i.MX RT1170ACE

Chip Errata

Rev. 1.6 — 24 October 2024 Errata

1 i.MX RT1170A Chip Errata

1.1 Revision History

This document details all known silicon errata for the:

i.MX RT1170A

Table 1. Revision History

Revision	Release Date	Significant Changes
1.6	10/2024	The following errata were added. • ERR052412 • ERR052365 • ERR052351
1.5	7/2024	The following errata were added. • ERR052152 • ERR052147 The following errata were revised. • ERR011573
1.4	3/2023	The following errata were added. • ERR051307 • ERR051248 • ERR051472 • ERR051315 • ERR051322 The following errata were revised. • ERR051091
1.3	10/2021	The following errata were added. • ERR051122 • ERR051091 The following errata were revised. • ERR006941
1.2	4/2021	The following errata were added. • ERR050790 The following errata were revised. • ERR011573
1.1	4/2021	The following errata were removed. • ERR011575 The following errata were added. • ERR003777 • ERR006941 • ERR050708

Table 1. Revision History...*continued*

Revision	Release Date	Significant Changes
1.0	4/2021	Initial Revision

1.2 Summary of Arm Software Developers Errata Notice

This table summarizes all known Arm errata and lists the corresponding silicon revision level to which they apply. A 'Yes' entry indicates the erratum applies to a particular revision level, and a 'No' entry means it does not apply.

Table 2. Summary of Arm Software Developers Errata Notice

SDEN ID	Category	NXP Errata Number	Summary of Erratum	Silicon Rev.
				1.47
Arm Cortex-M4F				
776924	CatB	ERR006940	Core: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used	Yes
Arm Cortex-M7				
1013783	CatB	ERR011573	Cortex-M7: Speculative accesses might be performed to memory unmapped in MPU.	Yes
Arm Cortex-M4,Cortex-M4F				
752770	CatB	ERR006939	Core: Interrupted loads to SP can cause erroneous behavior	Yes
838869	CatB	ERR009005	Core: Store immediate overlapping exception return operation might vector to incorrect interrupt	Yes
771919	CatB	ERR006941	Core: Asynchronous sampling of SWDIOTMS might cause incorrect operation of SerialWire/JTAG Debug Port	Yes

1.3 Summary of Silicon Errata

This table summarizes all known errata and lists the corresponding silicon revision level to which they apply. A 'Yes' entry indicates the erratum applies to a particular revision level, and a 'No' entry means it does not apply.

Table 3. Summary of Silicon Errata

Erratum ID	Erratum Title
ADC	
ERR051091	ADC: IO leakage current is observed when pins configured for ADC function
ADC_ETC	
ERR052412	ADC_ETC: SW_TRIG stuck high
CAAM	
ERR050579	CAAM: When Key Form is set to “form #4: p, q, dp, dq, cr, rrp, rrq” in the format of the PROTINFO field for the RSA Decrypt Protocol, CAAM module can generate an error in some conditions
CM4 core	
ERR006939	Core: Interrupted loads to SP can cause erroneous behavior

Table 3. Summary of Silicon Errata*...continued*

Erratum ID	Erratum Title
ERR006940	Core: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used
ERR006941	Core: Asynchronous sampling of SWDIOTMS might cause incorrect operation of SerialWire/JTAG Debug Port
ERR009005	Core: Store immediate overlapping exception return operation might vector to incorrect interrupt
CM7 core	
ERR011573	Cortex-M7: Speculative accesses might be performed to memory unmapped in MPU.
Debug	
ERR050708	Debug: CoreSight components are not linked to CoreSight ROM table
ENET	
ERR050539	ENET: ENET_QOS doesn't support RMII 10Mbps mode
ERR052152	ENET: The hardware acceleration feature for ICMP checksum generation and checking does not work for IPv6.
FLEXIO	
ERR050458	FLEXIO: Shifter Status/Error flag not generated correctly in Logic Mode
FSGPIO	
ERR052351	FSGPIO: A parametric shift over time is observed on FSGPIO output driver when it is powered above 1.98V
FlexSPI	
ERR011377	FlexSPI: DLL lock status bit not accurate due to timing issue
GPC	
ERR051307	GPC: STANDBY entry issue on single core parts
GPIO	
ERR050643	GPIO: During initial power-up, a brief pull-up pulse could occur on the port pins
GPT	
ERR003777	GPT: Possibility of additional pulse on src_clk when switching between clock sources
LCDIFv2	
ERR051315	LCDIFv2: Alpha blending happen on non-overlapped region
LPSPI	
ERR050606	LPSPI: TCR value does not get resampled when polling the register
ERR050607	LPSPI: TCR[FRAMSZ] can be ignored when TCR[TXMSK]=1b1
ERR051472	LPSPI: Disabling and enabling LPSPI would cause SR[REF] to assert
OCRAM	
ERR050634	OCRAM: No ECC interrupt for CM7
QDC	
ERR050659	QDC: A possible speed measurement issue when CTRL3[PMEN]=1

