iMX8ULPA2_P40A

Mask Set Errata

Mask Set Errata for Mask P40A

Revision History

This report applies to mask P40A for these products:

- MIMX8UD7DVK08SC
- MIMX8UD5DVK08SC
- MIMX8UD3DVK08SC
- MIMX8US5DVK08SC
- MIMX8US3DVK08SC
- MIMX8UD7DVP08SC
- MIMX8UD7CVP08SC
- MIMX8UD5DVP08SC
- MIMX8UD5CVP08SC
- MIMX8UD3DVP08SC
- MIMX8UD3CVP08SC
- MIMX8US5DVP08SC
- MIMX8US5CVP08SC
- MIMX8US3DVP08SC
- MIMX8US3CVP08SC

Table 1. Revision History

Revision	Date	Significant Changes
3.0	10/2023	The following errata were revised. - ERR050622
2.0	10/2023	Initial Revision

Errata and Information Summary

Table 2. Errata and Information Summary

Erratum ID	Erratum Title
ERR008151	PXP: Rotation Engine alignment and operation combination limitations
ERR008153	PXP: Rotation1 Engine format support limitation
ERR010589	GIC500:855721-C GICD_TYPER.CPUNumber!=0 when the ARE bits are fixed as RAO/WI
ERR050500	Core: Group priority of a Non-secure interrupt might be incorrect when AIRCR.PRIS is set
ERR050501	Core: DFSR.EXTERNAL is not set correctly when waking up from sleep
ERR050502	Core: Execution priority might be wrong for one cycle after AIRCR is changed
ERR050503	Core: Non-secure HardFault exception might preempt when disabled by AIRCR.BFHFNMINS

Table continues on the next page...

Table 2. Errata and Information Summary (continued)

Erratum ID	Erratum Title
ERR050504	Core: Sorting of pending interrupts might be wrong when high latency IRQs are pending
ERR050505	Core: Access permission faults are prioritized over unaligned Device memory faults
ERR050537	FlexSPI: Read timing sequence mismatches with several existing SPI NOR devices in dual, quad, and octal modes
ERR050542	SAI: The Bit Count Timestamp Register (TBCTR, RBCTR) may return a live rather than latched Timestamp
ERR050553	SPDIF: The SPDIF clock is not automatically turned off in low power modes
ERR050559	System Oscillator is enabled after Power On Reset
ERR050561	Debugger de-attachment while SoC is in low power modes with CKMODE > 0 causes CMC0 become irresponsive
ERR050563	Debugger can't access the CA35 registers in low power
ERR050581	LPSPI: Frame transfer starts when host request input is not asserted
ERR050596	CGC LOS recovery generates a glitch on slow clocks
ERR050599	Low Power: Low and high voltage condition does not reset the SoC as expected
ERR050604	[CSI SUBSYSTEM]: Swapped Y output for YUV420 8bit
ERR050616	Power Down/Deep Power Down wakeup can cause FSGPIO to generate spikes
ERR050617	CM33 FPU Exception Interrupt not connected to NVIC
ERR050618	High current when VDD_PTE and VDD_PTF are kept unpowered in Power Down mode
ERR050622	HiFi4 DDR space is limited to 256MB
ERR050630	FlexCAN: High resolution Time Stamp register may not be updated after CAN message reception
ERR050649	AD CGC generates Loss of Sync during AD reset
ERR050650	Wrong AD debug reset connection
ERR050655	AD XBAR stall request is stuck while there is on-going transaction from AD to RTD peripheral and RTD to AD peripheral
ERR051051	Core: A partially completed VLLDM might leave Secure floating-point data unprotected
ERR051225	ISI: Line stored Interrupts are missed by A35
ERR051226	SoC hang while reading MRC7 reg when HiFi4 is disabled
ERR051227	I3C0 bus functional clock requirements not meet on master mode
ERR051269	USB PHY registers not being reset after a warm reset
ERR051421	SAI: Synchronous mode with bypass is not supported
ERR051522	MIPI-DSI DBI cannot support 24bit pixel on 16bit bus
ERR051652	AHB initiator fails to write data to Shared SRAM
ERR052003	CA35x Hot plug AVD domain power domain switch system hang under stress test conditions

