Document identifier: I.MX93_0P87f
Rev. 0, 4/2023

I.MX93_0P87f

Mask Set Errata

Mask Set Errata for Mask 0P87f

Revision History

This report applies to mask 0P87f for these products:

- PIMX9352xxxxMAA
- PIMX9322xxxxMAA

Table 1. Revision History

Revision	Date	Significant Changes
0	4/2023	Initial Revision

Errata and Information Summary

Table 2. Errata and Information Summary

Erratum ID	Erratum Title
ERR051390	CCM: CCGR doesn't wait for stop ack from the peripherals in low power sequence.
ERR051051	Core: A partially completed VLLDM might leave Secure floating-point data unprotected
ERR050505	Core: Access permission faults are prioritized over unaligned Device memory faults
ERR050504	Core: Sorting of pending interrupts might be wrong when high latency IRQs are pending
ERR050839	Cortex-A55: Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation
ERR050838	Cortex-A55: Update of DBM or AP bits without break-before-make might result in incorrect hardware dirty bit update
ERR051220	DDRC: DDRC Register Access Not Allowed In Half-Speed Mode
ERR051388	DDRC: Exiting DDR retention mode causes data corruption
ERR051252	DDRC: Memory controller may not operate properly when per-bank refresh is enabled
ERR051301	DDRC: Memory controller Performance Degraded During Write Traffic
ERR051554	DDRC: The DDR Controller idle status is not accurate when using PLL bypass mode with data rate ≤ 533MT/s.
ERR051219	DDRC: The DDRC does not automatically apply derated timing values to some timing parameters
ERR051389	eDMA4: eDMA4 will not work well when two requests are asserted at the same time.
ERR051241	eDMA4: Channel preemption feature can operate incorrectly
ERR051336	eDMA4: Swap feature with 8-bit swap size and 16-bit transfer size does not work
ERR051325	eDMA4: The feature of cancelling the remaining data transfer is not working effectively
ERR051337	eDMA4: Unaligned transfer support does not work for transfer crossing 4KB
ERR051327	eDMA4:Scatter-gather feature does not work when NBYTES is not a multiple of 8 bytes
ERR051326	eDMA4:TCDn_CSR[ESDA] and TCDn_CSR[EEOP] are not functional

Table continues on the next page...

Table 2. Errata and Information Summary (continued)

Erratum ID	Erratum Title
ERR051029	ENET_QOS: Gate Control List Switching is Incorrect for Intermediate Cycles When CTR is Less Than GCL Execution Time
ERR051642	FlexCAN: Low-power wakeup does not work due to GPC issue
ERR011377	FlexSPI: DLL lock status bit not accurate due to timing issue
ERR051612	Fuse: Limitations for accessing fuse-disabled features
ERR051186	I3C: Extended data will be lost unexpectedly when I3C is in Slave Mode with EXTDATA set
ERR051120	I3C: The Not Acknowledge Error bit in the Master Errors and Warnings register (MERRWARN[NACK]) is not set when slave does not acknowledge High Data Rate - Double Data Rate (HDR-DDR) read.
ERR051392	ISI: Early vsync issue triggered all the time when streaming data from MIPI camera
ERR009156	LDB: OpenLDI 18-bit SPWG mode does not work correctly
ERR051608	LPSPI: PRESCALE bits in TCR Register has limitation
ERR051605	LPUART: Transmit Complete does not set if TX FIFO is flushed when CTS negated
ERR008244	PXP: pxp_compress step1 FIFO full may lead to data error
ERR008151	PXP: Rotation Engine alignment and operation combination limitations
ERR008153	PXP: Rotation1 Engine format support limitation
ERR051421	SAI: Synchronous mode with bypass is not supported
ERR051302	WDOG: WDOG_ANY does not function as expected
ERR051625	xCache: CM33 WFI/WFE can make SOC hang when xCache is enabled

Known Errata

ERR051390: CCM: CCGR doesn't wait for stop ack from the peripherals in low power sequence.

Description

In the low power sequence, once CCGR receive the request of LPCG step from GPC to gate off IP clocks, the CCGR will provide lpcg_done to GPC without waiting for IP's stop ack. Then, GPC will execute next low power step, such as shutoff PLLs, which may result in gating off the peripheral IP clock without finishing handshake between ccm and IP.

