

Chip Errata for the i.MX RT1050

This document details the silicon errata known at the time of publication for the i.MX RT1050 crossover processors.

[Table 1](#) provides a revision history for this document.

Table 1. Document Revision History

Rev. Number	Date	Substantive Changes
Rev 1.1	08/2018	<ul style="list-style-type: none">• Added following errata:<ul style="list-style-type: none">– ERR006032– ERR009527– ERR009595– ERR011207– ERR011377
Rev. 1	03/2018	<ul style="list-style-type: none">• Added following errata:<ul style="list-style-type: none">– ERR006223– ERR011225– ERR011262• Updated following errata:<ul style="list-style-type: none">– ERR011164– ERR011092– ERR011093– ERR011091– ERR011138– ERR011111– ERR011165– ERR011110– ERR011119– ERR011120
Rev. 0	10/2017	<ul style="list-style-type: none">• Initial version

Figure 1 provides a cross-reference to match the revision code to the revision level marked on the device.

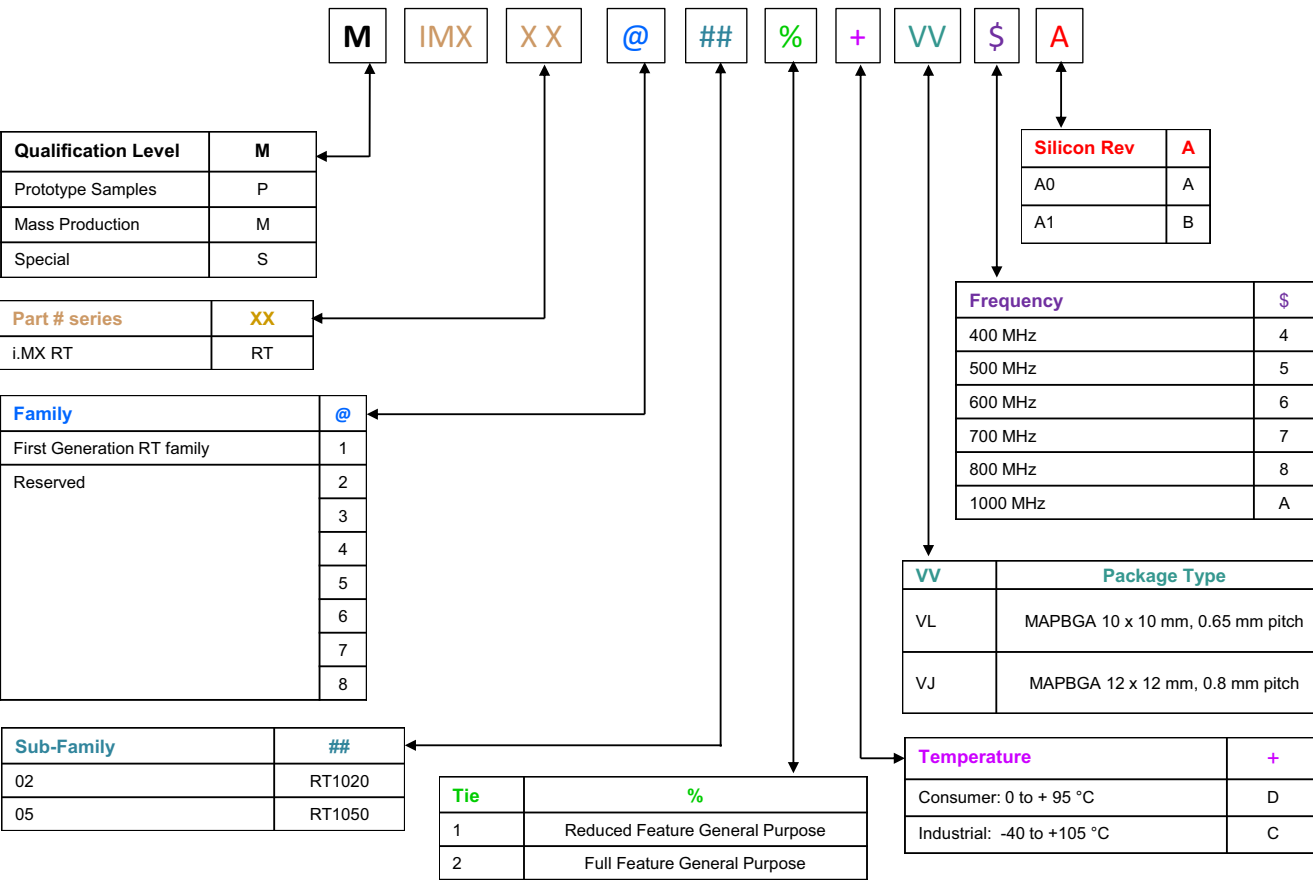


Figure 1. Revision Level to Part Marking Cross-Reference

For details on the Arm® configuration used on this chip (including Arm module revisions), please see the “Platform configuration” section of the “Arm Cortex®-M7 Platform” chapter of the *i.MX RT1050 Series Reference Manual (IMXRT1050RM)*.

Table 2 summarizes errata on the i.MX RT1050.

Table 2. Summary of Silicon Errata

Errata	Name	Solution	Page
ADC			
ERR011164	ADC: ADC_ETC fails to clear the ADC_ETC request signals automatically after receiving DMA ack	A0 Erratum, fixed in A1 silicon	6
CCM			
ERR006223	CCM: Failure to resume from Wait/Stop mode with power gating	No fix scheduled	7
ERR007265	CCM: When improper low-power sequence is used, the SoC enters low power mode before the Arm core executes WFI	No fix scheduled	8
DCDC			
ERR011092	DCDC: Some power up sequence may result in DCDC startup failure	A0 Erratum, fixed in A1 silicon	9
ERR011093	DCDC: Unexpected DCDC reset occurs on some chips	A0 Erratum, fixed in A1 silicon	10
FlexCAN			
ERR005829	FlexCAN: FlexCAN does not transmit a message that is enabled to be transmitted in a specific moment during the arbitration process	No fix scheduled	11