Known Errata

ERR008151: PXP: Rotation Engine alignment and operation combination limitations

Description

Rotation Engine Position 0:

When processing 'ps' and 'as' buffers that are unaligned (buffers are not aligned to block boundaries) then a rotation operation that is combined with a flip, decimation, or scaling operation will not execute correctly. Rotation operations must be done in separate passes. These combination operations execute correctly for aligned buffers.

Rotation Engine Position 1:

Unaligned buffer rotation does not execute correctly. Rotation operations combined with flip, scaling, or decimation do not execute correctly. Only simple aligned rotation is supported.

Workaround

Rotation Engine Position 0:

When processing 'ps' and 'as' buffers that are unaligned (buffers are not aligned to block boundaries) then a rotation operation that is combined with a flip, decimation, or scaling operation will not execute correctly. Rotation operations must be done in separate passes. These combination operations execute correctly for aligned buffers.

Rotation Engine Position 1:

Unaligned buffer rotation does not execute correctly. Rotation operations combined with flip, scaling, or decimation do not execute correctly. Only simple aligned rotation is supported.

ERR008153: PXP: Rotation1 Engine format support limitation

Description

Rotation of data in YUV420 format does not work correctly.

Workaround

Do not use rotation of data in YUV420 format

ERR010589: GIC500:855721-C GICD_TYPER.CPUNumber!=0 when the ARE bits are fixed as RAO/WI

Description

GICD_TYPER.CPUNumber field reads incorrectly report the number of configured CPUs (topping out at 7) irrespective of whether ARE=0 modes are supported.

Workaround

Ignore the GICD_TYPE.CPUNumber field as it has no meaning when ARE=1.

ERR050500: Core: Group priority of a Non-secure interrupt might be incorrect when AIRCR.PRIS is set

Description

Cortex-M33 1113997-C:

When the processor is configured with Security extension and AIRCR.PRIS is 1, the Armv8-M architecture requires that the priorities of Non-secure interrupts are modified to ensure that Secure interrupts are prioritized over Non-secure interrupts. The Armv8-M architecture requires that lower priority numbers take precedence over higher priority numbers. Because of this erratum, a Non-secure interrupt with higher priority number might be handled in the wrong order compared to another Non-secure or Secure interrupt.

Workaround

There is no workaround for this erratum.

ERR050501: Core: DFSR.EXTERNAL is not set correctly when waking up from sleep

Description

Cortex-M33 1367266-C:

An external debug event which causes the processor to enter Debug state or the debug monitor should set DFSR.EXTERNAL. It has been found that this field is not set if the event occurs while the processor is asleep.

Workaround

There is no workaround.

ERR050502: Core: Execution priority might be wrong for one cycle after AIRCR is changed

Description

Cortex-M33 1435973-C:

AIRCR is used in the NVIC active tree to calculate the execution priority, which in turn is used to determine fault escalation, exception preemption, and other NVIC-related behaviors. When the active tree is pipelined and there are high latency IRQs active, there might be a glitch in the active tree output for one cycle after AIRCR is changed. The glitch results in NVIC producing wrong execution priority that is neither based on the old AIRCR value nor the new one.

Workaround

There is no workaround for this erratum.

ERR050503: Core: Non-secure HardFault exception might preempt when disabled by AIRCR.BFHFNMINS

Description

Cortex-M33 1453380-C:

When the processor implements the Security Extension and AIRCR.BFHFNMINS is 1, the Non-secure banked version of SHCSR.HARDFAULTPENDED can be set to 1. This Non-secure pended HardFault might not preempt per architecture because it does not have enough priority (that is, the processor is in HardFault handler mode). If AIRCR.BFHFNMINS is subsequently changed to 0 with the Non-secure HardFault still pending, then the architecture requires that the Nonsecure HardFault should

never preempt regardless of execution priority. Because of this erratum, the pended Non-secure HardFault exception preempts when AIRCR.BFHFNMINS is 0 and current execution priority is larger than -1 (Non-secure HardFault having higher priority).

