Mask Set Errata for Mask 0N87W

This report applies to mask 0N87W for these products:

- MIMX8MM6DVTLZAA
- MIMX8MM5DVTLZAA
- MIMX8MM4DVTLZAA
- MIMX8MM3DVTLZAA
- MIMX8MM2DVTLZAA
- MIMX8MM1DVTLZAA
- MIMX8MM6CVTKZAA
- MIMX8MM5CVTKZAA
- MIMX8MM4CVTKZAA
- MIMX8MM3CVTKZAA
- MIMX8MM2CVTKZAA
- MIMX8MM1CVTKZAA
- MIMX8MM5DVTLZCA
- MIMX8MM5DVTLZDA

Table 1. Errata and Information Summary

Erratum ID	Erratum Title
ERR003774	AIPS: Unaligned access to AIPS internal registers will result in an abort response.
ERR006941	Core: Asynchronous sampling of SWDIOTMS might cause incorrect operation of SerialWire/JTAG Debug Port
ERR006939	Core: Interrupted loads to SP can cause erroneous behavior
ERR009004	Core: ITM can deadlock when global timestamping is enabled
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
ERR050028	CoreSight: Coresight debug components cannot be auto-detected by debug tool
ERR050823	DDR: Register corruption possible when software triggered mode register (MR) operations performed in DDR4 mode.
ERR050380	DRAM data may be lost while exiting from DDR IO retention mode
ERR050805	DRAM: Controller automatic derating logic may not work when the LPDDR4 memory temperature is above 85ºC at initialization
ERR009535	ECSPI: Burst completion by SS signal in slave mode is not functional

Table continues on the next page...

Table 1. Errata and Information Summary (continued)

Erratum ID	Erratum Title
ERR009606	ECSPI: In master mode, burst lengths of 32n+1 will transmit incorrect data
ERR009165	ECSPI: TXFIFO empty flag glitch can cause the current FIFO transfer to be sent twice
ERR011207	FlexSPI: In rare conditions when FLEXSPI_AHBCR[PREFETCHEN] is set, incorrect data can be returned
ERR050044	GPC: GPU power domain on/off operation leads to unexpected behavior
ERR050522	GPU: GPU driver cannot get expected value from “onComp2” logic, it would fail to initialize correctly
ERR007805	I2C: When the I2C clock speed is configured for 400 kHz, the SCL low period violates the I2C spec of 1.3 uS min
ERR050080	IO: Degradation of internal IO pullup/pulldown current capability for IO’s continuously driven in a 3.3V operating mode
ERR050045	IOMUX: Setting ODE control bit of I2C IOs causes malfunction
ERR011437	IOMUX: The read data is always zero when ODE bit of ENET PHY IOs is set
ERR011341	IOMUXC: Missing HW force_en for SD_CLK pin of USDHC3
ERR011193	PCIE: EP, PM_PME: L1 Exit Does Not Occur when PME Service Timeout Mechanism Expires
ERR051198	PWM: PWM output may not function correctly if the FIFO is empty when a new SAR value is programmed
ERR050640	ROM: TCM 64-bit alignment issue in ROM for NAND page with bad block marker at unaligned offset
ERR050542	SAI: The Bit Count Timestamp Register (TBCTR, RBCTR) may return a live rather than latched Timestamp
ERR050448	SPDIF: SPDIF clock limitation
ERR051272	TMU: Bit 31 of registers TMU_TSCR/TMU_TRITSR/TMU_TRATSR invalid
ERR050101	USB: Endpoint conflict issue in device mode
ERR050150	WDOG: Watchdog WDOG_RESET_B_DEB reset SoC failed

Table 2. Revision History

Revision	Changes
0, 01/2019	Initial revision
1, 09/2021	The following errata were added. • ERR006941 • ERR050101 • ERR050150 • ERR050380 • ERR050448 • ERR050522 • ERR050542 • ERR050640 • ERR050805 • ERR050823 The following errata were revised.

Table continues on the next page...

Table 2. Revision History (continued)

Revision	Changes
	• ERR003774 • ERR006940
2, 6/2022	The following errata were added. • ERR051198 • ERR051272

ERR003774: AIPS: Unaligned access to AIPS internal registers will result in an abort response.

