of the instruction can result in both the base register value and final stack-pointer value being incorrect. This can result in apparent stack corruption and subsequent unintended modification of memory.

Workaround

Most compilers are not affected by this, so a workaround is not required.

However, for hand-written assembly code to manipulate the stack, both issues may be worked around by replacing the direct load to the stack-pointer, with an intermediate load to a general-purpose register followed by a move to the stack-pointer.

If repeated reads are acceptable, then the base-update issue may be worked around by performing the stack pointer load without the base increment followed by a subsequent ADD or SUB instruction to perform the appropriate update to the base register.

ERR051032: KS1 overconsumption in some subsystems

Description

To get normal KS1 power consumption, after power on (either from full off or from KS0 state) some subsystems must be powered up at least once, and can subsequently be powered down again, before transitioning into KS1 state.

Specifically the Connectivity, ADMA and LSIO subsystems must be powered up. This should be done from SW interacting with the SCU FW using resources as defined in the SCFW documentation.

If KS1 is not used then the subsystems need not be powered up.

Workaround

To achieve normal KS1 power consumption, make sure Connectivity, ADMA and LSIO have transitioned to the ON state at least once after a power on event, and before KS1 is entered.

ERR011211: LPI2C: Slave Transmit Data Flag may incorrectly read as one when LPI2C_SCFGR1[TXCFG] is zero

Description

When LPI2C_SCFGR1[TXCFG]=0 and the address valid flag is set for a slave receive transfer, if the previous transfer was slave-transmit, then the transmit data flag (LPI2C_SSR[TDF]) may incorrectly assert for one cycle. If the transmit data interrupt or DMA request is enabled, then the interrupt or DMA request can pend in the NVIC or DMA controller.

Workaround

There are two possible workarounds:

- Set LPI2C_SCFGR1[TXCFG].

- Software must ensure the transmit data interrupt and DMA request are only enabled after the address valid flag is set and are disabled at the end of a transfer.

ERR009005: Core: Store immediate overlapping exception return operation might vector to incorrect interrupt

Description

Arm Errata 838869: Store immediate overlapping exception return operation might vector to incorrect interrupt

Affects: Cortex-M4, Cortex-M4F

Fault Type: Programmer Category B Rare

Fault Status: Present in: r0p0, r0p1 Open.

The Cortex-M4 includes a write buffer that permits execution to continue while a store is waiting on the bus. Under specific timing conditions, during an exception return while this buffer is still in use by a store instruction, a late change in selection of the next interrupt to be taken might result in there being a mismatch between the interrupt acknowledged by the interrupt controller and the vector fetched by the processor.

Configurations Affected

This erratum only affects systems where writeable memory locations can exhibit more than one wait state.

Workaround

For software not using the memory protection unit, this erratum can be worked around by setting DISDEFWBUF in the Auxiliary Control Register.

In all other cases, the erratum can be avoided by ensuring a DSB occurs between the store and the BX instruction. For exception handlers written in C, this can be achieved by inserting the appropriate set of intrinsics or inline assembly just before the end of the interrupt function, for example:

ARMCC:

...

schedule_barrier();

asm{DSB};

schedule_barrier();

}

GCC:

...

asm volatile ("dsb 0xf" ::: "memory"); volatile ("dsb 0xf" ::: "memory");

}

ERR050829: Core: Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation

Description

Arm errata 1608096

Affects: Cortex-A35

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r0p1, r0p2, r1p0. Open.

A speculative Address Translation (AT) instruction translates using registers that are associated with an out-of-context translation regime and caches the resulting translation in the TLB. A subsequent translation request that is generated when the out-of-context translation regime is current uses the previous cached TLB entry producing an incorrect virtual to physical mapping.

Configurations Affected:

All configurations are affected.

Conditions:

1. A speculative AT instruction performs a table walk, translating a virtual address to a physical address using registers associated with an out-of-context translation regime.