Workaround

There is no workaround for this erratum.

ERR050504: Core: Sorting of pending interrupts might be wrong when high latency IRQs are pending

Description

Cortex-M33 1540599-C:

The NVIC contains a pending tree which sorts all pending and enabled interrupts based on priorities. If DHCSR.C_DEBUGEN and DHCSR.C_MASKINTS are 1, DHCSR.S_SDE is 0 and halting debug is allowed, then Nonsecure PendSV, Non-secure SysTick, and Non-secure IRQs should be masked off and they should not affect the sorting of pending and enabled secure interrupts. If multiple high latency IRQs are pending and enabled with different security targets and priorities, then Non-secure IRQs which should be masked off might cause the pending tree output to be a pending Secure nterrupt without highest priority. This is because of incorrect masking before doing priority comparisons in the tree.

Workaround

There is no workaround for this erratum.

ERR050505: Core: Access permission faults are prioritized over unaligned Device memory faults

Description

Cortex-M33 1080541-C :

A load or store which causes an unaligned access to Device memory will result in an UNALIGNED UsageFault exception. However, if the region is not accessible because of the MPU access permissions (as specified in MPU_RBAR.AP), then the resulting MemManage fault will be prioritized over the UsageFault.

Workaround

There is no workaround.

However, it is expected that no existing software is relying on this behavior since it was permitted in Armv7-M.

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.

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.

ERR050553: SPDIF: The SPDIF clock is not automatically turned off in low power modes

Description

The SPDIF clock is not automatically turned off in low power modes.

Workaround

SPDIF clock needs to be manually turned off before going to low power mode.

To turn off the SPDIF clock, bit 30 (CGC) of the PCC_SPDIF register (address 0x2DA700AC) needs to be cleared.

ERR050559: System Oscillator is enabled after Power On Reset

Description

Bit 0 (SOSCEN) of register System OSC Control Status Register (SOSCCSR) comes enabled after POR, which is not correct. The expected behavior was that the System OSC should come disabled by default.

Workaround

Clear the bit 0 (SOSCEN) of register System OSC Control Status Register (SOSCCSR) to disable the System Oscillator and reduce power consumption.

ERR050561: Debugger de-attachment while SoC is in low power modes with CKMODE > 0 causes CMC0 become irresponsive

Description

CM33 enters in a low power mode after that debugger is attached and CMC0 wakes up, enabling the clocks, the CM33 core will not wake-up in this scenario.

Debugger performs some access, after that, the debugger is de-attached and is expected that SoC enters again in previous configured low power mode but CMC0 doesn't conclude this transition, and master and slave clocks are not gated.

This issue depends on the time that the debugger is de-attached and it impacts low power modes using CKMODE > 0.

Workaround

1) Do not de-attach the debugger while in low power

2) Enter in low power modes with CLKMODE=0

ERR050563: Debugger can't access the CA35 registers in low power

Description

In a low power mode with CKMODE > 0, the debugger can't access the CA35 registers due to its clock being gated.

The CMC1 DBGCTL[SOD] config does not affect this behavior.

Workaround

A workaround is to enter the low power mode with CKMODE = 0, then the CA35 core clock will not be gated and the registers will be accessible.

ERR050581: LPSPI: Frame transfer starts when host request input is not asserted

Description

If LPSPI has Host Request Enabled in master mode, then frame transfer can start even when host request input trigger is not asserted.

The issue occurs when the HREQ input (either pin or trigger) asserts for longer than the previous transfer that was triggered. The HREQ state is internally latched and if the LPSPI returns to idle and the HREQ input is still asserted then HREQ will remain latched. The next data written to the transmit FIFO will therefore start immediately without waiting for next HREQ assertion. If the HREQ input asserts for less than the time it takes for LPSPI to perform the transfer there is no issue.