Workaround

SW need guarantee these peripheral IP (I2C1~8/CAN1/eDMA1/CAN2/ENET1/eDMA2/FlexSPI.) in idle status before system enter low power mode, which can be guaranteed in the linux mechanism.

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)

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.

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.

ERR050839: Cortex-A55: Speculative AT instruction using out-of-context translation regime could cause subsequent request to generate an incorrect translation

Description

Arm errata 1530923

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r0p1, r1p0, and r2p0. Open.

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

Configurations Affected:

This erratum affects all configurations.

Conditions:

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

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

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

Implications:

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

Workaround

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

ERR050838: Cortex-A55: Update of DBM or AP bits without break-before-make might result in incorrect hardware dirty bit update

Description

Arm errata 1024718

Affects: Cortex-A55

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r0p1, r1p0, and r2p0. Open.

If the hardware dirty bit update is enabled, and the DBM or AP bits in the translation table descriptor are updated by software without using break-before-make, then it is possible for hardware to incorrectly update the AP bits based on the old value of those bits.

Configurations Affected:

This erratum affects all configurations of the Cortex-A55 processor.

Conditions:

1. The hardware dirty bit update is enabled for stage1 (TCR_ELx.HA and TCR_ELx.HD are both set) or stage2 (VTCR_EL2.HA and VTCR_EL2.HD are both set).

2. A store instruction is executed, which causes a hardware dirty bit update for the translation table descriptor which has DBM=1 and no write permission because of either AP[2]=1 or S2AP[1]=0.

3. At the same time as the store, the OS or hypervisor writes to the same translation table descriptor to update the AP, S2AP, or DBM bits, without using a break-before-make procedure.

4. The new translation table descriptor is a valid translation, but does not require the dirty bit update for the store because either the DBM bit is now clear, or the AP/S2AP bits would still cause a permission fault to occur even after a dirty bit update.

Implications:

The permission checking of the store is performed on the old version of the translation table descriptor, while the AP/S2AP bit is updated in the new version of the translation table descriptor. This could lead to an inconsistent situation where the store is executed but the OS or hypervisor is not expecting the store to have permission to execute.

Workaround

The OS or hypervisor can use a break-before-make procedure if it needs to update the DBM or AP/S2AP bits.

Alternatively, it can use software management of the dirty bit update.

ERR051220: DDRC: DDRC Register Access Not Allowed In Half-Speed Mode

Description

After issuing a Hardware Fast Frequency Change (HWFFC) request, the DDRC operates in half-speed mode. While operating in this mode, if a DDRC register read or write is requested, the internal state counters may get corrupted. In addition, some of the fields of TIMING_CFG_12/13/14 are not used during half-speed operation.

Workaround

If a DDRC register access is required while operating at half-frequency after an HWFFC switch, first switch to full-speed via HWFFC before accessing the register.

ERR051388: DDRC: Exiting DDR retention mode causes data corruption

Description

While exiting the DDR retention mode, the NICMIX fails to send a handshake signal to DDRMIX. This causes a data misalignment between the NICMIX and DDRMIX. As a result, after exiting DDR retention mode the misaligned data causes data corruption and a system failure.

Workaround

While exiting the DDR retention mode, reset both DDRMIX and NICMIX.

ERR051252: DDRC: Memory controller may not operate properly when per-bank refresh is enabled

Description

When the DDRC is in per-bank refresh mode, it may violate the tRFCpb timing by issuing a per-bank refresh to the same bank or an all-bank refresh after a prior per-bank refresh. The memory controller guarantees that TIMING_CFG_9[REFTOREF_PB] (programmed to tpbR2pbR) is met between any two per-bank refreshes. However, it does not guarantee the TIMING_CFG_9[REFREC_PB] (tRFCpb) when issuing a per-bank refresh to the same bank or when issuing an all-bank refresh after a per-bank refresh. Per bank refresh is an optional feature.

Workaround

While using per-bank refresh (TIMING_CFG_9[REFREC_PB] != 0x0), program both TIMING_CFG_9[REFTOREF_PB] and TIMING_CFG_9[REFREC_PB] to meet the maximum of (tRFCpb, tpbR2pbR). tRFCpb is expected to be the maximum of those two timing specs.

Another workaround is to disable per-bank refresh by programming TIMING_CFG_9 to 0x0.

ERR051301: DDRC: Memory controller Performance Degraded During Write Traffic

Description

During processing of certain write transactions, the write performance is degraded.

Workaround

To prevent the write performance degradation, set bit 7 (value of 0x80) to register offset 0xf04 within the DDRC register space.