ERR006032	FlexCAN: A frame with wrong ID or payload is transmitted into the CAN bus when the Message Buffer under transmission is either aborted or deactivated while the CAN bus is in the Bus Idle state	No fix scheduled	13
ERR009527	FlexCAN: The transmission abort mechanism may not work properly	No fix scheduled	15
ERR009595	FlexCAN: Corrupt frame possible if the Freeze Mode or the Low-Power Mode are entered during a Bus-Off state	No fix scheduled	16
FlexSPI			
ERR011207	FlexSPI: In rare condition when FLEXSPI_AHBCCR[PREFETCHEN] is set, incorrect data can be returned	A0 Erratum, fixed in A1 silicon	18
ERR011377	FlexSPI: FlexSPI DLL lock status bit not accurate due to timing issue	No fix scheduled	19
IO			
ERR011091	IO: High POR current if NVCC_EMC/NVCC_GPIO/NVCC_SD0/NVCC_SD1 is powered ahead of VDD_SOC_IN	A0 Erratum, fixed in A1 silicon	20
LCDIF			
ERR011138	LCDIF: LUT consecutive programming may fail in case two writes are close	A0 Erratum, fixed in A1 silicon	21
LPSP			
ERR011097	LPSP: Command word does not load properly when TXMSK = 1	No fix scheduled	22

Table 2. Summary of Silicon Errata (continued)

Errata	Name	Solution	Page
RTWDOG			
ERR011111	RTWDOG: Chip stuck in reset when RTWDOG low byte test mode is enabled	A0 Erratum, fixed in A1 silicon	23
SAI			
ERR011096	SAI: The internal bit clock cannot be generated when BCI = 1	No fix scheduled	24
SEMC			
ERR011225	SEMC: CPU AXI writes to SEMC NAND memory may cause incorrect data programmed into NAND memory	No fix scheduled	25
SNVS			
ERR011165	SNVS: Invalid ECC check failure	A0 Erratum, fixed in A1 silicon	26
System Boot			
ERR011262	System Boot: ROM cannot boot from SEMC NAND with default fuse setting	Fixed in A1 silicon	27
ERR011110	System Boot: SEMC NOR boot cannot support the signed image authentication under HAB closed mode	A0 Erratum, fixed in A1 silicon	28
ERR011119	System Boot: FlexSPI NOR encrypted XIP boot fails after system reset if the FAC region number is less than 2	A0 Erratum, fixed in A1 silicon	29
ERR011120	System Boot: FlexSPI NOR encrypted XIP boot fails after system reset if the IOMUXC_GPR18 to IOMUXC_GPR21 are locked	A0 Erratum, fixed in A1 silicon	30
USB			
ERR006281	USB: Incorrect DP/DN state when only VBUS is applied	No fix scheduled	31
ERR010661	USB: VBUS leakage occurs if USBOTG1 VBUS is on and USBOTG2 VBUS transitions from on to off	No fix scheduled	32