Workaround

The workaround is to ensure the HREQ negates before the end of the LPSPI transfer that was triggered by HREQ pin assertion.

ERR050596: CGC LOS recovery generates a glitch on slow clocks

Description

After CGC detects an LOS event, if CGC is configured to only reset the system clock dividers (not issuing a reset request to CMC), the procedure executed by CGC in attempt to recover from LOS can possibly cause slow system clocks (exclusive to RTD and AD CGC, cm33_slowclk and xbar_slowclk respectively) to overshoot frequency restrictions for one clock cycle. The logic issuing the reset for the slow clock dividers has no protection against glitching the slowclk signals - resets can be issued and the dividers restarted mid slowclk cycle.

Workaround

Configure the CGC0, CGC1 and CGC2 to generate the domain resets in LOS event, so that all clock dividers are reset while the domain is also being reset.

The CGC0 registers bits expected values are:

CGC0.CLKDIVRST[CM33_RESET_EN]=1

CGC0.CLKDIVRST[CM33_INTERRUPT_EN]=0

CGC0.CLKDIVRST[CM33_RST_DIVIDERS_EN]=0

CGC0.CLKDIVRST[FUSION_RESET_EN]=1

CGC0.CLKDIVRST[FUSION_INTERRUPT_EN]=0

CGC0.CLKDIVRST[FUSION_RST_DIVIDERS_EN]=0

CGC1.CLKDIVRST[NIC_RESET_EN]=1

CGC1.CLKDIVRST[NIC_INTERRUPT_EN]=0

CGC1.CLKDIVRST[NIC_RST_DIVIDERS_EN]=0

CGC2.CLKDIVRST[HIFI_RESET_EN]=1

CGC2.CLKDIVRST[HIFI_INTERRUPT_EN]=0

CGC2.CLKDIVRST[HIFI_RST_DIVIDERS_EN]=0

CGC2.CLKDIVRST[NICLPAV_RESET_EN]=1

CGC2.CLKDIVRST[NICLPAV_INTERRUPT_EN]=0

CGC2.CLKDIVRST[NICLPAV_RST_DIVIDERS_EN]=0

ERR050599: Low Power: Low and high voltage condition does not reset the SoC as expected

Description

When RTD, AD or LPAV HVD or LVD Reset are enabled through PMC MON_INTC register, HVD3RE, HVD2RE, HVD1RE, LVD3RE, LVD2RE, LVD1RE control bits and there is a HVD or LVD event in corresponding power domain, then all domains will be reset. RTD CMC0 SRS registers will have POR and LVD bits asserted.

Workaround

Leave HVD and LVD reset feature disabled and enable the HVD and LVD interrupts. For LVD adjust their levels to a higher value so that the SW handle the LVD condition for the specific domain.

ERR050604: [CSI SUBSYSTEM]: Swapped Y output for YUV420 8bit

Description

With Legacy YUV420 8 bit, video data is received in UYY…/VYY sequences in odd/even lines by CSI Rx Controller.

If First line Input data is aa_aa_01_01_00_00_aa_aa_01_01 (picking initial bytes).

It is in U1Y1Y2U2Y3Y4… sequence.

AS per this, Y and U outputs should be

U -> aa_01_aa_01

Y -> aa_01_00_00_aa_01

In ISI output Buffer, U data is correct but for Y data, Y1 and Y2 bytes are swapped and so on i.e. Y2Y1Y4Y3...

Workaround

This means that SW has to swap these bytes.