2. Address translation data that is generated during the walk is cached in the TLB.

3. The out-of-context translation regime becomes current and a subsequent memory access is translated using previously cached address translation data in the TLB, resulting in an incorrect virtual to physical mapping.

Implications:

If the above conditions are met, the resulting translation would be incorrect.

Workaround

When context-switching the register state for an out-of-context translation regime, system software at EL2 or above must ensure that all intermediate states during the context-switch would report a level 0 translation fault in response to an AT instruction targeting the out-of-context translation regime. A workaround is only required if the system software contains an AT instruction as part of an executable page.

ERR050498: LPUART: Setting and immediately clearing SBK bit can result in transmission of two break characters

Description

This errata applies to the LPUART on CM4 and SCU only. It does not apply to the general purpose LPUARTs.

When the LPUART transmitter is idle (LPUART_STAT[TC]=1), two break characters may be sent when using LPUART_CTRL[SBK] to send one break character. Even when LUART_CTRL[SBK] is set to 1 and cleared (set to 0) immediately.

Workaround

To queue a single break character via the transmit FIFO, set LPUART_DATA[FRETSC]=1 with data bits LPUART_DATA[T9:T0]=0.

ERR050812: System Timer and TimeStamp Counter can stop under some slow clock conditions.

Description

The A-core clock diagram shows that the core can be run from the 24Mhz oscillator for long term. This is not supported for the A cores due to a problem with the interaction with the system clock value. The minimum frequency for the A-cores for long term running is 200Mhz.

The M-cores must not run slower than 24 MHz. The clocking structure shows that slower frequencies can be selected for the M-cores but this is not supported.

Workaround

Do not run the A-cores at a lower frequency than 200 MHz and do not run the M-cores slower than 24 MHz.

The SCU Firmware does not protect against unsafe settings.

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

Description

If the receive or transmit bit clock (BCLK) is internally generated, enabled with DIV > 0 and is then disabled, due to software or Stop mode entry, and the BCLK is enabled again, the clock is not generated.

Workaround

If the receive or transmit BCLK is internally generated and a DIV value greater than 0 is used, the SAI must be reset before the BCLK is re-enabled. This is achieved by writing the SR bit in the respective RCSR or TCSR register first to 1 and then immediately to 0.

ERR050827: ADC0 generates phantom ADMA subsystem interrupts to SCU

Description

In i.MX8XL, ADC0 can assert the ADMA DSC interrupt to SCU when the debug STOP bit is modified.

The purpose of that interrupt is to wake up the system (through the SCU) in order to respond to an ADC event.

Workaround

Mask off ADMA subsystem interrupt #27 (ADC0 wakeup) in ADMA DSC:

DSC_MSI_ADDR_IRQMASK[27] == 1

ERR009004: Core: ITM can deadlock when global timestamping is enabled

Description

ARM ERRATA 806422

The Cortex-M4 processor contains an optional Instrumentation Trace Macrocell (ITM). This can be used to generate trace data under software control, and is also used with the Data Watchpoint and Trace (DWT) module which generates event driven trace. The processor supports global timestamping. This allows count values from a system-wide counter to be included in the trace stream.

When connected directly to a CoreSight funnel (or other component which holds ATREADY low in the idle state), the ITM will stop presenting trace data to the ATB bus after generating a timestamp packet. In this condition, the ITM_TCR.BUSY register will indicate BUSY.

Once this condition occurs, a reset of the Cortex-M4 is necessary before new trace data can be generated by the ITM.

Timestamp packets which require a 5 byte GTS1 packet, or a GTS2 packet do not trigger this erratum. This generally only applies to the first timestamp which is generated.

Devices which use the Cortex-M optimized TPIU (CoreSight ID register values 0x923 and 0x9A1) are not affected by this erratum.

Workaround

There is no software workaround for this erratum. If the device being used is susceptible to this erratum, you must not enable global timestamping.

ERR050272: DLL lock status bit not accurate due to timing issue

Description

Due to this erratum, the DLL lock status register STS2[AREFLOCK, ASLVLOCK] bit for channel A (or STS2[BREFLOCK, BSLVLOCK] bit for channel B) may indicate DLL is locked before DLL is actually locked.