ERR011164 ADC: ADC_ETC fails to clear the ADC_ETC request signals automatically after receiving DMA ack

Description:

If enable ADC_ETC to trigger DMA transfer, the DMA transfer data send ack out. It does not clear the request signals automatically and continue to trigger DMA.

Projected Impact:

This issue can lead to DMA failure when working with ADC_ETC.

Workarounds:

Configuring two DMA channels for ADC_ETC data transfer. The first DMA channel with low priority triggered by ADC_ETC request is to transfer ADC_ETC data. The second DMA channel with high priority links to the first channel is to clear request of ADC_ETC by writing DMA_CTRL register. Both channel's priority need to be higher than any channel used by other peripherals. This solution is result in DMA to transfer ADC_ETC data twice for one request signal and application need handle the redundant data properly.

Proposed Solution:

Fixed in A1 silicon

Software Status:

Software workaround is not in SDK.

ERR006223 CCM: Failure to resume from Wait/Stop mode with power gating**Description:**

When entering Wait/Stop mode with power gating of the Arm core(s), if an interrupt arrives during the power-down sequence, the system could enter an unexpected state and fail to resume.

Projected Impact:

Device might fail to resume from low-power state.

Workarounds:

Use REG_BYPASS_COUNTER (RBC) to hold off interrupts when the PGC unit is in the middle of the power-down sequence. The counter needs to be set/cleared only when there are no interrupts pending. The counter needs to be enabled as close to the WFI (Wait For Interrupt) state as possible. The PREG_BYPASS_COUNT value is equal or greater than 2.

Proposed Solution:

No fix scheduled

Software Status:

Software workaround in SDK

ERR007265 CCM: When improper low-power sequence is used, the SoC enters low power mode before the Arm core executes WFI

Description:

When software tries to enter Low-Power mode with the following sequence, the SoC enters Low-Power mode before the Arm core executes the WFI instruction:

1. Set CCM_CLPCR[1:0] to 2'b00.
2. Arm core enters WFI.
3. Arm core wakes up from an interrupt event, which is masked by GPC or not visible to GPC, such as an interrupt from a local timer.
4. Set CCM_CLPCR[1:0] to 2'b01 or 2'b10.
5. Arm core executes WFI.

Before the last step, the SoC enters WAIT mode if CCM_CLPCR[1:0] is set to 2'b01, or STOP mode if CCM_CLPCR[1:0] is set to 2'b10.

Projected Impact:

This issue can lead to errors ranging from module underrun errors to system hangs depending on the specific use case.

Workarounds:

Software workaround:

1. Software should trigger IRQ #41 (GPR_IRQ) to be always pending by setting IOMUXC_GPR_GPR1_GINT.
2. Software should then unmask IRQ #41 in GPC before setting CCM Low-Power mode.
3. Software should mask IRQ #41 right after CCM Low-Power mode is set (set bits 0-1 of CCM_CLPCR).

Proposed Solution:

No fix scheduled

Software Status:

Software workaround in SDK

ERR011092 DCDC: Some power up sequence may result in DCDC startup failure**Description:**

When NVCC_SNVS_IN is powered earlier than or with DCDC_IN/DCDC_IN_Q, DCDC may not be able to start up and no output on DCDC_LP.

Projected Impact:

This issue leads chip failed to power up when using on-chip DCDC.

Workarounds:

Apply power to DCDC_IN/DCDC_IN_Q with DCDC_PSWITCH pin, 1ms ahead of VDD_SNVS_IN.

Proposed Solution:

Fixed in A1 silicon

Software Status:

No software workaround available

ERR011093 DCDC: Unexpected DCDC reset occurs on some chips**Description:**

The possibility and frequent of the occurrence increases with DCDC_IN/DCDC_IN_Q voltage. When the unexpected reset happens, DCDC output drops first, then recovers after unexpected reset completed. In rare case, DCDC might not be able to recover and have output on DCDC_LP. This issue affects both CCM and DCM mode.