Table 3. Summary of Silicon Errata*...continued*

Erratum ID	Erratum Title
ERR052365	QDC: The position counter is not reset at the same place by index signal when the input quadrature signal direction changes
ROM	
ERR050790	ROM: ROM does not support SEMC_DCCR register configuration from DCD
ERR051322	ROM: Missing WDOG_B timeout assertion in ROM
SAI	
ERR050144	SAI: Setting FCONT=1 when TMR>0 may not function correctly
SNVS	
ERR051122	SNVS: Some fuse trim values are lost in SNVS mode
SOC	
ERR050396	SOC: Sparse write to CM7 TCM causes data corruption
USB	
ERR052147	USB: ISO schedule issue in FS Host mode
VIDEO_MUX	
ERR051248	VIDEO_MUX: Raw data and YUV422(10 bit) from MIPI CSI not supported

2 Known Errata

2.1 ERR052152: ENET: The hardware acceleration feature for ICMP checksum generation and checking does not work for IPv6.

Description

For ENET/ENET1G, the hardware acceleration feature for ICMP checksum generation and checking does not work for IPv6. It does work for IPv4.

Workaround

Use software for IPv6 ICMP checksum generation and checking.

2.2 ERR050539: ENET: ENET_QOS doesn't support RMII 10Mbps mode

Description

For ENET_QOS, 10M mode on RMII doesn't work.

RMII 100M mode, MII 10/100M mode and RGMII 10/100/1000M mode are not impacted.

Workaround

Select MII or RGMII if 10M mode is a must.

2.3 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.

2.4 ERR052412: ADC_ETC: SW_TRIG stuck high

Description

A write of 1 to TRIGx_CTRL[SW_TRIG] while TRIGx_CTRL[SW_TRIG] is high may get the ADC_ETC TRIGx_CTRL[SW_TRIG] register bit to be stuck high.

Workaround

Ensure TRIGx_CTRL[SW_TRIG] is clear before attempting a new TRIGx_CTRL[SW_TRIG] set.

2.5 ERR050579: CAAM: When Key Form is set to “form #4: p, q, dp, dq, cr, rrp, rrq” in the format of the PROTINFO field for the RSA Decrypt Protocol, CAAM module can generate an error in some conditions

Description

In the format of the PROTINFO field for the RSA Decrypt Protocol, the Key Form can be set 4 options such as “form #1: n, d “, “form #2: p, q, d”, “form #3: p, q, dp, dq, c” and “form #4: p, q, dp, dq, cr, rrp, rrq”. Using the form #4 option can generate errors if the size of p <= 128 Bytes and 128 Bytes < the size of q <= 256 Bytes.

Workaround

To avoid the error generated by CAAM module, please select key form #3 instead of key form #4 in the format of the PROTINFO field for the RSA Decrypt Protocol.

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

Description

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

Affects: Cortex-M4F

Fault Type: Programmer Category B

Fault Status: Present in: r0p0, r0p1 Open.

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

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

Workaround

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

There are two workarounds:

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

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

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

Description

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

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

Affects: Cortex-M4, Cortex-M4F

Fault Type: Programmer Category B

Fault Status: Present in: r0p0, r0p1 Open.

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

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

1) LDR SP,[Rn],#imm

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

3) LDR SP,[Rn,#imm]

4) LDR SP,[Rn]

5) LDR SP,[Rn,Rm]

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

1) LDR SP,[Rn],#imm

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

Conditions:

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

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

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

Implications:

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

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

Workaround

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

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

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

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

Description

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

Affects: Cortex-M4, Cortex-M4F

Fault Type: Programmer Category B Rare

Fault Status: Present in: r0p0, r0p1 Open.