Description: Unaligned access to AIPS internal registers will return an abort response.

Workaround: Only aligned AIPS internal register access is supported. Software should not issue unaligned accesses to AIPS internal registers.

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 & ~DBGDOEN ? (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

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.

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.

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”);
}
```

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.

ERR050028: CoreSight: Coresight debug components cannot be auto-detected by debug tool

Description: One ROM entry inside DAP is not pointed to valid memory space. When connecting the debug tool (such as DS-5) to CPU, the debug tool hangs and cannot continue to auto-detect the debug components because no response is received from DAP.

Workaround: The user can create configuration file manually for the debug tool and detect the debug components.

ERR050823: DDR: Register corruption possible when software triggered mode register (MR) operations performed in DDR4 mode.

Description: The DDR controller allows user software to manually trigger a mode register read or write operation by setting the MRCTRL0.mr_wr=1. Under certain specific conditions listed below it is possible that DDR controller registers can be corrupted while an internal hardware driven MR access is occurring. The impact of the corruption depends on the registers being accessed.

This issue can only occur when:

1. DDR controller is configured in DDR4 mode

2. A mode register read or write operation is performed by user software by programming the register MRCTRL0.mr_wr=1

3. Separate DDR controller register R/W (or R/W1S or R/W1C) APB write accesses occur close together and

4. An internal hardware driven MR access occurs concurrently

4.a. Entering and exiting Self-Refresh or MPSM or when

4.b. Per DRAM Addressability mode (PDA) or Per Buffer Addressability(PBA) mode is enabled

Workaround: When performing a software driven MR access, the following polling sequence must be done automatically before performing other DDR controller register accesses:

1. Set the register MRCTRL0.mr_wr=1 (When the MR operation is complete, the DDRC automatically clears this bit)

2. Check the DDR Controller register MRSTAT.mr_wr_busy = 0 If not, go to step (2)

3. Check the DDR Controller register MRSTAT.mr_wr_busy = 0 again (for the second time). If not, go to step (2)

ERR050380: DRAM data may be lost while exiting from DDR IO retention mode

Description: There are two signals “PwrOkIn” and “atpg_mode” to DDR PHY input are not controlled properly in always on domain.

So when DDR PHY exits from IO retention mode, these two signals cannot be sure in state 0 to meet the PHY requirement.

It is possible that the IO retention mode may not function correctly and data in the DRAM may be lost.

Workaround: Keep the VDD_DRAM on while in DDR IO retention mode.

ERR050805: DRAM: Controller automatic derating logic may not work when the LPDDR4 memory temperature is above 85ºC at initialization

Description: LPDDR4 memories require periodic refreshes to maintain memory contents. Per the JEDEC specification JESD209-4 the memory refresh rate needs to increase and timings de-rated as the memory operational temperature exceeds vendor-defined temperature thresholds. The LPDDR4 Mode Register 4 (MR4) contains temperature/refresh rate information and a Temperature Update Flag (TUF).

An issue exists with the automatic derating logic of the DDR controller that only samples the LPDDR4 MR4 register when the Temperature Update Flag (TUF) field (MR4[7]) is 1'b1. If the LPDDR4 memory is initialized and starts operation above 85ºC (MR4[2:0] > 3'b011), the MR4 Temperature Update Flag (TUF) will not be set. The DDR Controller will therefore not automatically adjust the memory refresh rate or de-rate memory timings based on the LPDDR4 memory temperature. This may cause the controller incorrectly setting the refresh period, potentially cause the LPDDR4 memory losing data contents and lead to possible data integrity issues above 85ºC. The actual memory temperature threshold values may vary depending on memory vendors.