Note: The "*_timing.c" file generated by DDR tool implements the workaround.

ERR051554: DDRC: The DDR Controller idle status is not accurate when using PLL bypass mode with data rate ≤ 533MT/s.

Description

The DDR Controller idle status flag is not accurate when using the PLL bypass mode with data rates ≤ 533MT/s. As a result when using the Software Fast Frequency Change(SWFFC) routine to switch from the PLL bypass mode with data rates ≤ 533MT/s, instead of polling for the DDRC idle state, a timeout should be used to make sure that the DDR Controller has finished all transactions and can enter self-refresh mode.

In addition, when in PLL bypass mode with data rates ≤ 533MT/s, the automatic clock gating feature cannot be enabled

Workaround

If the Software Fast Frequency Change(SWFFC) routine is used to switch from the PLL bypass mode, a delay timeout is required to ensure the DDR controller is idle.

Before resetting the DDRC;

3 micro second delay; //this delay to ensure DDRC is in Idle state.

then Reset the DDRC (DDR_SDRAM_CFG_3[31]= 1);

ERR051219: DDRC: The DDRC does not automatically apply derated timing values to some timing parameters

Description

While using the dynamic refresh rate feature (via DDR_SDRAM_CFG_3[DYN_REF_RATE_EN], the DDRC automatically adjusts the refresh rate based on the temperature of the DRAM. This works correctly. When the LPDDR4/4X MR4 provides a refresh rate of the value 3'b110 to the controller, the DDRC should use the values in TIMING_CFG_2[DERATE_VAL] to derate the following timing parameter configurations:

- {TIMING_CFG_3[EXT_PRETOACT], TIMING_CFG_1[PRETOACT]}
- {TIMING_CFG_3[EXT_ACTTOPRE], TIMING_CFG_1[ACTTOPRE]}
- {TIMING_CFG_3[EXT_ACTTORW], TIMING_CFG_1[ACTTORW]}
- {TIMING_CFG_3[EXT_ACTTOACT], TIMING_CFG_1[ACTTOACT]}
- TIMING_CFG_8[PRE_ALL_REC]

However, the derating for these specs are not applied correctly within the DDRC. Note that this erratum does not affect the dynamic refresh interval updates.

Workaround

1. If the application is specified to operate at a temperature less than 85 °C, you can enable or disable DDR_SDRAM_CFG_3[DYN_REF_RATE_EN]. In this case, no timing parameters require derating and if DDR_SDRAM_CFG_3[DYN_REF_RATE_EN] is enabled, the refresh rate will automatically adjust per feedback from the LPDDR4/4X device.

2. If the application is specified to operate at a temperature higher than 85 °C, apply the derating value as a fixed delay for all temperatures and also enable DDR_SDRAM_CFG_3[DYN_REF_RATE_EN] to ensure that the refresh rate is automatically adjusted based on the temperature of the DRAM.

While using the workaround #2, apply the derated timing values to the following register configurations when programming the DDRC:

- {TIMING_CFG_3[EXT_PRETOACT], TIMING_CFG_1[PRETOACT]}
- {TIMING_CFG_3[EXT_ACTTOPRE], TIMING_CFG_1[ACTTOPRE]}
- {TIMING_CFG_3[EXT_ACTTORW], TIMING_CFG_1[ACTTORW]}
- {TIMING_CFG_3[EXT_ACTTOACT], TIMING_CFG_1[ACTTOACT]}
- TIMING_CFG_8[PRE_ALL_REC]

Note that the derating for these is typically 1.875 ns, but the DRAM vendor datasheet should be referenced.

Note: Verify with DRAM vendor about the high temperature, which is 85 °C

Note: The "*_timing.c" file generated by DDR tool implements the workaround.

ERR051389: eDMA4: eDMA4 will not work well when two requests are asserted at the same time.

Description

when eDMA4 request[x] is asserted and not completed, if eDMA4 request[y] is also asserted and request[x] has the highest priority, then at the end of request[x] service, request[x] will get serviced again and eDMA4 will go with an extra iteration of minor loop, instead of switching to serve request[y].

Workaround

1 using CPU to move data, instead of eDMA4.

2 Monitors the Hardware Request Status Register “MP_HRS” for pending IPD requests. Only issue a new request if the HRS is all 0's. Basically ensure one-hot requests.

