RT600

Errata sheet RT600

Rev. 1.9 — March 31, 2022

Errata sheet

Document information

Info	Content
Keywords	MIMXRT685SFFOB, MIMXRT685SFVKB, MIMXRT685SFAWBR, MIMXRT633SFVKB, MIMXRT633SFAWBR
Abstract	RT600 errata

Revision history

Rev	Date	Description
1.9	20220331	Added Section 5.2 “ROM.2: Usage of ROM APIs when trust zone is disabled”.
1.8	20211111	Added ROM.1 errata, Section 3.8 “ROM.1: Non-Secure Boot ROM API initializes unused FlexSPI0 IO pins”.
1.7	20210420	Added USB.3 errata, Section 3.6 “USB.3: In USB high-speed device mode, when device isochronous IN endpoint sends a packet of MaxPacketSize of 1024 bytes in response to IN token from host, the isochronous IN endpoint interrupt is not set and the endpoint command/status list entry for the isochronous IN endpoint is not updated”. Added USB.4 errata, Section 3.7 “USB.4: In USB high-speed host mode, only one transaction per micro-frame is allowed for isochronous IN endpoints”.
1.6	20210225	Added USB.2 errata, Section 3.5 “USB.2: In USB high-speed device mode, device writes extra byte(s) to the buffer if the NBytes is not multiple of 8 for OUT transfer”.
1.5	20201218	Includes Section 5.1 “Leakage path between VDD1V8 and VDDIO_x”.
1.4	20201214	Includes Section 3.4 “USB.1: For the USB high-speed device controller, the detection handshaking fails when certain full-speed hubs are connected”.
1.3	20200729	Updates Section 3.2 “GPIO.1: During initial power-up, a brief pull-up pulse could occur on the port pins” and adds Section 3.3 “ADC.1: ADC misses software and hardware triggers when there is no ADC clock”.
1.2	20200706	Adds Section 3.2 “GPIO.1: During initial power-up, a brief pull-up pulse could occur on the port pins”.
1.1	20200508	Added FlexSPI DLL lock status timing issue and addressed part marking.
1.0	20200213	Initial version.

Contact information

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

For sales office addresses, please send an email to: **salesaddresses@nxp.com**

1. Product identification

The MIMXRT6xxSFAWBR WLCSP114 production samples has the following top-side package marking:

- First line: MRT6xxSF
- Second line: AW[R]R
- Third line: xxxxxx xx
- Fourth line: xxxxyyww
  - yyww: Date code with yy = year and ww = week

    Fifth line: xxx-yyy
- Sixth line: NXP

The MIMXRT6xxSFVKB VFBGA176 production samples has the following top-side package marking:

- First line: MRT6xxSFV
- Second line: K[R] xxxxxx
- Third line: xxyyww
- Fourth line: xxxxx
  - yyww: Date code with yy = year and ww = week

The MIMXRT685SFFOB FOWLP249 production samples has the following top-side package marking:

- First line: MRT6xxSFFOB
- Second line: xxxxxx
- Third line: xxxxxx
- Fourth line: xxxxxyyww
  - yyww: Date code with yy = year and ww = week

Table 1. Device revision table

Revision identifier	Revision description [R]
B	Initial device revision

2. Errata overview

Table 2. Functional problems table

Functional problems	Short description	Revision identifier	Detailed description
FlexSPI	FlexSPI DLL lock status bit not accurate due to timing issue.	B	Section 3.1 "FlexSPI.1: FlexSPI DLL lock status bit not accurate due to timing issue"
GPIO.1	During initial power-up, a brief pull-up pulse could occur on the port pins.	B	Section 3.2 "GPIO.1: During initial power-up, a brief pull-up pulse could occur on the port pins"
ADC.1	ADC misses software and hardware triggers when there is no ADC clock.	B	Section 3.3 "ADC.1: ADC misses software and hardware triggers when there is no ADC clock"
USB.1	For the USB high-speed device controller, the detection handshaking fails when certain full-speed hubs are connected.	B	Section 3.4 "USB.1: For the USB high-speed device controller, the detection handshaking fails when certain full-speed hubs are connected"
USB.2	In USB high-speed device mode, device writes extra byte(s) to the buffer if the NBytes is not multiple of 8 for OUT transfer.	B	Section 3.5 "USB.2: In USB high-speed device mode, device writes extra byte(s) to the buffer if the NBytes is not multiple of 8 for OUT transfer"

Table 2. Functional problems table *…continued*

Functional problems	Short description	Revision identifier	Detailed description
USB.3	In USB high-speed device mode, when device isochronous IN endpoint sends a packet of MaxPacketSize of 1024 bytes in response to IN token from host, the isochronous IN endpoint interrupt is not set and the endpoint command/status list entry for the isochronous IN endpoint is not updated.	B	Section 3.6 “USB.3: In USB high-speed device mode, when device isochronous IN endpoint sends a packet of MaxPacketSize of 1024 bytes in response to IN token from host, the isochronous IN endpoint interrupt is not set and the endpoint command/status list entry for the isochronous IN endpoint is not updated”
USB.4	In USB high-speed host mode, only one transaction per micro-frame is allowed for isochronous IN endpoints.	B	Section 3.7 “USB.4: In USB high-speed host mode, only one transaction per micro-frame is allowed for isochronous IN endpoints”
ROM.1	Non-Secure Boot ROM API initializes unused FlexSPI0 IO pins.	B	Section 3.8 “ROM.1: Non-Secure Boot ROM API initializes unused FlexSPI0 IO pins”