If the LPDDR4 memory temperature remains below 85ºC at initialization (Consumer-grade memory devices), then the derating logic works as intended, automatically adjusting the memory refresh period and memory timing during the entire system operation. The issue does not occur in this specific scenario since derating is not required.

This erratum does not impact other SoC supported DDR memory interfaces such as DDR4 or DDR3L.

Workaround: Analysis must be performed to determine if the particular design use case could be affected by the errata. There are three options to choose from based on this assessment:

Option 1: Keep the automatic derating logic of the DDR controller enabled. This is the default option and is suitable for designs that are not designated to boot at or above 85° (designs not affected by the errata).

Option 2: Disable the automatic derating logic of the DDR controller and apply fixed x2 refresh rate (0.5x refresh). This option is suitable for designs that are expected to boot at or above 85°C and memory's MR4[2:0] (Refresh Rate) DOES NOT report the following conditions:

3b101: 0.25x refresh, no de-rating

3b110: 0.25x refresh, with de-rating

3b111: SDRAM High temperature operating limit exceeded

Option 3: Disable the automatic derating logic of the DDR controller and apply fixed x4 refresh rate (0.25x refresh) together with derated timings for tRCD, tRAS, tRC, tRP and tRRD. This option is suitable for designs that are expected to boot at or above 85°C and memory's MR4[2:0] (Refresh Rate) DOES report the following conditions:

3b101: 0.25x refresh, no de-rating

3b110: 0.25x refresh, with de-rating

3b111: SDRAM High temperature operating limit exceeded

All the three options are implemented in the Register Programming Aid (RPA) spreadsheet (version v15 and later) where users can select the option that fits their use case.

For Option 2 and Option 3 it is highly recommended to consult with the DRAM vendor on supported temperature grades.

ERR009535: ECSPI: Burst completion by SS signal in slave mode is not functional

Description: According to the eCSPI specifications, when eCSPI is set to operate in the Slave mode (CHANNEL_MODE[x] = 0), the SS_CTL[x] bit controls the behavior of burst completion.

In the Slave mode, the SS_CTL bit should control the behavior of SPI burst completion as follows:

- 0—SPI burst completed when (BURST_LENGTH + 1) bits are received
- 1—SPI burst completed when the SS input is negated

Also, in BURST_LENGTH definition, it is stated “In the Slave mode, this field takes effect in SPI transfer only when SS_CTL is cleared.”

However, the mode SS_CTL[x] = 1 is not functional in Slave mode. Currently, BURST_LENGTH always defines the burst length.

According to the SPI protocol, negation of SSB always causes completion of the burst. However, due to the above issue, the data is not sampled correctly in RxFIFO when {BURST_LENGTH+1}mod32 is not equal to {actual burst length}mod32.

Therefore, setting the BURST_LENGTH parameter to a value greater than the actual burst does not resolve the issue.

Workaround: Do not use the SS_CTL[x] = 1 option in the Slave mode. The accurate burst length should always be specified using the BURST_LENGTH parameter.

ERR009606: ECSPI: In master mode, burst lengths of 32n+1 will transmit incorrect data

Description: When the ECSPI is configured in master mode and the burst length is configured to a value 32n+1 (where n=0,1, 2,…), the ECSPI will transmit the portions of the first word in the FIFO twice.

For example, if the transmit FIFO is loaded with:

[0] 0x00000001

[1] 0xAAAAAAAA

And the burst length is configured for 33 bits (ECSPIx_CONREG[BURST_LENGTH]=0x020), the ECSPI will transmit the first bit of word [0] followed by the entire word [0], then transmit the data as expected.

The transmitted sequence in this example will be:

[0] 0x00000001

[1] 0x00000001

[2] 0x00000000

[3] 0xAAAAAAAA

Workaround: Do not use burst lengths of 32n+1 (where n=0,1, 2,…).

ERR009165: ECSPI: TXFIFO empty flag glitch can cause the current FIFO transfer to be sent twice