ERR050616: Power Down/Deep Power Down wakeup can cause FSGPIO to generate spikes

Description

Conditions:

I. VDD_PTA/B powered OFF in RTD PowerDown or DeepPowerDown(vdd_dig0_sw=vdd_lv=OFF)

II. VDD_PTE/F powered OFF in APD DeepPowerDown(vdd_dig1=vdd_lv=OFF). (Once FSGPIO large current is fixed for vdde=OFF &vdd_lv=ON, then APD PowerDown will be another potential affected mode)

III. In a wakeup event the PowerSys will request PMIC to turn ON the power supplies again, including above VDD_PTx.

IV. PMIC turns ON all power rails simultaneously after a PMIC_STBY_REQ deassertion

V. Once PowerSys detects VDD_DIG0/VDD_DIG1 are at their nominal voltage it deasserts corresponding padrst signals

VI. If padrst is deasserted before VDD_PTx has reached its nominal voltage, then we can have spikes in the IOs outputs.

Workaround

If customer application is sensitive to spikes in affected IOs, then user should configure PMIC to not power OFF the VDD_PTA/B in RTD PowerDown/DeepPowerDown modes and not power OFF VDD_PTE/F in APD PowerDown/DeepPowerDown modes.

ERR050617: CM33 FPU Exception Interrupt not connected to NVIC

Description

The CM33 FPU offers 6 possible flags based on its operation results which are reflected in MCM ISCR register as well. Further MCM has bits to enable a collective interrupt for the same.

But the interrupt is not assigned any interrupt line on NVIC and without this, the FPU exceptions cannot be trapped.

Workaround

Use alternative blocks for performing floating point operations, such as PowerQuad, or the Fusion DSP.

ERR050618: High current when VDD_PTE and VDD_PTF are kept unpowered in Power Down mode

Description

Due to FSGPIO design implementation issue, there is a high current leakage when VDD_PTE and VDD_PTF are kept unpowered in Power Down mode.

Workaround

Keep VDD_PTE and VDD_PTF powered for maximum SoC power savings.

ERR050622: HiFi4 DDR space is limited to 256MB

Description

The HiFi4 peripheral address range (0x20000000 - 0x2FFFFFFF) and HiFi4 DDR range (0x30000000 - 0x3FFFFFFF) are in the same 512M cache attribute region.

If the cache is enabled for DDR region, this will cause the peripheral addresses to fail, since caches should be disabled for peripheral accesses.

Workaround

512MB DDR is mapped in two separate regions (256MB + 256MB) and they are separate 256MB region partition.

The first 256MB partition can be configured with Cache-Enabled that includes DDR/FlexSPI and the second region partition as "No cache" for Peripherals and SRAM.

This will restrict HiFi4 to 256MB DDR.

ERR050630: FlexCAN: High resolution Time Stamp register may not be updated after CAN message reception

Description

When the Time Stamp Capture Point field (TSTAMPCAP) of the Control 2 register (CTRL2) is different from 00b, the hardware may not update the High Resolution Time Stamp (HR_TIME_STAMP) register as described below.

1. A frame is received with a particular message identifier.

2. The frame is transferred to the corresponding MB.

3. CPU locks the MB so that until it is read, another incoming message cannot overwrite it.

4. Meanwhile another frame for the same MB is received. As the MB is locked, the frame is parked in temporary buffer (this buffer can store up to 2 received frames). Corresponding timestamp is also stored in temporary buffer.

5. Now a third frame starts being received for legacy or enhanced FIFO (whichever is applicable).

6. When the frame from temporary buffer is being processing by the hardware to FIFO, CPU unlocks the MB having read the data from there.

7. Thus the second frame which was parked in temporary buffer starts to get transferred to unlocked MB.

8. The High Resolution Time Stamp Register (HR_TIME_STAMP) associated with the MB must also receive the corresponding timestamp value stored in the temporary buffer. But due to incorrect control value, this timestamp doesn't reach the .

High Resolution Time Stamp Register.

9. This issue is valid only if the MB is locked for more than 20 CAN bit times for classical CAN frames or for more than16 nominal CAN bit plus 6 data phase CAN bit times for CAN FD frames.

Workaround

One of the three workaround below can be implemented.