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

Configurations Affected

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

Workaround

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

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

ARMCC:

...

schedule_barrier();

asm{DSB};

schedule_barrier();

}

GCC:

...

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

}

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

Description

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

SerialWire/JTAG Debug Port

Status

Affects: Cortex-M4, Cortex-M4F

Fault Type: Implementation Category B Rare

Fault Status: Present in: r0p0, r0p1 Open.

Description

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

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

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

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

Conditions

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

- Serial Wire mode is being used
- The physical implementation does not prevent glitch generation.

Implications

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

Workaround

Check the following points after implementation:

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

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

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

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

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

- 5'b01000 (SWDP_SLEPARKH)
- 5'b01010 (SWDP_SLETRNH2)

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

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

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

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

- JTAGcurr

2.10 ERR050458: FLEXIO: Shifter Status/Error flag not generated correctly in Logic Mode

Description

Some shifters will not generate status or error flags correctly when configured for logic mode (SHIFTCTLn[SMOD] = 0b111). Shifters 0, 1, 2, and 3 behave correctly. All other shifters are affected.

Workaround

In logic mode, if the Status/Error flags are required, use shifter 0, 1, 2, or 3. If the Status/Error flag is not required, then any shifter could be used in logic mode.

2.11 ERR052365: QDC: The position counter is not reset at the same place by index signal when the input quadrature signal direction changes

Description

When CTRL[XIP]=1, CTRL2[REVMOD]=0 and CTRL2[MOD]=0, the position counter is reset by index signal and the revolution counter is also incremented/decremented by the index signal. In this case, the position counter reset and revolution counter change always occur at the rising edge of the index signal disregard of the actual rotation direction of the external encoder. Since the index pulse usually has a width that lasts couple of position counts, the places where the encoder index rising edge occurs in forward and reverse directions have a difference of couple of position counts as well.

Workaround

Users should not use index signal to reset position counter or drive revolution counter by keeping CTRL[XIP]=0, CTRL2[REVMOD]=1 and CTRL2[MOD]=1, so that the position counter is reset by modulo counting(register INIT and MOD defines the modulus boundary) and the revolution counter is driven by modulo counting roll-over/ under events.

2.12 ERR050659: QDC: A possible speed measurement issue when CTRL3[PMEN]=1

Description

When CTRL3[PMEN]=1, and reading POSD occurs simultaneously with any edge of phase A or phase B signal, the captured position difference value in POSDH register may not match the time period captured in POSDPERH register, which causes inaccuracy in speed measurement.

Workaround

No workaround.

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

Description

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

Workaround

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

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

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

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

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

Change clock source CLKSRC in GPTCR

Clear Status register—Write 003F in GPTSR

Set ENMOD in GPTCR

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

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

Description

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

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

Workaround

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

2.15 ERR051472: LPSPI: Disabling and enabling LPSPI would cause SR[REF] to assert

Description

The SR[REF] asserts if software disables the LPSPI module after receiving some data. It then enables the LPSPI again without performing a software reset.

Workaround

When software disables the LPSPI and enables it again, either:

a) Clear SR[REF] flag before starting any data transfer or

b) Software reset the LPSPI before enabling

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

Description

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

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

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

Workaround

There are two workarounds:

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

OR

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

2.17 ERR051307: GPC: STANDBY entry issue on single core parts

Description

On single core devices the CM4's GPC controller is still logically present, but the registers within it cannot be written. The GPC mask registers default to all interrupts and debug enabled (unmasked). Because the CM4

GPC cannot be reprogrammed to mask sources, any pending interrupts or debug will block entry to STANDBY mode.

Workaround

Before attempting to enter STANDBY mode, mask all interrupts that you do not want to block STANDBY entry at the module. Also make sure there is not an active debug session as it will also block STANDBY entry.

2.18 ERR011573: Cortex-M7: Speculative accesses might be performed to memory unmapped in MPU.

Description

Arm errata 1013783-B

Fault Type: Programmer Cat B

Cortex-M7 can perform speculative memory accesses to Normal memory for various reasons. All other types of memory should never be subject to speculative accesses.

The memory attributes for a given address are defined by the settings of the MPU when it is enabled. Regions that are not mapped in the MPU do not have any explicit attributes and should not be subject to any speculative accesses.

Because of this erratum, Cortex-M7 can incorrectly perform speculative accesses to such unmapped regions.

Conditions:

To trigger this erratum, the data cache must be enabled and the MPU must be enabled with the default memory map disabled. That is:

• CCR.DC = 1; data cache is enabled.