Description: When using DMA to transfer data to the TXFIFO, if the data is written to the TXFIFO during an active ECSPI data exchange, this can cause a glitch in the TXFIFO empty signal, resulting in the TXFIFO read pointer (TXCNT) not updating correctly, which in turn results in the current transfer getting resent a second time.

Workaround: This errata is only seen when the SMC (Start Mode Control) bit is set. A modified SDMA script with TX_THRESHOLD = 0 and using only the XCH (SPI Exchange) bit to initiate transfers prevents this errata from occurring. There is an associated performance impact with this workaround. Testing transfers to a SPI-NOR flash showed approximately a 5% drop in write data rates and a 25% drop in read data rates.

ERR011207: FlexSPI: In rare conditions when FLEXSPI_AHBCR[PREFETCHEN] is set, incorrect data can be returned

Description: When prefetching is enabled (FLEXSPI_AHBCR[PREFETCHEN]) for non-cacheable space, there are conditions where write-read order might not be guaranteed. The problem can occur if data is written and then read back using the AHB interface, while a region containing the data location is in the process of being loaded into the FlexSPI's AHB Rx buffer.

Workaround: There are two workarounds:

- If FlexSPI space is not cached (configured as device or strongly-ordered type in the MPU), then FLEXSPI_AHBRXBUFnCR0[PREFETCHEN] should be cleared.
- If the write is critical and the following read is to the same address, FlexSPI_STS0[SEQIDLE] bit can be checked to make sure the write has completed (SEQIDLE is 1) before issuing the subsequent read.

ERR050044: GPC: GPU power domain on/off operation leads to unexpected behavior

Description: GPU modules include two types of reset signals:

1. Hardware reset for GPUMIX which can be asserted during power-off state.

2. Software reset for GPU2D and GPU3D. The software reset signal cannot be asserted during GPU2D or GPU3D power-off state.

When only the GPU2D or the GPU3D is powered-off and then powered-on, it is possible for the GPU2D or GPU3D to enter into an unknown status due to the missed reset. This may cause unexpected GPU2D or GPU3D interrupt to be received. The typical failure symptom is that system fails to suspend and resume because of the unexpected GPU interrupt.

Workaround: To power the GPU2D and the GPU3D on and off simultaneously, use following the sequence:

1. Assert the entire GPUMIX reset by setting the GPU_RESET bit of the SRC_GPU_RCR register.

2. Power on GPU2D and GPU3D.

3. De-assert the entire GPUMIX reset by clearing the GPU_RESET bit of the SRC_GPU_RCR register.

ERR050522: GPU: GPU driver cannot get expected value from “onComp2” logic, it would fail to initialize correctly

Description: OpenGL ES 2.0 Conformance test suite may fail when the “onComp2” non-resettable flops found in the Fetch Engine (FE) power-up value is 1 (one) instead of 0 (zero). On i.MX8M Mini, the “onComp2” flop has an initial value of 1, and the GPU driver expects a value of 0, the GPU would fail to initialize correctly after power-up. The GPU driver which did not reference the “onComp2” logic during initialization would reset the value to 0 (zero) only after the initial draw. This issue could produce an intermittent failure in the Google Earth test or in the fixed_data_type tests.

Workaround: Software with the workaround prevents exposure of this issue by avoiding cross mapping between streamID and attribute. Additionally, the GPU driver initialization sequence includes a draw as part of the reset sequence. This sets the non-resettable flops (including the onComp2 flop) to 0.

A robust workaround of adding a dummy draw with at least one vertex consisting of 4 32-bit components as part of the reset sequence during the GPU driver initialization sequence. When FSM wiggled once, the problematic flop gets a proper value, and subsequent operations have correct values. This ensures that the non-resettable flops (including the onComp2 flop) are always set to 0, regardless of the exact scenario details.

ERR007805: I2C: When the I2C clock speed is configured for 400 kHz, the SCL low period violates the I2C spec of 1.3 uS min

Description: When the I2C module is programmed to operate at the maximum clock speed of 400 kHz (as defined by the I2C spec), the SCL clock low period violates the I2C spec of 1.3 uS min. The user must reduce the clock speed to obtain the SCL low time to meet the 1.3us I2C minimum required. This behavior means the SoC is not compliant to the I2C spec at 400kHz.