1. Disable all other interrupts in the SoC when an MB has to be serviced so that it can be read before 20 CAN bit times for classical CAN frames and 16 nominal CAN bit plus 6 data phase CAN bit times for CAN FD frames.

2. Use only FIFO or only Message Buffers for receiving CAN messages (FIFO is used for gateway kind of application to transmit information in the same order as it is received).

3. Use the 16-bit time stamp field (TIME_STAMP) of the message buffer instead of the High Resolution Time Stamp (HR_TIME_STAMP) register setting the Message Buffer Time Stamp Base (MBTSBASE) field of the Control 2 (CTRL2) register as 01b to capture the 16 least significant bits of the high resolution timer or 10b to capture the 16 most significant bits of the high resolution timer.

The problem occurs only when both FIFO and Message buffers are configured for receiving CAN frames.

ERR050649: AD CGC generates Loss of Sync during AD reset

Description

During an AD reset all AD clock sources are reset but AD system clock dividers aren't, which can bring AD CGC to detect a Loss of Sync event during reset (it's time-sensitive), so AD CMC captures two reset requests instead of one.

Workaround

Mask the generation of a loss of sync interrupt during a reset sequence.

ERR050650: Wrong AD debug reset connection

Description

According to ARM documentation, it is expected that the debug reset is controlled by the External debug interface and not the CMC1.

In this case, the AD pin reset will also reset the debug logic related to AD avoiding the debug of CA35 reset.

Workaround

Use the MU reset hold feature as a workaround to re-configure the debug components that gets reset by the L2 cold reset:

Set the MU reset hold.

Trigger the AD reset.

The debugger should wait until the AD reset-exit.

Configure the CA35_0 CTI to trigger the entry in debug mode (Halt).

Write 0xC5ACCE55 at CTI_LAR (0x00420FB0) register.

Set bit0 at CTICONTROL (0x00420000) register.

Set bit0 at CTIOUTEN0 (0x004200A0) register.

Set bit0 at CTIAPPPULSE (0x0042001C) register.

Clear the MU reset hold.

Poll the EDPRSR (0x00410314) register until bit4 (HALTED) equals to 1.

Release the CA35_0 CTI, writing 0x1 at CTIINTACK (0x00420010) register.

The debugger can re-configure the CA35_0 internal registers and perform the configurations needed before the Halt release.

OBS: These addresses are to use with CA35 APB-AP (APSEL = 0x1).

ERR050655: AD XBAR stall request is stuck while there is on-going transaction from AD to RTD peripheral and RTD to AD peripheral

Description

Based on the fact that the RTD has AXBS0 and AXBS1, and AD_AXBS access->RTD peripheral (e.g. FLEXSPI1) and RTD (e.g. Sentinal/uPower) access->AD memory (e.g. SRAM1) are both going through AXBS1 M7 port, when AD AXBS receives the halt request (no RTD_AXBS halt), and if there is on-going traffic from AD (e.g. DMA1/uSDHC0) accessing the RTD peripherals (e.g. FlexSPI1), and at the same time, there is traffic from RTD AXBS0 masters (e.g. uPower or Sentinel) accessing the AD peripherals (e.g. SRAM0), these two traffic are both routed via the AXBS1 M7 port, deadlock can occur.

Workaround

SW needs to ensure there is no traffic from XBAR_AD masters that can reach XBAR_RTD (via XBAR_AD.S5) before XBAR_AD stall is requested.

ERR051051: Core: A partially completed VLLDM might leave Secure floating-point data unprotected

Description

Arm errata 2219175

Affects: Cortex-M33

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r0p4, r1p0. Open.

The VLLDM instruction allows Secure software to restore a floating-point context from memory. Due to this erratum, if this instruction is interrupted or it faults before it completes, then Secure data might be left unprotected in the floating point register file, including the FPSCR.

Configurations affected:

This erratum affects all configurations of the Cortex-M33 processor configured with the Armv8-M Security Extension and the Floating-point Extension.

