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

Table 1. Errata and Information Summary

Erratum ID	Erratum Title
e3774	AIPS: Unaligned access causes abort on writes to the internal registers
e6939	Core: Interrupted loads to SP can cause erroneous behavior
e9004	Core: ITM can deadlock when global timestamping is enabled
e9005	Core: Store immediate overlapping exception return operation might vector to incorrect interrupt
e6940	Core: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used
e50028	CoreSight: Coresight debug components cannot be auto-detected by debug tool
e9535	ECSPI: Burst completion by SS signal in slave mode is not functional
e9606	ECSPI: In master mode, burst lengths of 32n+1 will transmit incorrect data
e9165	ECSPI: TXFIFO empty flag glitch can cause the current FIFO transfer to be sent twice
e11207	FlexSPI: In rare conditions when FLEXSPI_AHBCR[PREFETCHEN] is set, incorrect data can be returned
e50044	GPC: GPU power domain on/off operation leads to unexpected behavior
e7805	I2C: When the I2C clock speed is configured for 400 kHz, the SCL low period violates the I2C spec of 1.3 uS min
e50080	IO: Degradation of internal IO pullup/pulldown current capability for IO's continuously driven in a 3.3V operating mode

Table continues on the next page...

Table 1. Errata and Information Summary (continued)

Erratum ID	Erratum Title
e50045	IOMUX: Setting ODE control bit of I2C IOs causes malfunction
e11437	IOMUX: The read data is always zero when ODE bit of ENET PHY IOs is set
e11341	IOMUXC: Missing HW force_en for SD_CLK pin of USDHC3
e11193	PCIE: EP, PM_PME: L1 Exit Does Not Occur when PME Service Timeout Mechanism Expires

Table 2. Revision History

Revision	Changes
0, 01/2019	Initial revision

e3774: AIPS: Unaligned access causes abort on writes to the internal registers

Description: Unaligned access to AIPS can be driven high by SAHARA, DAP, and FEC. If they access the AIPS internal registers during an unaligned access, an ABORT occurs.

Workaround: Make only aligned accesses to the AIPS internal registers.

e6939: 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.

e9004: 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.

e9005: 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”);

}

e6940: Core: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used

Description: Arm Errata 709718: 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.

e50028: 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.

e9535: 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.

e9606: 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,…).

e9165: 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.

e11207: 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.

e50044: 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.

e7805: 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.

e50080: 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 certain IO pads are continuously driven in a 3.3V operating condition which limits the pads pull up/down ability. The impedance of these pads changes over time from 20K to upwards of 92K ohm.

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

Workaround: For 3.3V IO operation use external resistors for pull up/ pull down and disable the internal pull up/down via software.

e50045: 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.

e11437: 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.

e11341: 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.

e11193: 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” .

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.