Workaround: To meet the clock low period requirement in fast speed mode, SCL must be configured to 384KHz or less.

ERR050080: IO: Degradation of internal IO pullup/pulldown current capability for IO’s continuously driven in a 3.3V operating mode

Description: There is a degradation of the internal IO pullup/pulldown capability when the IO pads are continuously driven in the opposite logic level, for example, when internal pullup is enabled with external logic driving the pin low, and a 3.3V operating condition which limits the pads pullup/pulldown ability.

All IO pin groups are impacted except for XTAL, DDR, PCI, USB, and MIPI PHY IO’s.

Workaround: Where the IO’s are used for 3.3V operation and the circuit requires pullup or pulldown, use external resistors for the pullup/pulldown and disable the internal pullup/pulldown via software.

ERR050045: IOMUX: Setting ODE control bit of I2C IOs causes malfunction

Description: The I2C module supports open drain. The I2C module drives the open-drain signal of the output data. However, setting the ODE bit in the I2C IOMUXC registers results in malfunctions due to internal logic.

Workaround: Do not set the ODE bit in the I2C IOMUX registers because I2C module already supports open drain.

ERR011437: IOMUX: The read data is always zero when ODE bit of ENET PHY IOs is set

Description: ODE (Open Drain Enable) bit of IOMUXC_SW_PAD_CTL_PAD_ENET_MDIO is required to be set according to Ethernet MDIO spec. However when ODE bit is set, the read data from MDIO pin is always zero due to the following factors:

1. Before reading data from the MDIO pin, the data which is pre-set by software in register ENET_MMFR is sent out. If the last bit of the data is zero, the output data maintains the last bit of previous data which is zero when the direction of pin is changed to input mode for data read。

2. When the ODE bit is set, internal logic always drives low if the output data is zero, even if the direction is changed to input mode.

Workaround: There are 2 workarounds according to application use:

1. If the application uses one Ethernet PHY, disable open-drain mode by configuring IOMUXC_SW_PAD_CTL_PAD_ENET_MDIO bit5 as 0.

2. If the application uses multiple Ethernet PHYs, the software can set the lower 16-bits of the ENET_MMFR register to avoid triggering the issue. This workaround depends on an external PHY to set the lower 16-bits.

ERR011341: IOMUXC: Missing HW force_en for SD_CLK pin of USDHC3

Description: The IOMUXC register, IOMUXC_SW_MUX_CTL_PAD_NAND_WE_B is missing the HW force_en bit which is used to configure the SD_CLK pin of USDHC3.

Workaround: Register IOMUXC_SW_MUX_CTL_PAD_NAND_WE_B bit-4 must be set when NAND_WE_B is used as usdhc3_clk for loopback.

ERR011193: PCIE: EP, PM_PME: L1 Exit Does Not Occur when PME Service Timeout Mechanism Expires

Description: Impacted Configuration(s): Upstream Port configurations:

CC_DEVICE_TYPE =DM and device_type =4’b0000

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 10 “PCIE_CTRL_APPS_PME” of register “SRC_PCIEPHY_RCR” .

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.

ERR050640: ROM: TCM 64-bit alignment issue in ROM for NAND page with bad block marker at unaligned offset

Description: There is an known issue with M4-TCM memory only for parts before ROM patch V5 (Date Code before 1933). When M4-TCM memory is used for boot-up (image loading and execution), and it is configured as normal memory in MMU table, corruption might happen on any unaligned access.The NAND page carries the bad block marker (bbM) and its position in the page data depends on the ECC parameters that the BCH ECC block supports. For some ECC levels, the bbM can offset to unaligned 64-bit. During this offset operation, the unaligned 64-bit accessing in the TCM memory where page data resides might cause hard fault.

Workaround: This ROM issue can be avoided by using the image with below uboot/kernel patch.

Upscale the ECC strength so that the ECC will fall on an aligned boundary, as shown below.

For u-boot:

drivers/mtd/nand/mxs_nand.c