Workaround

Once STS2[AREFLOCK, ASLVLOCK] bit for channel A (or STS2[BREFLOCK, BSLVLOCK] bit for channel B) is set to one, insert a delay of 4 µs before use of FlexSPI.

ERR011184: ADC: Setting CFG[PUDLY] = 0 may result in an incorrect trigger delay time

Description

In the case where CFG[PWREN] = 0, ADC analog circuitry is powered down, setting CFG[PUDLY] = 0 and TCTRLx[TDLY] set to a non-zero value will result in the trigger delay being twice the programmed time.

Workaround

If the application permits, set CFG[PWREN] =1, and then set TCTRLx[TDLY] values for each trigger source.

If the desire is to clear CFG[PWREN] to zero to reduce current consumption, and if CFG[PUDLY] must be zero, then set TCTRLx[TDLY] so that twice this value is equal to, or greater than, the minimum power-up delay time.

ERR050821: ADC: Loop With automatic channel Increment (LWI) is not supported.

Description

Loop with automatic channel increment may produce corrupt data or skip channels.

The loop function without channel increment works as intended.

Workaround

Disable the automatic channel increment by setting the LWI bit in the ADC CMDH register to 0. This applies to registers CMDH0 - CMDH4.

ERR050831: Core: Mismatch between EDPRSR.SR and EDPRSR.R

Description

Arm errata 857487

Affects: Cortex-A35

Fault Type: Programmer Category C

Fault Status: Present in r0p0, r0p1, r0p2, r1p0. Open.

The processor provides access to the EDPRSR through the APB interface. If this access is done at the same time as the core leaves a Warm reset, then a subsequent read of the same register will read an incorrect value of the SR field.

Configurations affected:

This erratum affects all configurations of the processor.

Conditions:

1. The core is in Warm reset.

2. A debugger reads the EDPRSR register over the APB interface.

3. The core comes out of Warm reset during the APB read.

4. A second APB read is made to the EDPRSR register.

Implications:

The first read of the EDPRSR will read the SR field and R field both as 0b1.

The second read of the EDPRSR.SR field will read 0b0 whereas the previous read of the EDPRSR.SR was 0b1 while in Warm reset. Because the first read took place while in Warm reset, the sticky bit should still be set on the second read.

Workaround

If the debugger reads the EDPRSR and sees both the SR and R fields set, then it must remember this result and on the next EDPRSR read treat the SR bit as if it was set.

ERR010930: PCIE: EOM single point sample error/valid result is not correct

Description

There is an eye monitor in the SerDes analysis which can monitor the following:

a. Error and valid bits of a certain duration

b. Eye width

c. Eye height

d. Eye area

However, there is a design issue with item (a) causing incorrect error/valid bit results.

Workaround

Customers must not use the error/valid count results to check the eye quality. Instead, use the eye width, eye height, or eye area.

ERR050948: FSM does not transition to reset state from IDLE when reg_pwren isn't set

Description

ADC: FSM does not transition to reset state from IDLE when reg_pwren isn't set. This is needed for power up delay.

Workaround

Set PWREN and wait for ADC Analog to power up before enabling ADC and accepting triggers

ERR010752: LPI2C: Slave Busy Flag may incorrectly read as zero when 10-bit address mode enabled in slave mode.

Description

When operating in slave mode with 10-bit addressing enabled and an address match is detected on the first address byte, the Slave Busy Flag, SSR[SBF], may incorrectly read as zero.

Workaround

When using the LPI2C in slave mode when 10-bit addressing is enabled, and the SSR[SBF] bit is read as a one, then the flag is set. If it is read as a zero, it must be read a second time and this second read will be the correct state of the bit.