• MPU_CTRL.ENABLE = 1; MPU is enabled.

• If MPU_CTRL.PRIVDEFNA = 1, then this erratum cannot occur from privileged mode.

• If MPU_CTRL.HFNMIENA = 1, then this erratum cannot occur from the NMI or HF handlers or exception handlers when FAULTMASK = 1.

In these situations, a PLD instruction targeting an unmapped region might result in an incorrect speculative access. The PLD instruction itself could be speculative because of branch prediction. Even a literal data value that corresponds to a PLD encoding could theoretically cause this issue. This makes it difficult to scan code to check if these conditions apply.

Therefore, Arm recommends that any software with the MPU and data cache configured as mentioned in the conditions above uses the workaround below.

Implications:

Processor execution is not directly affected by this erratum. The data returned from the speculative access is never used and if the access is inferred by the program, then an abort will be taken as required.

The only implications of this erratum are the access itself which should not have been performed. This might have an impact on memory regions with side-effects on reads or on memory which never returns a response on the bus.

Workaround

Instead of leaving memory unmapped, software should use MPU region 0 to cover all unmapped memory and make this region execute-never and inaccessible. That is, MPU_RASR0 should be programmed with:

• MPU_RASR0.ENABLE = 1; MPU region 0 enable.

• MPU_RASR0.SIZE = b11111; MPU region 0 size = 2^32 bytes to cover entire memory.

• MPU_RASR0.SRD = b00000000; All sub-regions enabled.

• MPU_RASR0.XN = 1; Execute-never to prevent instruction fetch.

• MPU_RASR0.AP = b000; No read or write access for any privilege level.

• MPU_RASR0.TEX = b000; Attributes = Strongly-ordered.

• MPU_RASR0.C = b0; Attributes = Strongly-ordered.

• MPU_RASR0.B = b0; Attributes = Strongly-ordered.

Note that the MPU supports addressing hitting in multiple regions with the highest numbered region taking priority.

Therefore, use of MPU region 0 in this way does not affect the existing organization and use of MPU regions.

2.19 ERR050634: OCRAM: No ECC interrupt for CM7

Description

OCRAM space from 0x2020_000 to 0x2023_FFFF cannot trigger ECC interrupt for CM7 core.

Workaround

This memory space cannot be used as ECC memory for CM7 core, while it can be used as normal memory.

2.20 ERR051122: SNVS: Some fuse trim values are lost in SNVS mode

Description

Some trim values will be lost when entering in SNVS mode. This may lead to missed and/or false triggers of voltage, clock, and temperature tampers. GPIO_SNVS_xx pins will automatically switch to tamper function in SNVS mode, even for the devices that do not support tamper and have configured the pins for their GPIO13 functions.

Workaround

1. Disable voltage, clock, and temperature tampers before entering in SNVS mode. The tampers can be re-enabled after exiting SNVS.

2. Do not use GPIO_SNVS_xx pins as GPIO functions in SNVS mode.

2.21 ERR050643: GPIO: During initial power-up, a brief pull-up pulse could occur on the port pins

Description

By default (reset state), the GPIO pins are in the high-Z state and typically stay high-Z until the application code changes its state. The internal pull-up and internal pull-down resistors are disabled by default.

During power up, the internal pull-up resistor on the GPIO_AD, GPIO_LPSR, and DISP_B2 pins may enable during the early part of the IO ramp up, resulting in a brief pull-up current pulse on some port pins that drops to zero before the VDD supplies reach the minimum operating voltage.

Workaround

A pulldown resistor (~10K) can be added to the GPIO pin(s) to minimize the peak voltage where the application is sensitive to potential pulses.

2.22 ERR051248: VIDEO_MUX: Raw data and YUV422(10 bit) from MIPI CSI not supported

Description

Due to an issue in the Video Mux Controller (VIDEO_MUX), raw data and YUV422 (10 bit) formats to the MIPI_CSI2 block are not supported.

Workaround

When using the MIPI_CSI2 interface, use a camera or video decoder with one of the supported data formats. Please refer to the MIPI_CSI2 chapter in the RM for the list of supported data formats.

Use the parallel CSI instead. The parallel CSI interface is not affected and supports more data formats than the MIPI_CSI2. Please refer to the CSI chapter in the RM for the list of supported data formats.

