

Stellaris® LM3S9B92 RevB1 Errata

This document contains known errata at the time of publication for the Stellaris LM3S9B92 microcontroller. The table below summarizes the errata and lists the affected revisions. See the data sheet for more details.

See also the ARM® Cortex™-M3 errata, ARM publication number PR326-PRDC-009450 v2.0.

Table 1. Revision History

Date	Revision	Description
March 2011	3.6	<ul style="list-style-type: none"> Changed title of issue "GPTM 2A and 2B are not usable with μDMA" to "The μDMA does not generate a completion interrupt when transferring to and from GPTM 2A and 2B" on page 13 and reworded description. Removed Appendix A since information is now in the data sheet.
February 2011	3.5	<ul style="list-style-type: none"> Clarified issue "The PIOSC is not trimmed by the factory" on page 7. Added issue "The PIOSC cannot be calibrated by the user" on page 7. Added issue "Flash memory endurance cycle specification is 100 cycles" on page 13. Added issue "The μDMA does not generate a completion interrupt when transferring to and from GPTM 2A and 2B" on page 13. Added issue "PB0 and PB1 have permanent internal pull-down resistance" on page 15. Added issue "Differential pair encodings are incorrect" on page 23. Added issue "PWM fault latch does not operate correctly" on page 32. Added issue "PWM6 and PWM7 do not function". Added Appendix A.
January 2011	3.4	<ul style="list-style-type: none"> Added issue "PB0 and PB1 have permanent internal pull-down resistance" on page 15.
November 2010	3.2	<ul style="list-style-type: none"> Added clarification to issue "Flash Write Buffer does not function above 50 MHz" on page 12.
October 2010	3.0	<ul style="list-style-type: none"> Added issue "ROM_USBHostMode function is incorrect". Added issue "ROM_CANBitRateSet function is incorrect". Added issue "USB compliance test issue: USB full-speed, far-end signal compliance tests fail with 5 m cable" on page 29. Added issue "USB compliance test issue: USB embedded host low-speed, far-end signal compliance tests fail" on page 30.
September 2010	2.9	<ul style="list-style-type: none"> Removed the "ROM_I2CMasterErr function is incorrect" issue because the data sheet has been changed such that the <code>ERROR</code> bit no longer is set when the <code>ARBLST</code> bit is set. Additional minor clarifications and corrections.
July 2010	2.8	<ul style="list-style-type: none"> Added issue "The RTRIS bit in the UARTRIS register is only set when the interrupt is enabled" on page 25.
June 2010	2.7	<ul style="list-style-type: none"> Minor edits.

Date	Revision	Description
April 2010	2.6	<ul style="list-style-type: none"> Based on further examination of the "I²C arbitration may be lost when operating as a master" issue, this issue has been moved to the GPIO section and renamed as "Schmitt input feature does not function correctly" on page 14. Added issue "Encoding error in the Ethernet MAC LED Encoding (MACLED) register" on page 28. Added information about items fixed on Rev C3.
March 2010	2.5	<ul style="list-style-type: none"> Added issue "The prescaler does not work correctly when counting up in periodic or one-shot mode" on page 21. Added issue "Snapshot must be enabled in both Timer A and B when in 32-bit snapshot mode" on page 22. Added issue "Phantom interrupts occur in Smart Card mode" on page 25. Added issue "I²C arbitration may be lost when operating as a master".
Mar 2010	2.4	<ul style="list-style-type: none"> Added issue "The option to force the ROM boot loader to execute at reset with an external pin does not function" on page 11. Amended the workaround for issue "A spurious DMA request is generated when the timer rolls over in Input-Edge Time mode" on page 19. Reworded description of issue "The value of the prescaler register is not readable in Edge-Count mode" on page 19. Removed "Prescaler register must have a non-zero value in 16-bit Edge-Time mode" as it has been determined this item was included erroneously. Added issue "ADC trigger and Wait-on-Trigger may assert when the timer is disabled" on page 20. Added issue "Wait-on-Trigger does not assert unless the TnOTE bit is set" on page 20 . Added issue "Do not enable match and timeout interrupts in 16-bit PWM mode" on page 20. Added issue "Do not use μDMA with 16-bit PWM mode" on page 21. Added issue "Writing the GPTMTnV register does not change the timer value when counting up" on page 21.
Feb 2010	2.3	<ul style="list-style-type: none"> Added issue "A spurious DMA request is generated when the timer rolls over the 16-bit boundary" on page 19. Added issue "The value of the prescaler register is not readable in Edge-Count mode" on page 19. Added issue "Prescaler register must have a non-zero value in 16-bit Edge-Time mode." Added issue "The ADCSPC register does not function" on page 23.
Jan 2010	2.2	<ul style="list-style-type: none"> Modified description for "The General-Purpose Timer match register does not function correctly in 32-bit mode" on page 18 to include DMA operation. Added issue "A spurious DMA request is generated when the timer rolls over in Input-Edge Time mode" on page 19. Changed workaround for "Latch-up may occur if power is applied to the VBUS pin but not to VDD" on page 29 and changed status to "Fixed in Rev C."