ERR006941: Core: Asynchronous sampling of SWDIOTMS might cause incorrect operation of SerialWire/JTAG Debug Port

Description

Arm Errata 771919: Asynchronous sampling of SWDIOTMS might cause incorrect operation of

SerialWire/JTAG Debug Port

Status

Affects: Cortex-M4, Cortex-M4F

Fault Type: Implementation Category B Rare

Fault Status: Present in: r0p0, r0p1 Open.

Description

The signal SWDIOTMS is bi-directional and can be driven from either the debugger or the SWJ-DP, or pulled up by an external resistor during the turnaround periods.

The SerialWire protocol is defined with a high PARK bit at the end of the header before the turnaround period that precedes the ACK from the SWJ-DP. This ensures that the line is high, and the resistor keeps it high during the ACK period. Therefore, if the SWJ-DP does not respond, the debugger will reliably sample the line SWDIOTMS high during the missing ACK.

However, during the turnaround period after the ACK or read data there is no PARK bit to guarantee that the line is high immediately before the turnaround period. In this case, if the pull-up resistor does not pull the line high within a single SWCLKTCK cycle, the incorrect state of SWDIOTMS might be sampled.

Functionally, the logic is insensitive to the state of SWDIOTMS during these periods, but synthesis tools might introduce multiple path logic that is sensitive to SWDIOTMS glitches around the clock edges.

Conditions

All write transactions and some read transactions might be vulnerable to this erratum when both:

- Serial Wire mode is being used

- The physical implementation does not prevent glitch generation.

Implications

The SWJ-DP might sample SWDIOTMS incorrectly and enter an UNPREDICTABLE state. At the time of publication, ARM is not aware of any reports of observed failures due to this erratum.

Workaround

Check the following points after implementation:

1) Ensure that the evaluation of NextState in DAPSwjWatcher.v is not sensitive to SWDITMSSync1 when State_cdc_check has the value 10'b1100100000 (SWJ_SSLP).

2) Ensure that the following logic in DAPSwDpProtocol.v is implemented using AND gates or a CDC-safe mux for each bit:

assign ResetCountD = DBGDI & $_{\text{DBGDOEN ? (ResetCountReg+6'd1) : \{6\{1'b0\}\};}}$? (ResetCountReg+6'd1) : {6{1'b0}};

3) Ensure that the ResetCountReg flops in DAPSwDpProtocol.v are implemented using metastability-hardened cells if possible.

4) Ensure that the evaluation of NxtState in DAPSwDpProtocol.v is insensitive to DBGDI when State has any of the following values:

- 5'b01000 (SWDP_SLEPARKH)
- 5'b01010 (SWDP_SLETRNH2)
- 5'b01011 (SWDP_SLETRNH1)
- 5'b01100 (SWDP_SLETRNH0)
- 5'b10011 (SWDP_SLEPARKW)
- 5'b10100 (SWDP_SLETRNW3)
- 5'b10101 (SWDP_SLETRNW2)
- 5'b10110 (SWDP_SLETRNW1)
- 5'b10111 (SWDP_SLETRNW0)

5) Ensure that the following flops in DAPSwDpProtocol.v are implemented with CDC-safe recirculation muxes:

- SerBank
- SerDir
- SerAddr
- ShiftReg
- Parity
- ErrorChk
- WriteErr
- WbufReq

6) Ensure that the following flops in DAPJtagDpProtocol are implemented with CDC-safe recirculation muxes:

- JTAGcurr

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

Description

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

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

Workaround

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

ERR050533: DRC performance counters are not counting during DRC clock gating.

Description

The performance counters in the DRC are clockgated along with the DRC controller. As this occurs when the module is idle, most of the counters are not being incremented in this period. However there is an option to select a “time count” when CSV=8’00 for one of the counters

This will increment every cycle WHEN THE DRC IS NOT CLOCKGATED. Hence it is not real time

Definition of this count is changed from Time to Non-clock gated time

Workaround

Disable autoclock gating of the DRC if an accurate count is required, alternative is to keep the gating enabled and simply understand the exact meaning of the count

ERR050822: ADC: No support for pending trigger when same trigger is actively converting.

Description

Triggers on a currently converting trigger will be ignored.