2.23 ERR052351: FSGPIO: A parametric shift over time is observed on FSGPIO output driver when it is powered above 1.98V

Description

When FSGPIO is powered above 1.98V, a parametric shift over time is observed on FSGPIO output driver, where the output low drive current (IOL) is degraded, leading to a longer fall time and output low voltage level (VOL) is increased. Analog and input functionality is not impacted.

For i.MX RT1170, GPIO_AD/GPIO_LPSR/GPIO_DISP_B2 banks are FSGPIO.

For new or updated designs, use 1.8V mode (1.71-1.98V) for the FSGPIO power supply. For the legacy designs, refer to Technote TN00188 for IOL and fall time degradation information when operating above 1.98V. If it is determined that IO does not meet the mission profile requirement of the end application, implement the workaround.

Workaround

The power supply pins, NVCC_GPIO, NVCC_LPSR and NVCC_DISP2 should be connected to 1.8V mode (1.71-1.98V).

And configure the related IO bank supply voltage range selection bitfield in GPR register as continuous voltage range or low voltage range through the below description. Both continuous voltage range and low voltage range are allowed. Recommend to use low voltage range for better delay performance and power consumption if the power supply is 1.8V mode (1.71-1.98V).

- For GPIO_AD bank, set GPIO_AD0/1_LOW_RANGE and GPIO_AD0/1_HIGH_RANGE in GPR69 register.

• Continuous voltage range: GPIO_AD0/1_LOW_RANGE = 0, GPIO_AD0/1_HIGH_RANGE = 0

• Low voltage range: GPIO_AD0/1_LOW_RANGE = 1, GPIO_AD0/1_HIGH_RANGE = 0

- For GPIO_DISP_B2 bank, set GPIO_DISP_B2_LOW_RANGE and GPIO_DISP_B2_HIGH_RANGE in GPR69 register.

• Continuous voltage range: GPIO_DISP_B2_LOW_RANGE = 0, GPIO_DISP_B2_HIGH_RANGE = 0

• Low voltage range: GPIO_DISP_B2_LOW_RANGE = 1, GPIO_DISP_B2_HIGH_RANGE = 0

- For GPIO_LPSR bank, set GPIO_LPSR_LOW_RANGE and GPIO_LPSR_HIGH_RANGE in GPR34 register.

• Continuous voltage range: GPIO_LPSR_LOW_RANGE = 0, GPIO_LPSR_HIGH_RANGE = 0

• Low voltage range: GPIO_LPSR_LOW_RANGE = 1, GPIO_LPSR_HIGH_RANGE = 0

A fix will be implemented on the next i.MX RT1170 revision.

2.24 ERR051091: ADC: IO leakage current is observed when pins configured for ADC function

Description

When using a 3.3 V pad supply and input voltage greater than 1.25 V and less than 1.8 V, or when using a 1.8 V pad supply and input voltage greater than 0.92 V, a leakage current of up to 3.5 uA may be observed flowing out from the I/O pads. When a pin is used for an ADC function, the leakage current affects the ADC conversion accuracy.

Workaround

There are two workarounds:

1. Keep the ADC input resistance as low as possible, this minimizes the voltage deviation generated by the leakage current (3.5 µA). For example, with a 12-bit ADC and 1.8 V reference voltage, an external resistance of less than 125 ohms keep the voltage deviation (due to leakage current) less than 1 LSB.

2. If the ADC input resistance cannot be limited:

- If using 3.3 V as the pad supply:

a) Configure the pads for high-range mode operation and guarantee the voltage on a given ADC pin stays below 1.25 V. If an ADC pin goes above 1.25 V at any point, then the pin should be forced to 0 temporarily (configure as an output driving a low) before attempting a new ADC measurement.

b) Configure the pads for high-range mode operation and guarantee the voltage on a given ADC pin stays above 1.8 V.

- If using 1.8 V as the pad supply:

a) Configure the pads for continuous range mode operation (default) and guarantee the voltage on all ADC pins stays below 0.92 V. If an ADC pin goes above 0.92 V at any point, then the pin should be forced to 0 temporarily (configure as an output driving a low) before attempting a new ADC measurement.

b) Configure the pads for high-range mode operation and guarantee the voltage on all ADC pins stays below 1.25 V. If an ADC pin goes above 1.25 V at any point, then the pin should be forced to 0 temporarily (configure as an output driving a low) before attempting a new ADC measurement. Note: in this mode, any digital pins within the bank will operate as an open drain.