Table 3. AC/DC deviations table

AC/DC deviations	Short description	Product version(s)	Detailed description
n/a	n/a	n/a	n/a

Table 4. Errata notes

Errata notes	Short description	Revision identifier	Detailed description
VDD.1	Leakage path between VDD1V8 and VDDIO_x	B	Section 5.1 “VDD.1: Leakage path between VDD1V8 and VDDIO_x”.
ROM.2	Usage of ROM APIs when trust zone is disabled	B	Section 5.2 “ROM.2: Usage of ROM APIs when trust zone is disabled”

3. Functional problems detail

3.1 FlexSPI.1: FlexSPI DLL lock status bit not accurate due to timing issue

Introduction

Based on the sample clock source selection, the DLL control register (DLLxCR) can be used to set the delay line chain which allows a fixed number of delay cells or auto-adjusted to lock on a certain phase delay to the reference clock.

Problem

After configuring the DLL and setting the lock status bit, data may not be in sync if a read/write is performed immediately from a FLEXSPI based external flash due to timing issues.

Work-around

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

3.2 GPIO.1: During initial power-up, a brief pull-up pulse could occur on the port pins

Introduction

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

Problem

During VDDIO_x power-up, the internal pull-up resistor may not initialize during the early part of the VDDIO_x ramp up, resulting in a brief pull-up current pulse on some port pins that drops to zero before the VDDIO_x supplies reach the minimum operating voltage. Except for PIO1_19 to PIO1_31, PIO2_0 to PIO2_8, PIO0_21, PIO0_22, and PIO_23 pins, all fail safe GPIOs are affected by this issue.

Typically, for a 20 ms power-up ramp, the pulse width of the glitch is approximately 8 ms and the amplitude is about 1.2 V.

Work-around

As a 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.

3.3 ADC.1: ADC misses software and hardware triggers when there is no ADC clock

Introduction

ADC command execution can be initiated from up to 16 trigger sources. These triggers can be generated either via software trigger or hardware trigger.

Problem

When no ADC clock is present, the ADC will not properly capture software or hardware trigger events. The following set of conditions will cause this behavior:

- First, the system enters low power state (both bus and functional clocks get disabled).
- Then, the system receives a wake up and, upon exit from the lower power state, the CPU clocks start running.
- Finally, the ADC receives a software or hardware trigger before the functional ADC clock has completed start up and misses the trigger event.

Work-around

In order to use software or hardware triggers, the ADC must be left powered and the ADC source clock must be kept enabled upon entry into the low power state. Please note, when exiting the low power state, 3 ADC Clock cycle synchronization delay is required between waking up until SWTRIG can be accepted.

3.4 USB.1: For the USB high-speed device controller, the detection handshaking fails when certain full-speed hubs are connected

Introduction

See the USB2.0 specification for details regarding the USB High-speed Detection Handshake protocol.

Problem

As a high-speed device, when certain full-speed hubs are connected, the USB device does not detect the HOST KJ sequence correctly and, as a result, does not recognize the speed of the connected host. In this case, the USB device can act erratically due to the wrong speed detection.

Work-around

There are two workarounds:

1. The software work-around below can be implemented in usb_dev_hid_mouse where API is called "USB_DeviceHsPhyChirpIssueWorkaround()". In event handler in USB_DeviceCallback(),
   - On "kUSB_DeviceEventBusReset" event, USB_DeviceHsPhyChirpIssueWorkaround() should be called to identify the speed of the host connected to. If full-speed host is connected or "isConnectedToFsHostFlag" is set , FORCE_FS (bit 21) of DEVCMDSTAT register should be set to force the device operating in full-speed mode.
   - On "kUSB_DeviceEventDetach" event, FORCE_FS (bit 21) of DEVCMDSTAT register should be cleared.
2. The software workaround below is available in tech note (TN00071) In event handler in USB_DeviceCallback(),
   - On "kUSB_DeviceEventAttach" event, set PHY_RX register trip-level voltage to the highest. USBPHY->RX &= ~(USBPHY_RX_ENVADJ_MASK);USBPHY->RX |= 2;.

- On "kUSB_DeviceEventBusReset" event, check the DEVCMDSTAT[SPEED] to determine the connected bus speed. (SPEED are bits 22 and 23). If DEVCMDSTAT[SPEED]=FS, FORCE_FS (bit 21) of DEVCMDSTAT should be set to force the device operating in full-speed mode.
- On "kUSB_DeviceEventGetDeviceDescriptor" event, or first SETUP packet has arrived, Set the USBPHY_RX[ENVADJ] field back to default 0. Otherwise, USBPHY_RX[ENVADJ] field will remains as 2 unless a disconnect event occurs.
- On "kUSB_DeviceEventDetach" event, Clear FORCE_FS (bit 21) of DEVCMDSTAT register to zero. Reset USBPHY_RX[ENVADJ] field back to default 0.