Workaround

Wait until active trigger is complete, indicated by valid data loaded to FIFO, before triggering the same trigger again.

ERR050830: Core: PMU counter might be inaccurate when monitoring BUS_ACCESS and BUS_ACCESS_ST

Description

Arm errata 1010162

Affects: Cortex-A35 MPCore

Fault type: Programmer Category C

Fault status: Present in r0p0, r0p1, r0p2, r1p0. Open.

The Cortex-A35 processor implements a Performance Monitor Unit (PMU). The PMU allows programmers to gather statistics on the operation of the processor during runtime. Because of this erratum, the PMU counter values might be inaccurate when monitoring BUS_ACCESS and BUS_ACCESS_ST events.

Configurations Affected:

To be affected by this erratum, one or more of the following must be true:

• The processor is configured with an ACE or CHI bus interface

• The processor is configured with more than one core

• The processor is configured with an L2 cache

• The processor is configured with CPU cache protection

Conditions:

1. A performance counter is enabled and configured to count BUS_ACCESS or BUS_ACCESS_ST events.

2. A write or eviction occurs on the bus.

Implications:

The PMU counter will erroneously increment or erroneously fail to increment. The inaccuracy varies between cores. Some bus accesses for core 0 might be attributed to core 1. Some bus accesses for core 1 might be attributed to core 2 or core 3. Bus accesses for cores 2 and 3 will not be attributed to any core. The number of cores present in the configuration does not affect how the bus accesses are attributed. For example, some core 1 accesses might be attributed to cores 2 and 3 even if the configuration has only two cores. The impact on the final counter value in each core is high.

This might lead to inaccurate results when using the PMU to debug or profile code.

Workaround

The BUS_ACCESS and BUS_ACCESS_ST events can be counted accurately for core 0 by enabling the counter on both core 0 and core 1 and taking the total. This workaround requires core 1 to be present in the configuration and idle during testing. Similarly, the total count for core 0 and core 1 can be found by enabling the counter on all four cores and taking the total. This workaround requires cores 2 and 3 to be present in the configuration and idle during testing. In a configuration with four cores, the events can be used with an intensive memory test that runs on two cores to give a reasonable indication of maximum bandwidth for the cluster.

The event L2D_CACHE_WB counts write-backs from the L2 cache that are attributable to a core. For a test focused on cacheable memory bandwidth, this might be a suitable replacement for BUS_ACCESS_ST. This event includes:

• Write-backs as a result of evictions and cache maintenance instructions.

• Writes in read allocate mode (write-streaming mode).

L2D_CACHE_WB does not include:

• Write-backs as a result of snoop requests from outside of the cluster.

• Write-backs as a result of ACP interface accesses.

ERR010945: DRAM: PUB does not program LPDDR4 DRAM DDRPHY_MR22 prior to running DRAM ZQ calibration

Description

When the PHY Utility Block (PUB) initializes the DRAM, the DDRPHY_MR22 is programmed after ZQ Calibration. This may result in incorrect ZQ calibration results on the LPDDR4 DRAM side, because DDRPHY_MR22[CODT] works as the controller On Die Termination (ODT) replica during the Pull-Up calibration. Therefore the expected controller ODT must be programmed into DDRPHY_MR22 prior to the DRAM ZQ calibration.

Workaround

Run DRAM Initialization twice. Scripts provided by NXP’s DRAM RPA (Register Programming Aid) implement the required workaround.

ERR010858: LPCG: IP clock gating register synchronization logic is sensitive to root clock gating

Description

LPCG registers are accessed using a clock domain that is independent of the gated clock. LPCG registers require up to 4 cycles of the gated clock to ensure synchronization with the gating logic. Back-to-back writes to LPCG registers that occur within the 4-cycle window can be ignored by the LPCG logic and the LPCG register will not reflect the state of the gating logic.

Workaround

Software must ensure that at least 4 cycles of the gated clock elapse between writes to the LPCG registers. Writes to LPCG registers across the system interconnect will incur latency that avoids the LPCG synchronization window unless the gated clock is running at less than 24 MHz.