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

Description

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

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

Workaround

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

2.26 ERR050790: ROM: ROM does not support SEMC_DCCR register configuration from DCD

Description

The ROM does not allow for writes to the SEMC_DCCR register. When the SEMC_DCCR is not written, the maximum SDRAM frequency at boot time is 166MHz.

Workaround

Use the DCD to configure SDRAM for a maximum operating frequency of 166MHz. If faster operation is required, then application code can be used to write the SEMC_DCCR value and then increase the SEMC_CLK_ROOT frequency (up to 200MHz). Please refer to the SDK for examples showing booting with a 166MHz SDRAM clock, and then increasing the clock to 200MHz.

2.27 ERR051322: ROM: Missing WDOG_B timeout assertion in ROM

Description

The ROM option to assert the WDOG_B pin on a watchdog timeout is not supported. Blowing the WDOG_B_PIN_EN fuse has no effect. The WDOG timeout option can still be used to reset the device, but the ROM will not configure the WDOG_B pin to assert on timeout.

Workaround

There are two different workarounds for the issue:

- Set WDOG1_WCR[WDT] bit from the application code to enable the WDOG_B pin.

- Configure WDOG2 from the application with the WDOG2_WCR[WDT] bit set.

2.28 ERR052147: USB: ISO schedule issue in FS Host mode

Description

When working in FS Host mode, for ISO communication, if the first ISO package is equal or greater than about 238 bytes, then the second ISO transaction could not be scheduled in the same 1 ms frame.

For example, a typical application case affected is audio bi-directional communication with 48 KHz sample rate, 24 bit per sample and 2 channels for both direction. In this case, we need one ISO IN 288 bytes (>238 bytes) + one ISO OUT 288 bytes (>238 bytes) in one 1 ms frame. But actually, only one ISO OUT 288 bytes or only one ISO IN 288 bytes is available in one 1 ms frame due to the IP limitation.

Workaround

No workaround.

2.29 ERR050708: Debug: CoreSight components are not linked to CoreSight ROM table

Description

There are CoreSight components integrated at 0xE004_4000 to 0xE004_8FFF including SWO, TSGEN, TPIU, ATB funnel and CTI. There is no CoreSight ROM integrated, so they are not discoverable through a ROM table. These components can still be used by accessing the corresponding addresses shown in RM.

Workaround

These components can still be used by accessing the corresponding addresses shown in RM. It's a compliance issue that they are not discoverable through a ROM table.

2.30 ERR050396: SOC: Sparse write to CM7 TCM causes data corruption

Description

For some bus masters, writing access to CM7 TCM is handled by a NIC301 block which does not support sparse write conversion.

It results in a data corruption when sparse writing to CM7 TCM happens.

The affected bus masters includes CAAM, ENET_1G, ENET_QOS, GC355, LCDIFv2, PXP, uSDHC, ENET, and USB.

Workaround

For CAAM, ENET_1G, ENET_QOS, GC355, LCDIFv2, and PXP: Do not write TCM, use OCRAM or external RAM instead.

For uSDHC: If CM7 TCM is the destination for writing, IOMUXC_GPR_GPR28[AWCACHE_USDHC] should be cleared. If IOMUXC_GPR_GPR28[AWCACHE_USDHC] is set, write buffers must be placed in OCRAM or external RAM.

For ENET: If CM7 TCM is the destination for writing, IOMUXC_GPR_GPR28[CACHE_ENET] should be cleared. If IOMUXC_GPR_GPR28[CACHE_ENET] is set, write buffers must be placed in OCRAM or external RAM.

For USB: If CM7 TCM is the destination for writing, IOMUXC_GPR_GPR28[CACHE_USB] should be cleared. If IOMUXC_GPR_GPR28[CACHE_USB] is set, write buffers must be placed in OCRAM or external RAM.

2.31 ERR051315: LCDIFv2: Alpha blending happen on non-overlapped region

Description

The expected behavior of alpha blending is to take place only on the region of the two layers where the pixels overlap. But in actual fact, all pixels of the layers are involved in the alpha blending operation.

Workaround

Use PXP or GPU instead.

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.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://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.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Quick reference data — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

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.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

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.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell 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.

Contents

For more information, please visit: https://www.nxp.com

Date of release: 24 October 2024
Document identifier: IMXRT1170ACE