Projected Impact:

This issue results in system error, lock up reset, or power loss of VDD_SOC_IN during chip running when using on-chip DCDC.

Workarounds:

Power supply to the DCDC_IN/DCDC_IN_Q should be ≥ 2.8 V and ≤ 3.0 V

Proposed Solution:

Fixed in A1 silicon

Software Status:

No software workaround available

ERR005829 FlexCAN: FlexCAN does not transmit a message that is enabled to be transmitted in a specific moment during the arbitration process

Description:

FlexCAN does not transmit a message that is enabled to be transmitted in a specific moment during the arbitration process. The following conditions are necessary for the issue to occur:

- Only one message buffer is configured to be transmitted.
- The write which enables the message buffer to be transmitted (write on Control/Status word) happens during a specific clock during the arbitration process.
- After this arbitration process occurs, the bus goes to the Idle state and no new message is received on the bus.

For example:

1. Message buffer 13 is deactivated on RxIntermission (write 0x0 to the CODE field from the Control/Status word) [First write to CODE]
2. Reconfigure the ID and data fields
3. Enable the message buffer 13 to be transmitted on BusIdle (write 0xC on CODE field) [Second write to CODE]
4. CAN bus keeps in Idle state
5. No write on the Control/Status from any message buffer happens.

During the second write to CODE (step 3), the write must happen one clock before the current message buffer 13 to be scanned by arbitration process. In this case, it does not detect the new code (0xC) and no new arbitration is scheduled.

The problem can be detected only if the message traffic ceases and the CAN bus enters into Idle state after the described sequence of events.

There is no issue if any of the conditions below holds:

- Any message buffer (either Tx or Rx) is reconfigured (by writing to its CS field) just after the Intermission field.
- There are other configured message buffers to be transmitted.
- A new incoming message sent by any external node starts just after the Intermission field.

Projected Impact:

FlexCAN does not transmit a message that is enabled to be transmitted in a specific moment.

Workarounds:

To transmit a CAN frame, the CPU must prepare a message buffer for transmission by executing the following standard 5-step procedure:

1. Check if the respective interrupt bit is set and clear it.

2. If the message buffer is active (transmission pending), write the ABORT code (0b1001) to the CODE field of the Control/Status word to request an abortion of the transmission. Wait for the corresponding IFLAG to be asserted by polling the IFLAG register or by the interrupt request if enabled by the respective IMASK. Then read back the CODE field to check if the transmission was aborted or transmitted. If backwards compatibility is desired (MCR[AEN] bit negated), just write the INACTIVE code (0b1000) to the CODE field to inactivate the message buffer, but then the pending frame might be transmitted without notification.
3. Write the ID word.
4. Write the data bytes.
5. Write the DLC, Control, and CODE fields of the Control/Status word to activate the message buffer.
6. The workaround consists of executing two extra steps:
7. Reserve the first valid mailbox as an inactive mailbox (CODE = 0b1000). If RX FIFO is disabled, this mailbox must be message buffer 0. Otherwise, the first valid mailbox can be found using the “RX FIFO filters” table in the FlexCAN chapter of the chip reference manual.
8. Write twice INACTIVE code (0b1000) into the first valid mailbox.

NOTE

The first mailbox cannot be used for reception or transmission process.

Proposed Solution:

No fix scheduled

Software Status:

Software workaround is not in SDK.

ERR006032 FlexCAN: A frame with wrong ID or payload is transmitted into the CAN bus when the Message Buffer under transmission is either aborted or deactivated while the CAN bus is in the Bus Idle state

Description:

The FlexCAN module may transmit an incorrect message if one or more Message Buffers (MBs) are configured for transmission while FlexCAN is in the Bus Idle state, and the MB selected for transmission is either aborted or deactivated at the exact moment it starts to be transmitted. This will cause FlexCAN to transmit a syntactically correct message, but with either incorrect ID or data field. The CRC information will be calculated over the incorrect data (in case data is affected) and all other fields of the frame will be correct.

The probability of the problem occurring is limited to one CAN bit during the transmission of one frame, however under a very specific combination of simultaneous events:

1. Bug event may take place in one specific CAN bit per frame.
2. The CAN bus must be in Bus Idle state.
3. The CPU must be triggered to configure one or more MBs for transmission while in the Bus Idle state.
4. The CPU must be triggered to remove the MBs just configured, by abortion or deactivation, in a short period after starting the configuration in step 3.