ERR051241: eDMA4: Channel preemption feature can operate incorrectly

Description

Channel preemption can cause incorrect behavior under the following conditions:

• The preempted channel has CHx_PRI[ECP]=1

• The preempting channel has a higher priority than the preempted channel based on the setting of CHx_PRI[APL]

• The preempting channel is activated while the preempted channel has an active transfer on the AXI bus

Workaround

Do not use the preemption feature. For all channels, the TCD register CHn_PRI[31:30] must have at least one of the two settings:

1. Keep CHn_PRI[ECP] = 1'b0 - The channel cannot be suspended by a higher priority channel's service request (default).

2. Set CHn_PRI[DPA] = 1'b1 - The channel can be temporarily suspended by the service request of a higher priority channel.

ERR051336: eDMA4: Swap feature with 8-bit swap size and 16-bit transfer size does not work

Description

The DMA SWAP function that allows software to select data portions to be swapped between the read data to the write data is not working in the following cases:

1. CHn_CSR[15:12]=0001 and SSIZE=001

2. CHn_CSR[15:12]=1001 and DSIZE=001

Workaround

Do not use SWAP function for the described cases.

ERR051325: eDMA4: The feature of cancelling the remaining data transfer is not working effectively

Description

eDMA4 does not cancel the current bus activity. Bus transactions will continue normally and complete the transfer size requested by the NBYTES parameter.

Workaround

Do not use the cancel feature.

ERR051337: eDMA4: Unaligned transfer support does not work for transfer crossing 4KB

Description

For TCD source or destination addresses that are unaligned with the transfer sizes and if the NBYTES setting exceeds 4 KB, it will not work.

Workaround

Make sure the burst is less than 8 bytes.

ERR051327: eDMA4:Scatter-gather feature does not work when NBYTES is not a multiple of 8 bytes

Description

If the current TCD that will initiate the copy has a total transfer size (TCDn_NBYTES_MLOFFYES[NBYTES]) which is not a multiple of 8 bytes, data will not be copied correctly.

Workaround

For cases where the data is 8-byte aligned, use DMA and set NBYTES=8.

For cases where the data is not aligned, use a CPU core to conduct the transfer instead.

ERR051326: eDMA4:TCDn_CSR[ESDA] and TCDn_CSR[EEOP] are not functional

Description

If Store Destination Address is enabled (ESDA = 1) and End of Packet Processing is enabled (EEOP = 1), the address stored will be invalid if the EOP signal is triggered. It will not correspond correctly to the last address transferred when EOP is signalled.

Workaround

TCDn_CSR[EEOP] must be disabled.

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

Description

Impacted Configurations:

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

Parameter:

DWC_EQOS_AV_EST == 1

Versions Affected: 5.00a and later

Defect Summary:

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

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

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

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

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

Impacted Use Cases:

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

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

Workaround

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

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

ERR051642: FlexCAN: Low-power wakeup does not work due to GPC issue

Description

Due to a bug in the GPC module, Flexcan is disabled before low-power handshake. The Flexcan is stopped before entering stop mode, which results in failure to wakeup by Flexcan working in stop mode.

Workaround

FlexCAN as low-power wake up source does not work.

ERR011377: FlexSPI: DLL lock status bit not accurate due to timing issue

Description

After configuring DLL and the lock status bit is set, still may get wrong data if immediately read/write from FLEXSPI based external flash due to timing issue

Workaround

Adding a delay time (equal or more than 512 FlexSPI root clock cycle) after the DLL lock status is set.

ERR051612: Fuse: Limitations for accessing fuse-disabled features

Description

The NPU, USB2, PXP, CSI, DSI and LVDS fuse disable features are erroneous:

• Disabling the NPU and PXP fuses causes the core to hang.

• Disabling the CSI and DSI fuses causes the read register to return all 0x00000000 (write not 0x00000000).

• Disabling the USB2 and LVDS (LVDS register is in mediamix GPR) fuses causes the register to be still accessible.

Workaround

Not to access IP which is fuse disabled or use TRDC SW workaround to block register access for disabled IP.

ERR051186: I3C: Extended data will be lost unexpectedly when I3C is in Slave Mode with EXTDATA set

Description

When I3C in slave mode, async mode 0 (TIMECTRL bits in SSTATUS register) and EXTDATA (EXTDATA bit in SCTRL register) are both set, data transmission appears to end after the async mode 0 timestamp, but it will continue on and confuse the I3C controller.