If software cannot ensure sufficient delay between LPCG writes to avoid the issue, LPCG must not be used for clock control. As an alternative, SCFW APIs can be used to gate clock root inputs to LPCG cells.

ERR050607: LPSPI: TCR[FRAMSZ] can be ignored when TCR[TXMSK]=1b1

Description

TCR (Transmit Command Register) is used to write new command word to the LPSPI transmit FIFO.

TCR[FRAMESZ] configures the frame size of the data to be transmitted in number of bits equal to (FRAMESZ + 1). When TCR[TXMSK] is set, transmit data is masked (no data is loaded from transmit FIFO and output pin is tristated). In master mode, the Transmit Data Mask bit will initiate a new transfer which cannot be aborted by another command word; the Transmit Data Mask bit will be cleared by hardware at the end of the transfer. TCR[CONTC] controls the continuous transfer mode. TCR[CONTC]=1b1 enables continuous transfer. In master mode, continuous transfer will keep the PCS asserted at the end of the frame size, until a command word is received that starts a new frame.

If command word is written with TCR[TXMSK]=1 and TCR[FRAMESZ]>32 and the next command word with TCR[CONTC]=0 is in the FIFO then at the end of any 32-bit word of the first command, the frame will terminate early and negate PCS.

Workaround

There are two workarounds:

1. Do not write a 2nd command word after writing command word with TXMSK=1 and FRAMESZ>32 until the first one has completed.

OR

2. Divide the command word into multiple command words with TXMSK=1 and FRAMESZ=32 (or remainder) using a continuous transfer.

ERR011370: PCIE: EP, PM_PME: L1 Exit Does Not Occur when PME Service Timeout Mechanism Expires

Description

Impacted Configuration(s): Upstream Port configurations:

device_type =4’b0000, located at PCIEX1_CTRL0, address 0x5f140000, bit[27:24]

Defect Summary:

When a function issues a PM_PME Message, it sets the PME_Status bit. If the Downstream port has not cleared the PME_Status bit within 100ms, a PME Service timeout occurs.

At this point, the Upstream port must resend the PM_PME message.

In the current implementation of the controller, the PME Service timeout does not trigger an exit from L1 to resend the PM_PME message.

System Usage Scenario:

Upstream ports using a wake-up mechanism followed by a power management event (PME) message.

Consequence(s):

The defect has the following effect:

The PME service routine cannot make forward progress until the PM_PME message is resent.

Workaround

Poll the PME_Status bit after sending the PME message to exit L1 state. If this bit remains 1 for 100ms or more, SW must re-toggle bit 8 “APPS_PM_XMT_PME” of HSIO GPR register "PCIEX1_CTRL2", address 0x5f140008.

ERR011543: FlexCAN: Nominal Phase SJW incorrectly applied at CRC Delimiter

Description

During the reception of a CAN-FD frame when the Bit Rate Switch (BRS) is enabled, the Synchronization Jump Width (SJW) for the CRC Delimiter bit is incorrectly defined by the Nominal Phase SJW. The CAN specification stipulates that the CRC Delimiter bit should have a SJW set by the Data Phase SJW.

When a resynchronization event is triggered for the CRC delimiter bit (recessive in correct operation), the sample point will be adjusted by an amount as defined by the Nominal Phase SJW rather than the specified Data Phase SJW. This may result in the incorrect detection of a dominant bit leading to a CAN error frame. However, as the CRC delimiter bit position will only apply the SJW upon the detection of an unexpected dominant bit on the CAN bus, an error frame is already likely. For the case the SJW is applied at the CRC delimiter and a recessive bit is not detected, the receiving node will issue an error frame.

The CAN protocol is designed to handle resynchronization errors and hence the CAN bus will recover from the insertion of the incorrect SJW at the CRC delimiter. Upon detecting the error frame the transmitting node will re-transmit the frame.