In summary, the probability of occurrence is very low, in the order of 1 per 10 million. Moreover, the procedure of configuring a MB followed by abortion or deactivation of the same MB in a short interval is unlikely to occur in normal applications.

In practice, there is no issue if the CPU guarantees that any MB configured for transmission will be aborted or deactivated just in the next frame.

Workarounds:

The user can avoid the error by preventing to make MB configurations for transmission when the CAN bus is in the Bus Idle state.

There are bits in a FlexCAN debug register that can be used to determine when the CAN bus is in the Idle state.

This debug register is located at: FlexCAN Debug 1 Register (CAN_DBG1) - Base + 0x0058

The CAN Finite State Machine (CFSM) bits of CAN_DBG1 register monitor the FlexCAN's internal state. The CFSM is the six least significant bits of the CAN_DBG1 register. The CAN Bit Number (CBN) is the five bits long field at bit position 3 to 7 in the CAN_DBG1 register that indicates the current bit number in a given CFSM state value.

CAN_DBG1.CFSM = 0x0000_003F

CAN_DBG1.CBN = 0x1F00_0000

There are several internal states values that need to be looked for, listed below with their corresponding CFSM value.

RXINTERMISSION - 0x2F

TXINTERMISSION - 0x14

BUSIDLE - 0x02

The following procedure must be performed to configure a MB transmission:

1. Disable all interrupt
2. Read CAN_DBG1.CFSM and CAN_DBG1.CBN fields.
3. Check if CFSM value is either BUSIDLE, RXINTERMISSION or TXINTERMISSION.

For the later two values, also check if CBN value is 3, to determine the paired conditions RXINTERMISSION bit 3 or TXINTERMISSION bit 3, and processed as described below.

- 3.1. If CAN_DBG1 fields indicate BUSIDLE, wait N CPU clocks.
- 3.2. Else if CAN_DBG1 fields indicate either RXINTERMISSION bit 3 or TXINTERMISSION bit 3 wait until CFSM is different from either RXINTERMISSION or TXINTERMISSION.
- 3.3. Check again CAN_DBG1 fields, if they indicate BUSIDLE, wait for DELAY time.
4. Write 0x0 into Code field of CS word.
5. Enable all interrupts.
6. Write the ID word.
7. Write the DATA words.
8. Write 0xC into Code field of CS word.

NOTE

$$\text{DELAY} = \{2 \times (\text{MAXMB} + 1) + 18\} \times \text{peripheral_clock_period} + 3 \times \text{PE_clock_period} + 1 \times \text{CAN_bit_period}$$

The Number of the Last Message Buffer (MAXMB) are the 7 least significant bits in the Module Configuration Register (CAN_MCR: base + 0x0).

Proposed Solution:

No fix scheduled

Software Status:

Software workaround is not in SDK.

ERR009527 FlexCAN: The transmission abort mechanism may not work properly**Description:**

The Flexible Controller Area Network (FlexCAN) is not able to abort a transmission frame and the abort process may remain pending on the following cases:

1. If a pending abort request occurs while the FlexCAN is receiving a remote frame.
2. When a frame is aborted during an overload frame after a frame reception.
3. When an abort is requested while the FlexCAN has just started a transmission.
4. When the Freeze Mode request occurs and the FlexCAN has just started a transmission.

Workarounds:

Use the Mailbox Inactivation mechanism instead of the transmission abort mechanism. The Abort Enable (AEN) bit of the Module Configuration Register can be kept cleared and the abort code value "0b1001" cannot be written into the CODE field of the Message Buffer Control and Status word.

Proposed Solution:

No fix scheduled

Software Status:

Software workaround is not in SDK.

ERR009595 FlexCAN: Corrupt frame possible if the Freeze Mode or the Low-Power Mode are entered during a Bus-Off state

Description:

If the Freeze Enable bit (FRZ) of the Module Configuration Register (MCR) is asserted and the Freeze Mode is requested by asserting the Halt bit (HALT) of the MCR register during the Bus Off state, the transmission after exiting the Bus-Off condition will be corrupted. The issue occurs only if a transmission is pending before the Freeze Mode request. In addition, the same issue can happen if the Low-Power Mode is requested instead of the Freeze Mode.