Conditions:

This erratum occurs when all the following conditions are met:

• There is no active floating-point context, (CONTROL.FPCA==0)

• Secure lazy floating-point state preservation is not active, (FPCCR_S.LSPACT==0)

• The floating-point registers are treated as Secure (FPCCR_S.TS==1)

• Secure floating-point state needs to be restored, (CONTROL_S.SFPA == 1)

• Non-secure state is permitted to access to the floating-point registers, (NSACR.CP10 == 1)

• A VLLDM instruction has loaded at least one register from memory and does not complete due to an interrupt or fault

Implications:

If the floating-point registers contain Secure data, a VLSTM instruction is usually executed before calling a Non-secure function to protect the Secure data. This might cause the data to be transferred to memory (either directly by the VLSTM or indirectly by the triggering of a subsequent lazy state preservation operation). If the data has been transferred to memory, it is restored using VLLDM on return to Secure state. If the VLLDM is interrupted or it faults before it completes and enters a Non-secure handler, the partial register state which has been loaded will be accessible to Non-secure state.

Workaround

To avoid this erratum, software can ensure a floating-point context is active before executing the VLLDM instruction by performing the following sequence:

• Read CONTROL_S.SFPA

• If CONTROL_S.SFPA==1 then execute an instruction which has no functional effect apart from causing context creation (such as VMOV S0, S0)

ERR051225: ISI: Line stored Interrupts are missed by A35

Description

Line stored interrupts of ISI are missed by A35 core and all interrupts are not getting serviced at higher bitrates.

Workaround

It is not recommended to use ISI Line Stored Interrupt at higher bitrates and all interrupts are not guaranteed to be serviced by core. Instead use frame done interrupt for each frame captured from sensor.

ERR051226: SoC hang while reading MRC7 reg when HiFi4 is disabled

Description

Access to following IPs hangs when HIFI4_DISABLE fuse is blown.

- MRC7

- MRC9

The reason of this hang is because clocks to above IPs are gated when HIFI4_DISABLE fuse bit is set.

Workaround

The HIFI4_DISABLE fuse should not be blown, if MRC7 or MRC9 is accessed.

In case HIFI4 is not required in a particular use-case, HIFI4 can be configured to lowest frequency possible to save power.

If a particular use-case requires this HIFI4_DISABLE fuse to be blown (for example - aggressive low power requirements), then MRC7 and MRC9 access should be avoided.

ERR051227: I3C0 bus functional clock requirements not meet on master mode

Description

I3C0 bus 0 (RTD) does not meet functional clock requirements for I3C master mode, clock timings are lower and cannot produce 12MHz or 12.5MHz as max frequency on the I3C0 bus.

Workaround

Use I3C0 bus on legacy i2c master mode. Under I2C modes clock speed has no impact as can achieve Fm, Fm+ for I2C.

ERR051269: USB PHY registers not being reset after a warm reset

Description

USB PHY registers are not properly reset after a warm reset

Workaround

The following USB PHY registers must be written by SW to restore the reset value after a warm reset:

USB0:

Reg: ctrl Addr: 0x29910030 Data: 0xc000_0000

Reg: pwd Addr: 0x29910000 Data: 0x001e_1c00

Reg: debug0 Addr: 0x29910050 Data: 0x7f18_0000

Reg: pll_sic Addr: 0x299100a0 Data: 0x00d1_2000

USB1:

Reg: ctrl Addr: 0x29930030 Data: 0xc000_0000

Reg: pwd Addr: 0x29930000 Data: 0x001e_1c00

Reg: debug0 Addr: 0x29930050 Data: 0x07f18_0000

Reg: pll_sic Addr: 0x299300a0 Data: 0x00d1_2000

ERR051421: SAI: Synchronous mode with bypass is not supported

Description

The SAI does not receive or transmit when:

Scenario 1. The transmitter is configured for synchronous mode (TCR2[SYNC] = 0b1), in the Transmit Configuration 2 register, and the receiver is in bypass (RCR2[BYP]=0b1), in the Receiver Configuration 2 register, then there will not be a bit clock as it is the source of the BCLK.