The following FlexCAN configurations are not affected:

- Classical CAN frames (CAN 2.0B)
- CAN FD frames with bit rate switch disabled (BRS = 0)
- CAN FD frames with Nominal Phase SJW equal to Data Phase SJW
- CAN FD transmissions

Configuration for the FlexCAN:

- Nominal Phase SJW is configured by the Resync Jump Width bit in the CAN Control Register 1 (CAN_CTRL1[RJW]) or by the Extended Resync Jump Width bit in the CAN Bit Timing Register (CAN_CBT[ERJW])
- Data Phase SJW is configured by the Fast Resync Jump Width bit in the CAN FD Bit Timing Register (CAN_FDCBT[FRJW])

Workaround

The robustness of the CAN protocol ensures that the receiver automatically recovers from the application of the incorrect SJW. The CAN protocol is designed to recover from resynchronization errors and hence any frame that is not correctly received will be re-sent by the transmitting node.

ERR051198: PWM: PWM output may not function correctly if the FIFO is empty when a new SAR value is programmed

Description

When the PWM FIFO is empty, a new value programmed to the PWM Sample register (PWM_PWMSAR) will be directly applied even if the current timer period has not expired.

If the new SAMPLE value programmed in the PWM_PWMSAR register is less than the previous value, and the PWM counter register (PWM_PWMCNR) that contains the current COUNT value is greater than the new programmed SAMPLE value, the current period will not flip the level. This may result in an output pulse with a duty cycle of 100%.

Workaround

Program the current SAMPLE value in the PWM_PWMSAR register before updating the new duty cycle to the SAMPLE value in the PWM_PWMSAR register. This will ensure that the new SAMPLE value is modified during a non-empty FIFO, and can be successfully updated after the period expires.

ERR050819: ADC: Higher Priority Trigger Interrupt is not supported.

Description

The ADC can be configured to abort the current conversion and start a new one when a higher priority trigger occurs. This is not supported and should not be used.

Higher priority triggers must wait until the current conversion has finished.

Workaround

Set CFG.TPRICTRL to '1' to disable higher priority trigger interrupts.

ERR010629: USDHC: EMMC HS200/SD SDR104 tuning process may fail based on delay cell number

Description

In the eMMC HS200/SD SDR104 tuning process, the tuning process may fail if the first pass delay cell number specified is <10 during the tuning process.

If an internal clock is used as the tuning clock (recommended), then the delay cell number for first pass tuning command will be more than 10. In this case, there is no issue. If a feedback clock from the pad is used as a tuning clock (not recommended), then it is possible that the delay cell number for first pass tuning command will be less than 10. In this case, the tuning process will fail because the tuning state will judge the tuning window one cycle before the tuning window is calculated.

Workaround

If the tuning process fails because the first pass delay cell number specified is less than 10, the USDHC_TUNING_CTRL[TUNING_START_TAP] field should be set to 10.

ERR050314: eDMA: High priority channel cannot preempt a low priority channel

Description

When the Enhanced Direct Memory Access (eDMA) module is in fixed arbitration mode, a high priority channel cannot preempt a lower priority channel unless the high priority channel’s Enable Channel Preemption bit is set (CHn_PRI[ECP]=1).

Workaround

Set the CHn_PRI[ECP] bits for any high priority channels needing to preempt a lower priority channel.

Two items to note when implementing this workaround. First, the high priority channel may be preempted if it executes a normal start and if NBYTES divided by larger of either SSIZE or DSIZE is greater than or equal to three (NBYTES/S{D}SIZE >= 3). Second, any channel with NBYTES divided by the larger of either SSIZE or DSIZE less than three (NBYTES/S{D}SIZE < 3) cannot be preempted regardless of the ECP setting.

ERR050873: USDHC: External Pull-up needed on pin ENET0_RGMII_TX_CTL to boot from USDHC1

Description

SD-card boot may fail if no external pull-up resistor is present on pin ENET0_RGMII_TX_CTL.