Workarounds:

The workaround depends on whether the Bus-Off condition occurs prior to requesting Freeze Mode or the Low-Power Mode.

Procedure to enter the Freeze Mode:

1. Set the Freeze Enable bit (FRZ) in the Module Control Register (MCR).
2. Check if the Module Disable bit (MDIS) in the MCR register is set. If yes, clear the MDIS bit.
3. Poll the MCR register until the Low-Power Mode Acknowledge (LPMACK) bit in the MCR is cleared (timeout for software are implementation is two CAN bits length).
4. Read the Fault Confinement State (FLTCONF) field in the Error and Status 1 Register (ESR1) to check if FlexCAN is in bus-off state. If yes, go to the step 5A. Otherwise, go to step 5B.

5A. Set the Soft Reset bit ((SOFTTRST) in the MCR.

6A. Poll the MCR register until the Soft Reset (SOFTTRST) bit is cleared (timeout for software are implementation is two CAN bits length).

7A. Poll the MCR register until the Freeze Acknowledge (FRZACK) bit is set (timeout for software are implementation is two CAN bit length).

8A. Reconfigure the MCR.

9A. Reconfigure all the Interrupt Mask Registers (IMASKn).

5B. Set the Halt Flex CAN (HALT) bit in the MCR.

6B. Poll the MCR register until the Freeze Acknowledge (FRZACK) bit is set (timeout for software are implementation is 178 CAN bits length).

NOTE

The time between step 4 and step 5B must be less than 1353 CAN bit periods.

Procedure to enter the Low-Power Mode:

1. Enter the Freeze Mode (execute the procedure A).
2. Request the Low-Power Mode.

3. Poll the MCR register until the Low-Power Mode Acknowledge (LPMACK) bit in the MCR is set (timeout for software are implementation is two CAN bit length).

Proposed Solution:

No fix scheduled

Software Status:

Software workaround is not in SDK.

**ERR011207 FlexSPI: In rare condition when
FLEXSPI_AHB[CR[PREFETCHEN]] is set, incorrect data can be
returned**

Description:

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

Workarounds:

There are two workarounds:

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

Proposed Solution:

Fixed in A1 silicon

Software Status:

Software workaround is not in SDK.

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

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

Workarounds:

Add delay time (100 NOP) again after the DLL lock status is set.

Proposed Solution:

No fix scheduled

Software Status:

Software workaround is not in SDK.

**ERR011091 IO: High POR current if
NVCC_EMC/NVCC_GPIO/NVCC_SD0/NVCC_SD1 is powered
ahead of VDD_SOC_IN**

Description:

When NVCC_EMC/NVCC_GPIO/NVCC_SD0/NVCC_SD1 is powered and VDD_SOC_IN is not powered, it will be high leakage current during the period of VDD_SOC_IN absence. The leakage current on NVCC_EMC/NVCC_GPIO could be high enough to damage the chip permanently. There could also be abnormal leakage on VDD_SNVS_IN when it is powered up, however the period is very short and the current is not high enough to damage the chip.

Projected Impact:

This issue causes high leakage power on IO supply, even permanent chip damage during power up stage, or anytime VDD_SOC_IN is absent but IO is powered.

Workarounds:

When using on-chip DCDC, apply power to DCDC_IN/DCDC_IN_Q 2ms ahead of NVCC_EMC/NVCC_GPIO/NVCC_SD0/NVCC_SD1. If on-chip DCDC is not used, apply power to VDD_SOC_IN ahead of NVCC_EMC/NVCC_GPIO/NVCC_SD0/NVCC_SD1.

Proposed Solution:

Fixed in A1 silicon

Software Status:

No software workaround available

ERR011138 LCDIF: LUT consecutive programming may fail in case two writes are close**Description:**

When two writes to LCDIF_LUT0/1_DATA register are very close, data may not be programmed into LUT successfully. It is related with CPU, IP bus, and LCDIF DISPLAY CLOCK frequency. More faster the CPU runs, more slower the DISPLAY CLOCK is, more likely this issue happens.

Projected Impact:

LUT may not be programmed successfully and LCDIF output data is incorrect.

Workarounds:

Delay need be inserted between two LUT writes. The safe delay should be 6 BUS CLOCK (apb_clk) cycles + 3 DISPLAY CLOCK (pix_clk) cycles.