3.5 USB.2: In USB high-speed device mode, device writes extra byte(s) to the buffer if the NBytes is not multiple of 8 for OUT transfer

Introduction

The RT600 device family include a USB high-speed interface (USB1) that can operate in device mode at high-speed. The NBytes value represents the number of bytes that can be received in the buffer.

Problem

The RT600 USB device controller writes extra bytes to the receive data buffer if the size of the transfer is not a multiple of 8 bytes since the USB device controller always writes 8 bytes. For example, if the transfer length is 1 bytes, 7 extra bytes will be written to the receive data buffer. If the transfer length is 7 bytes, 1 extra bytes will be written to the receive data buffer.

Work-around

Reserve an additional, intermediary buffer along with the buffer used by the application for USB data. After the USB data transfer into the intermediary buffer has been completed, use memcpy to move the data from the intermediary buffer into the application buffer, skipping the extraneous extra byte. This software work-around is implemented on the SDK software platform.

3.6 USB.3: In USB high-speed device mode, when device isochronous IN endpoint sends a packet of MaxPacketSize of 1024 bytes in response to IN token from host, the isochronous IN endpoint interrupt is not set and the endpoint command/status list entry for the isochronous IN endpoint is not updated

Introduction

The RT600 device family include a USB high-speed interface (USB1) that can operate in device mode at high-speed. The isochronous IN endpoint supports a MaxPacketSize of 1024 bytes.

Problem

When device isochronous IN endpoint sends a packet of MaxPacketSize of 1024 bytes in response to IN token from host, the isochronous IN endpoint interrupt is not set and the endpoint command/status list entry for the isochronous IN endpoint is not updated.

Work-around

Restrict the isochronous IN endpoint MaxPacketSize to 1023 bytes in device descriptor.

3.7 USB.4: In USB high-speed host mode, only one transaction per micro-frame is allowed for isochronous IN endpoints

Introduction

The RT600 device family include a USB high-speed interface which can operate in host mode. Up to three high-speed transactions are allowed in a single micro-frame to support high-bandwidth endpoints. This mode is enabled by setting the Mult (Multiple) field in the Proprietary Transfer Descriptor (PTD) and is used to indicate to the host controller the number of transactions that should be executed per micro-frame. The allowed bit settings are:

00b Reserved. A zero in this field yields undefined results.

01b One transaction to be issued for this endpoint per micro-frame.

10b Two transactions to be issued for this endpoint per micro-frame.

11b Three transactions to be issued for this endpoint per micro-frame.

Problem

For High-bandwidth mode, using multiple packets (MULT = 10b or 11b) in a frame causes unreliable operation. Only one transaction (MULT = 01b) can be issued per micro-frame.

Work-around

There is no software workaround. Only one transaction can be issued per micro-frame.

3.8 ROM.1: Non-Secure Boot ROM API initializes unused FlexSPI0 IO pins

Introduction

Each port IO pin has a dedicated control register in the IOPCTL module that allows control of various functions and characteristics. By default, the port IO pins have their input buffer disabled. This keeps pins that may be left floating from causing excess current leakage.

During the FlexSPI boot flow, the ROM API IAP_FlexspiNorAutoConfig() is called, for initialization of the FlexSPI module before accessing the Flash memory. The ROM configures the FLEXSPI0 pins (PIO1_18 - PIO1_28) enabling the input buffers for those respective pins regardless of what memory device is used.

Problem

If the application does not use these FlexSPI0 pins as inputs, it should disable the input buffers for these pins in the IOPCTL registers via bit 6 (IBENA).

Work-around

None.

4. AC/DC deviations detail

5. Errata notes detail

5.1 VDD.1: Leakage path between VDD1V8 and VDDIO_x

On RT600, the power sequencing specification in the datasheet mentions that the VDDIO_x rail can be optionally powered after the VDD1V8 and the delta voltage between VDDIO_x and VDD1V8 must be 1.89 V or less.

Before the VDDIO_x is powered, there is a leakage path between the VDD1V8 and VDDIO_x domain. The leakage is approximately 1.5 mA (VDD1V8 - VDDIO / 800 ohm). This leakage does not cause any reliability issues. There is no leakage once the VDDIO_x rail is above VDD1V8 - 0.4 V.

5.2 ROM.2: Usage of ROM APIs when trust zone is disabled

On the RT600, ROM APIs are not available only when trust zone is disabled using OTP setting or image header, and when user image boots in non-secure mode.

6. Contents

All information provided in this document is subject to legal disclaimers.

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

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

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

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.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. 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.

Table continues on the next page...

and service marks licensed by Power.org. M, M Mobileye and other Mobileye trademarks or logos appearing herein are trademarks of Mobileye Vision Technologies Ltd. in the United States, the EU and/or other jurisdictions.

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

Date of release: March 31, 2022
Document identifier: ES-RT600