Scenario 2. The receiver is configured for synchronous mode (RCR2[SYNC] = 0b1) in the Receiver Configuration 2 register and the transmitter is in bypass (TCR2[BYP]=0b1), in the Transmit Configuration 2 register, then there will not be a bit clock as it is the source of the BCLK.

Workaround

If scenario 1, then set the TCR2[BCI] = 0b1, in the Transmit Configuration 2 register.

If scenario 2, then set the RCR2[BCI] = 0b1, in the Receiver Configuration 2 register.

ERR051522: MIPI-DSI DBI cannot support 24bit pixel on 16bit bus

Description

RGB888 24bits pixel format output from MIPI-DSI DBI interfaces not supported. MIPI-DSI DBI interface only support 16bits bus.

DCNano can output the correct DBI 24bits option on the DBI bus to MIPI-DSI. But MIPI-DSI cannot adjust the pixel layout and pack to DSI packet correctly. This causes incorrect pixel data in the DSI packet, the display panel cannot read the correct pixel data lead to wrong display content.

Workaround

From DCNano RGB data from DBI 16bit data and swap between to adjacent pixel to format data as expected by MIPI-DSI DBI interface.

ERR051652: AHB initiator fails to write data to Shared SRAM

Description

The SSRAM arbiter module manages the read/write access to a continuously mapped 768KB system memory split into 8 partitions of 32 to 256 KB; There are 7 concurrent paths to access the system memory.

When an AHB initiator (initiator A) writes to the Shared SRAM, the Shared SRAM controller would discard the write when all of the following conditions are met. In this scenario, the write data cannot be written to the memory.

1. Back-to-back AHB write transfers without idle cycles on the AHB bus

2. The write addresses in the back-to-back write transfers crossing two memory partitions

3. The second memory partition that initiator A is trying to write to is parking to another initiator (initiator B).

Below is explanation of the 3rd condition: example, use Cortex-M33 as Initiator A and eDMA as initiator B.

When a memory partition is accessed by initiator eDMA, it is parking to the initiator. When initiator Cortex-M33 issues an access (read or write) to the memory partition, the Shared SRAM controller transfers the ownership of the partition from initiator eDMA to initiator Cortex-M33.

Example scenario:

• Cortex-M33 write data back-to-back (e.g. memset operation) to say partition 2 (ownership of partition 2 assigned to Cortex-M33) – This is the condition #1.

• eDMA read SSRAM, say partition 3 (ownership of partition3 is assigned to eDMA)

• DMA completion interrupt to Cortex-M33 (currently Cortex-M33 is writing data to partition 2, memset operation)

• Cortex-M33 hardware auto push value of registers to stack in say partition 3 - This is condition #2 and condition #3, results in write failure to SSRAM.

Workaround

Make sure that all AHB initiator does not access SSRAM crossing memory partitions, this can be achieved by not sharing the memory partitions with other initiators and restricting AHB initiator accesses its own memory partitions. This invalidates the condition #2 (write addresses in the back-to-back write transfers crossing two memory partitions) to workaround the issue.

ERR052003: CA35x Hot plug AVD domain power domain switch system hang under stress test conditions

Description

In case an APD Power Switch ON sequence like the A35_x hotplug is followed by Powering up an AVD Domain Power Switch, system might hang under stress test conditions.

Scenario: CPU hot-plug test when running CSI stress test. APD NIC frequency needs to be slow down before offline CPU is powered ON once CPU is online APD NIC frequency original frequency can be restored.

Workaround

Before Switching ON the APD Domain Switches, the APD NIC Frequency must be reduced to achieve a clock ratio of at least 1:2 between the APD_NIC AD_XBAR_BUSCLK and RTD CLK . Once the APD Power switching is complete the APD_NIC frequency can be changed back to the use-case (original) frequency.

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.