```
@@ 263,6 +263,7 @@ static int mxs_nand_get_ecc_strength(struct mtd_info *mtd) } else
{ ecc_strength = chip>ecc_strength_ds; ecc_strength += ecc_strength & 1;

+ ecc_strength <<= 1;
```

For Linux kernel:

drivers/mtd/nand/gpmi-nand/gpmi-nand.c

```
@@ -294,6 +294,7 @@ static int set_geometry_by_ecc_info(struct gpmi_nand_data *this)

geo->ecc_chunk0_size = chip->ecc_step_ds; ecc_chunkn_size =

geo->chip->ecc_step_ds; ecc_strength = round_up(chip->ecc_strength_ds,

geo->2);

+ geo->ecc_strength <<= 1;
```

ERR050542: SAI: The Bit Count Timestamp Register (TBCTR, RBCTR) may return a live rather than latched Timestamp

Description: A SAI Timestamp Counter instance implements independent 32-bit counters for BCLK and a Timestamp based on the sub-system clock (AUDIO_AHB_CLK_ROOT, typically 400MHz). The current value of the timestamp count is latched on a BCLK edge and the contents of that latch is further latched into the xBCTR register whenever the BCLK count is read (xBCR). However, reading xBCR sometimes results in xBCTR latching the current value of the timestamp count, not the value latched on the most recent BCLK edge. This introduces uncertainty in the timestamp of up to 1 BCLK period.

Workaround: A BCLK period is sampling frequency and format dependent e.g. 142 sub-system clocks for 44.1kHz I2S or 33 sub-system clocks for 48kHz TDM8. These represent 3.5ppm or 825ppb respectively when measuring at 10Hz, compared to the 25ppb design aim. The uncertainty in the timestamp is instantaneous not accumulating and should be considered when designing any PLL or ASRC correction.

ERR050448: SPDIF: SPDIF clock limitation

Description: The SPDIF IP includes a DPLL driven from the subsystem clock, which is used to generate a data strobe to sample the incoming bitstream

When the subsystem clock to SPDIF bitrate ratio is too high, the DPLL might not lock to the correct sampling frequency and phase.

For example: If the DPLL is using a 400MHz subsystem clock, it is able to lock in normal use cases of 44.1kHz and above, but it cannot track jitter reliably.

Workaround: Configure the audio_ahb_clk as 200MHz which was connected to gclkw_t0. It impacts the SDMA which share the audio_ahb_clk, while SDMA performance can be restored by partitioning the workload across both SDMA2 and SDMA3.

ERR051272: TMU: Bit 31 of registers TMU_TSCR/TMU_TRITSR/TMU_TRATSR invalid

Description: Bit 31 of registers TMU_TSCR/TMU_TRITSR/TMU_TRATSR might be set as invalid value when the temperature varies in range.

Workaround: Do not use Bit 31 of registers TMU_TSCR/TMU_TRITSR/TMU_TRATSR. Suggest to read TMU value and use 1 point calibration to justify if the temperature is in range.

NXP Linux BSP does not use those bits since rel_imx_4.14.98_2.0.0_ga.

ERR050101: USB: Endpoint conflict issue in device mode

Description: An endpoint conflict occurs when the USB is working in device mode and an isochronous IN endpoint exists.

When the endpointA IN direction is an isochronous IN endpoint, and the host sends an IN token to endpointA on another device, then the OUT transaction may be missed regardless the OUT endpoint number. Generally, this occurs when the device is connected to the host through a hub and other devices are connected to the same hub.

The affected OUT endpoint can be either control, bulk, isochronous, or an interrupt endpoint.

After the OUT endpoint is primed, if an IN token to the same endpoint number on another device is received, then the OUT endpoint may be unprimed (Cannot be detected by SW), which causes this endpoint to no longer respond to the host OUT token, and thus, no corresponding interrupt occurs.

Workaround: Do not connect to a hub in the case when ISO IN endpoint(s) is used. When the hub(s) must be connected in this scenario, the endpoint number(s) of the ISO IN endpoint(s) should be different from the endpoint number(s) of any type of IN endpoint(s) used in any other device(s) connected to the same host.

ERR050150: WDOG: Watchdog WDOG_RESET_B_DEB reset SoC failed

Description: WDOG_RESET_B_DEB is not connected as a reset source of SRC_GPR registers. So when the boot address in SRC_GPR registers is set as a different value other than the default ROM address, WDOG_RESET_B_DEB cannot make ARM CA53 restart from default boot ROM.

Workaround: Using WDOG_B(GPIO1_IO02) to reset the system power can avoid this issue.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

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

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.