Proposed Solution:

Fixed in A1 silicon

Software Status:

Software workaround in SDK

ERR011097 LPSPI: Command word does not load properly when TXMSK = 1**Description:**

When writing the TCR[TXMSK] = 1 in the Transmit Command Register and next write to the TX FIFO is another command, the first command may not load properly.

Projected Impact:

Command word does not load properly and LPSPI does not work properly.

Workarounds:

When writing the TCR[TXMSK] = 1 in the Transmit Command Register, wait for the TX FIFO to empty (FSR[TXCOUNT] = 0) before writing another command to the Transmit Command Register.

Proposed Solution:

No fix scheduled

Software Status:

Software workaround is not in SDK.

ERR011111 RTWDOG: Chip stuck in reset when RTWDOG low byte test mode is enabled**Description:**

When the TST bits of the RTWDOG CS register are set to b10, the watchdog low byte test mode will be enabled. In this mode, CNT[**CNTLOW**] is compared with TOVAL[**TOVALLOW**]. System reset happens when they are equal. Then the chip stuck in the reset.

Projected Impact:

RTWDOG test mode cannot be used to test low byte of the watchdog counter.

Workarounds:

Do not set the TST bits of the RTWDOG CD register to b10. If watchdog test is required, set the TST bits to b10 in the User mode to test low byte of the watchdog counter.

Proposed Solution:

Fixed in A1 silicon

Software Status:

No software workaround available

ERR011096 SAI: The internal bit clock cannot be generated when BCI = 1**Description:**

When SAI transmitter or receiver is configured for the internal bit clock with BCI = 1, the bit clock cannot be generated for either of the following two configurations:

1. SYNC = 00 and BCS = 0
2. SYNC = 01 and BCS = 1

Projected Impact:

The SAI bit clock cannot be generated properly.

Workarounds:

When SAI transmitter or receiver is configured for the internal bit clock with BCI = 1, using one of the following two configurations:

1. SYNC = 01 and BCS = 0
2. SYNC = 00 and BCS = 1

Proposed Solution:

No fix scheduled

Software Status:

Software workaround is not in SDK.

ERR011225 SEMC: CPU AXI writes to SEMC NAND memory may cause incorrect data programmed into NAND memory**Description:**

When SEMC NAND memory region is Normal type, non-cacheable, cacheable write-through, or writeback, non-allocate, and not hit, CM7 AXI writes to the region could program incorrect data to the NAND memory.

Projected Impact:

CPU cannot perform AXI write to SEMC NAND memory when it is the Normal memory type.

Workarounds:

1. Set SEMC NAND memory region to Device type or Strongly-ordered type in MPU, and CPU only perform 32-bit write to SEMC NAND memory region or;
2. Use eDMA to perform 64-bit AXI write to SEMC NAND memory region or;
3. Use IP command to program SEMC NAND memory.

Proposed Solution:

No fix scheduled

Software Status:

Software workaround is not in SDK.

ERR011165 SNVS: Invalid ECC check failure**Description:**

When setting LPMKCR[ZMK_ECC_EN] bit, it may generate ZMK ECC Check Failure Violation even the ZMK and its ECC values are correct.

Projected Impact:

ZMK is not usable in case the ZMK ECC check is enabled.

Workarounds:

Not enable ZMK ECC check

Proposed Solution:

Fixed in A1 silicon

Software Status:

Software workaround is not in SDK.

ERR011262 System Boot: ROM cannot boot from SEMC NAND with default fuse setting

Description:

ROM cannot boot from SEMC NAND with default fuse setting because the parameter configuration of SEMC AXI read is incorrect.

For SEMC NAND AXI read, tWHR is not the one defined in the ONFI standard. Instead of tWHR, it is tCSS, which needs to be read from device. It is unique to device, there is no uniform standard.

Projected Impact:

ROM cannot boot from SEMC NAND with default fuse setting.

Workarounds:

Configure FUSE to select non-ONFI device (BOOT_CFG2[0] = 1'b1) and enable EDO mode (0x6E0[4] = 1'b1), then AXI read can work properly (because the ONFI timing mode is selected by default, and tWHR in the ONFI timing mode 0 is long enough for AXI read). The NAND configuration block in FCB is able to adjust tWHR parameter. So it can switch to any timing mode.