T bit goes to 0 but it should stay 1/High-Z after the 3rd item (the timestamp) and then continues emitting the extended data.

The I3C controller receives data 0 from T bit and thinks data transmission is done. So, extended data will be lost unexpectedly.

Workaround

Do not use EXTDATA (data past the MDB) or enable async mode 0 at the same time.

Since async mode 0 is only intended for MEMS sensors but not normal IBIs, so the EXTDATA is more commonly used.

ERR051120: I3C: The Not Acknowledge Error bit in the Master Errors and Warnings register (MERRWARN[NACK]) is not set when slave does not acknowledge High Data Rate - Double Data Rate (HDR-DDR) read.

Description

I3C: The Master Errors and Warnings register (MERRWARN) is used to debug I3C/I2C errors and warnings in Master mode. The MERRWARN[NACK] bit does not set when slave does not accept read while HDR-DDR mode is used. This bit is set to 1 in Single Data Rate (SDR) mode when slave does not acknowledge.

Workaround

If a slave does not accept HDR-DDR read and master side is not able to debug, the slave availability/readiness can be checked by sending SDR read request. The MERRWARN[NACK] will reflect the slave response.

ERR051392: ISI: Early vsync issue triggered all the time when streaming data from MIPI camera

Description

The frame timing signals, and vertical or horizontal synchronism in the Image Pixel Interface (IPI) of MIPI-CSI controller, are generated based on the camera sensor packets such as Frame Start, Line Start, and blanking packets. If MIPI camera output does not contain line synchronization packets, ISI will always asserts CHNL_STS[EARLY_VSYNC] interrupt bit due to mismatched IPI timing sequence. And the last line from camera sensor may also be missing in some condition, for example when ISI Color Space Conversion is enabled.

Workaround

To avoid trigger this issue, it requires camera sensor has line synchronization packets or adding blanking lines before Frame End packet. Some camera sensors don't support this feature.

ERR009156: LDB: OpenLDI 18-bit SPWG mode does not work correctly

Description

When the device is configured to work in 18 bit SPWG Mode, the LVDS Display Bridge (LDB) incorrectly selects the least significant 18 bits of the output bus rather than the 6 most significant bits from each color component.

Workaround

Configure the Module in 24 bit JEIDA mode and do not use the last line (TX3) from the OpenLDI module. By doing this the module behaves exactly as required in SPWG 18-bit mode.

ERR051608: LPSPI: PRESCALE bits in TCR Register has limitation

Description

LPSPI TCR[PRESCALE] can only be configured to be 0 or 1, other values are not valid and will cause LPSPI to not work.

All 8 LPSPI instances are the same.

Workaround

Driver can use CCR1 register to divide SCK, whose biggest div rate is 512. Software workaround integrated in Linux BSP codebase starting in release imx_5.15

ERR051605: LPUART: Transmit Complete does not set if TX FIFO is flushed when CTS negated

Description

If TX FIFO is flushed by software when CTS is enabled (MODIR[TXCTSE] field is set) and its value is negated and the transmitter is idle waiting for CTS to assert, but Transmit Complete (STAT[TC]) bit is not set.

Workaround

Do not use TC bit to check the TX status.

ERR008244: PXP: pxp_compress step1 FIFO full may lead to data error

Description

Due to a pxp compression bug, when pxp_compress step1 FIFO is full there is a Read source data channel error.

Workaround

Set OT=1 and check the flag output from: pxp_compress module, flag fifo_full, and error_prone.

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

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.

ERR051302: WDOG: WDOG_ANY does not function as expected

Description

The WDOG_ANY toggles immediately once the interrupt vector fetch. But NXP PMIC has the debounce time which is sufficient for SW to serve the IRQs.

Workaround

The PMIC needs to guarantee debounce time for the WDOG ANY PAD, typically 120us.

ERR051625: xCache: CM33 WFI/WFE can make SOC hang when xCache is enabled

Description

CM33 WFI/WFE could hang the system when xCache is on. The issue happens when CM33 execute WFI/WFE to stop CM33 clock while the xCache at same time has ongoing access to the downstream memory like OCRAM, FlexSPI etc.

Workaround

Disable xCache before entering WFI/WFE for CM33 and enable it again after wakeup. xCache may be required to flush the modified lines before disabling.

Legal information

Definitions

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

Disclaimers

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Trademarks

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

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

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

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

Date of release: 4/2023
Document identifier: I.MX93_0P87f