The Boot ROM uses the UHDSC1 controller with I/O multiplexed on ENET0_RGMII pins for SD-card boot. ENET0_RGMII_TX_CTL serves as USDHC1_RESET_B. This pin has a 50K pull-down resistor configured as reset default. However, for SD-card operation, this pin needs to be high during reset to allow the SD-card to power-up whilst the CPU is resetting and starts to boot. pull-down resistor configured as reset default. However, for SD-card operation, this pin needs to be high during reset to allow the SD-card to power-up whilst the CPU is resetting and starts to boot.

Workaround

If booting from SD-card is required:

- add an external 10K pull-up resistor on pin ENET0_RGMII_TX_CTL to override the internal pull-down.

- turn off the the internal pull-down early in the application or SCU software during board initialization by writing 0x3 to the PULL bit-field in the ENET0_RGMII_TX_CTL pad/mux control register. This is to avoid current flowing through the pull-up/pull-down. The current will be very low (< 50 uA) but can be avoided.

ERR010947: DRAM: DQS/DQSN glitch suppression resistors must be enabled during read-leveling

Description

By default DQS/DQSN glitch suppression resistors are disabled. When external DQS/DQSn are not driven to valid differential states, the DQS cell's core-side outputs become unknown. This causes errors in the read-leveling gate training.

Workaround

Enable the strongest 355 ohm glitch suppression resistors during gate training. Scripts provided by NXP’s DRAM RPA (Register Programming Aid) implement the required workaround through DDRPHY_DX8SLbDQSCTL register.

ERR051078: ENET_AVB: ENETx_RGMII_TXC clock output may glitch when switching modes.

Description

When changing protocol speed in the Ethernet MAC (ENET) with the controller in RGMII mode it will change the speed of the clock generated on pin ENETx_RGMII_TXC. The clock frequency depends on the mode selected via the ENET_RCR[RMII_MODE], ENET_RCR[RMII_10T], ENET_RCR[RGMII_EN], and ENET_ECR[SPEED] registers. Clock speeds are 125 MHz (Gbit), 25 MHz (100T) and 2.5 MHz (10T).

Switching clock frequency between any of the 3 speeds can cause the TXC clock output ENETx_RGMII_TXC to not meet the minimum cycle time of the PHY and could manifest as a glitch. This is limited to the clock transitioning edge.

It is expected that in most cases the ethernet physical interface IC (PHY) will be able to ignore this glitch on the clock output since no transmission will be occurring during the initialization or clock speed changes, but the impact has to be determined by the individual PHY in use.

Workaround

Ensure the PHY is not affected by the glitch by powering-down the PHY or asserting reset to the PHY whilst the ENET module is being initialized or speed changes are being made.

ERR010948: DRAM: Timing Violation from Read/Write to MRW in LPDDR4 mode

Description

When software sends a MRW command in parallel with a Read/Write transaction, the Read/Write command can be followed by the MRW command, which can result in the following timing violations:

1. RD to MRW

2. RDA to MRW

3. WRA/MWRA to MRW

This can occur only in LPDDR4 mode. When the memory clock frequency is lower than 450 MHz, then one of above 3 violations may occur, when the memory clock frequency is NOT lower than 450 MHz, then above item 1 or item 2 violation may occur.

The above timing constraints were introduced in the LPDDR4 specification JESD209-4A.

Workaround

MRW commands sends in parallel with a Read/Write transaction must follow a specific sequence.

ERR050513: GPV access end in bus fault

Description

Access to the GPV address space (5C40 0000 to 5C4F FFFF) is only permitted during the DCD phase of the boot process, this primarily impacts the configuration of the QOS mechanism for the main interconnect which must therefore be setup at this time and cannot later be modified

Workaround

Leave the QOS functions at the default OR modify them by adding commands to the DCD file.

Do not access after DCD has run.

Legal information

Definitions

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

Disclaimers

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

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

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

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

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

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

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

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

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

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

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

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

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

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

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

Trademarks

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

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

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 11/2022
Document identifier: IMX8XLB0_0P79F