Proposed Solution:

Fixed in A1 silicon

ERR011110 System Boot: SEMC NOR boot cannot support the signed image authentication under HAB closed mode**Description:**

When the device is configured as HAB closed mode, BootROM cannot boot from SEMC NOR device because the signed image authentication is not supported.

Projected Impact:

This issue primarily impacts the application that requires Parallel NOR boot.

Workarounds:

Do not boot application via Parallel NOR under HAB closed mode.

Proposed Solution:

Fixed in A1 silicon

Software Status:

No software workaround available

ERR011119 System Boot: FlexSPI NOR encrypted XIP boot fails after system reset if the FAC region number is less than 2

Description:

If the total FAC region number in PRDB0 and PRDB1 is less than 2, ROM fails to boot from FlexSPI NOR on RT1051 and RT1052 A0 chip after system reset.

Conditions:

There are two conditions cause a boot failure:

- Only PRDB0 is available and FAC region is 1 in PRDB0.
- Only PRDB1 is available and FAC region is 1 in PRDB1.

Projected Impact:

User application cannot run after WDOG reset or system reset.

Workarounds:

There are two workarounds:

- Set IOMUXC_GPR20 = 0 and IOMUXC_GPR21 = 0 in user application after boot.
- Enable more than one FAC region in PRDB0 and PRDB1.

Proposed Solution:

Fixed in A1 silicon

Software Status:

Software workaround is not in SDK.

ERR011120 System Boot: FlexSPI NOR encrypted XIP boot fails after system reset if the IOMUXC_GPR18 to IOMUXC_GPR21 are locked**Description:**

If the IOMUXC_GPR18 to IOMUXC_GPR_21 are locked by either BootROM or the user application, ROM fails to boot from FlexSPI NOR on RT1051 and RT1052 A0 chip after system reset.

Conditions:

There are two conditions cause a boot failure:

- Enable the lock option in PRDB0 or PRDB1.
- Enable the lock bit for IOMUXC_GPR18 to IOMUXC_GPR21 in user application.

Projected Impact:

User application cannot run after WDOG reset or system reset.

Workarounds:

Not available

Proposed Solution:

Fixed in A1 silicon

Software Status:

No software workaround available

ERR006281 USB: Incorrect DP/DN state when only VBUS is applied**Description:**

When VBUS is applied without any other supplies, incorrect communication states are possible on the data (DP/DN) signals. If VDDHIGH_IN is supplied, the problem is removed.

Projected Impact:

This issue primarily impacts applications using charger detection to signal power modes to a PMIC in an undercharged battery scenario where the standard USB current allotment is not sufficient to boot the system.

Workarounds:

Apply VDDHIGH_IN if battery charge detection is needed. Otherwise, disable charger detection by setting the EN_B bit in USB_ANALOG_USBx_CHRG_DETECTn to 1.

Proposed Solution:

No fix scheduled

Software Status:

Software workaround is not in SDK.

ERR010661 USB: VBUS leakage occurs if USBOTG1 VBUS is on and USBOTG2 VBUS transitions from on to off**Description:**

When two USB ports work as OTG or device simultaneously. One VBUS (selected by PMU_REG_3P0.vbus_sel bit) voltage will not drop after cable unplug, causing the port to fail to detect the cable detach. If these two ports do not need to support detach detection, simultaneously using two OTGs or devices can be supported.

Conditions:

When two USB ports work as OTGs or devices simultaneously.

Projected Impact:

Do not use two OTGs or devices simultaneously. Only four scenarios are supported:

- One for OTG/Device, another for Host.
- One for OTG/Device, another is un-used.
- One for Host, another for Host.
- One for Host, another is un-used.

Workarounds:

Only one port can be used as OTG or device. The other port must be used as host. Set the PMU_REG_3P0.vbus_sel bit to select the host port.

Proposed Solution:

No fix scheduled

Software Status:

No software workaround available



How to Reach Us:

Home Page:

nxp.com

Web Support:

nxp.com/support

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

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

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

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. These are contained within the 6 series RMs Arm, AMBA, Arm Powered, Cortex, Jazelle, Thumb, and TrustZone are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. CoreLink, CoreSight, and NEON are trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2017-2018 NXP B.V.

Document Number: IMXRT1050CE

Rev. 1.1

08/2018