Date	Revision	Description
Dec 2009	2.1	<ul style="list-style-type: none"> ■ The status of "The Recover Locked Device sequence does not work as expected" on page 6 has been changed to "Fixed in Rev C." ■ "Hard Fault possible when waking from Sleep or Deep-Sleep modes and Cortex-M3 Debug Access Port (DAP) is enabled" has been removed and the content added to the LM3S9B92 data sheet. ■ Added additional APIs to "Some ROM functions are unsupported" on page 8. ■ "The μDMA controller fails to generate capture mode DMA requests from Timer A in the Timer modules" on page 13 has been added. ■ "Ethernet packet count decremented before the FCS is read" has been removed and the content added to the LM3S9B92 data sheet. ■ The status of "Latch-up may occur if power is applied to the VBUS pin but not to VDD" on page 29 has been changed to "Not fixed in Rev C."
Nov 2009	2.0	Started tracking revision history.

Table 2. List of Errata

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
1.1	JTAG INTEST instruction does not work	JTAG	B1
1.2	The Recover Locked Device sequence does not work as expected	JTAG	B1
2.1	Sleep and Deep-Sleep mode not usable at higher speeds when ISRs reside in Flash memory	System Control	B1
2.2	Device Capabilities registers may not accurately reflect available signals	System Control	B1
2.3	The PIOSC is not trimmed by the factory	System Control	B1
2.4	The PIOSC cannot be calibrated by the user	System Control	B1
3.1	Ethernet fails to connect when using the Boot Loader software in ROM	ROM	B1
3.2	Some ROM functions are unsupported	ROM	B1
3.3	ROM mapping check for the Boot loader does not function properly	ROM	B1
3.4	ROM_SSICfgSetExpClk function is incorrect	ROM	B1
3.5	ROM_USBFIFOFlush function is incorrect	ROM	B1
3.6	The option to force the ROM boot loader to execute at reset with an external pin does not function	ROM	B1
4.1	Cumulative page erases may introduce bit errors in Flash memory	Flash Memory	B1
4.2	Flash Write Buffer does not function above 50 MHz	Flash Memory	B1
4.3	Flash memory endurance cycle specification is 100 cycles	Flash Memory	B1, C3, C5
5.1	The μ DMA controller fails to generate capture mode DMA requests from Timer A in the Timer modules	μ DMA	B1, C3, C5
5.2	The μ DMA does not generate a completion interrupt when transferring to and from GPTM 2A and 2B	μ DMA	B1, C3, C5
6.1	Port B [1:0] pins require external pull-up resistors	GPIO	B1

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
6.2	Schmitt input feature does not function correctly	GPIO	B1
6.3	PB0 and PB1 have permanent internal pull-down resistance	GPIO	B1, C3, C5
7.1	EPI dual-chip select function does not work	EPI	B1
7.2	EPI Host-Bus 16 mode does not work	EPI	B1
7.3	Clock signal in EPI General-Purpose mode is inverted	EPI	B1
8.1	The General-Purpose Timer match register does not function correctly in 32-bit mode	General-Purpose Timers	B1, C3, C5
8.2	A spurious DMA request is generated when the timer rolls over in Input-Edge Time mode	General-Purpose Timers	B1, C3, C5
8.3	A spurious DMA request is generated when the timer rolls over the 16-bit boundary	General-Purpose Timers	B1, C3, C5
8.4	The value of the prescaler register is not readable in Edge-Count mode	General-Purpose Timers	B1, C3, C5
8.5	ADC trigger and Wait-on-Trigger may assert when the timer is disabled	General-Purpose Timers	B1, C3, C5
8.6	Wait-on-Trigger does not assert unless the TnOTE bit is set	General-Purpose Timers	B1, C3, C5
8.7	Do not enable match and timeout interrupts in 16-bit PWM mode	General-Purpose Timers	B1, C3, C5
8.8	Do not use μ DMA with 16-bit PWM mode	General-Purpose Timers	B1, C3, C5
8.9	Writing the GPTMTnV register does not change the timer value when counting up	General-Purpose Timers	B1, C3, C5
8.10	The prescaler does not work correctly when counting up in periodic or one-shot mode	General-Purpose Timers	B1, C3, C5
8.11	Snapshot must be enabled in both Timer A and B when in 32-bit snapshot mode	General-Purpose Timers	B1, C3, C5
9.1	Writes to Watchdog Timer 1 module WDTLOAD register sometimes fail	Watchdog Timers	B1, C3, C5
10.1	ADC hardware averaging produces erroneous results in differential mode	ADC	B1, C3, C5
10.2	The ADCSPC register does not function	ADC	B1
10.3	Differential pair encodings are incorrect	ADC	B1, C3, C5
11.1	UART Smart Card (ISO 7816) mode does not function	UART	B1
11.2	When in IrDA mode, the UnRx signal requires configuration even if not used	UART	B1
11.3	Phantom interrupts occur in Smart Card mode	UART	B1
11.4	The RTRIS bit in the UARTRIS register is only set when the interrupt is enabled	UART	B1, C3, C5
12.1	An interrupt is not generated when using μ DMA with the SSI module if the EOT bit is set	SSI	B1
13.1	Some bits in the I2SMCLKCFG register do not function	I2S	B1

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
13.2	I ² S SCLK signal is inverted in certain modes	I2S	B1
14.1	Ethernet receive packet corruption may occur when using optional auto-clock gating	Ethernet Controller	B1
14.2	Ethernet packet loss with cables longer than 50 meters	Ethernet Controller	B1
14.3	Ethernet PHY interrupts do not function correctly	Ethernet Controller	B1
14.4	Encoding error in the Ethernet MAC LED Encoding (MACLED) register	Ethernet Controller	B1, C3, C5
15.1	USB0ID and USB0VBUS signals are required to be connected regardless of mode	USB	B1
15.2	Latch-up may occur if power is applied to the VBUS pin but not to VDD	USB	B1
15.3	USB compliance test issue: USB full-speed, far-end signal compliance tests fail with 5 m cable	USB	B1, C3
15.4	USB compliance test issue: USB embedded host low-speed, far-end signal compliance tests fail	USB	B1, C3
16.1	PWM generation is incorrect with extreme duty cycles	PWM	B1
16.2	Sync of PWM does not trigger "zero" action	PWM	B1
16.3	PWM "zero" action occurs when the PWM module is disabled	PWM	B1
16.4	PWM Enable Update register bits do not function	PWM	B1
16.5	PWM fault latch does not operate correctly	PWM	B1, C3, C5
17.1	Momentarily exceeding V _{IN} ratings on any pin can cause latch-up	Electrical Characteristics	B1
17.2	Power-on event may disrupt operation	Electrical Characteristics	B1

1 JTAG

1.1 JTAG INTEST instruction does not work

Description:

The JTAG INTEST (Boundary Scan) instruction does not properly capture data.

Workaround:

None.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

1.2 The Recover Locked Device sequence does not work as expected

Description:

If software configures any of the JTAG/SWD pins as GPIO or loses the ability to communicate with the debugger, there is a debug sequence that can be used to recover the microcontroller, called the Recover Locked Device sequence. After reconfiguring the JTAG/SWD pins, using the Recover Locked Device sequence does not recover the device.

Workaround:

To get the device unlocked, follow these steps:

1. Power cycle the board and run the debug port unlock procedure in LM Flash Programmer. DO NOT power cycle when LM Flash Programmer tells you to.
2. Go to the Flash Utilities tab in LM Flash Programmer and do a mass erase operation (check "Entire Flash" and then click the Erase button). This erase appears to have failed, but that is ok.
3. Power cycle the board.
4. Go to the Flash Utilities tab in LM Flash Programmer and do another mass erase operation (check "Entire Flash" and then click the Erase button).

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

2 System Control

2.1 Sleep and Deep-Sleep mode not usable at higher speeds when ISRs reside in Flash memory

Description:

Sleep and Deep-Sleep modes cannot be used when running the processor at 66 or 80 MHz when the Interrupt Service Routines (ISRs) and vector table reside in Flash memory. If Sleep or Deep-Sleep mode is used at those speeds, an invalid PC is sometimes returned for the interrupt vector address when exiting sleep mode.

Workaround:

There are two possible workarounds for this issue:

1. Store the ISRs and vector table in the on-chip SRAM when running the processor at 66 or 80 MHz.
2. Run the processor at 50 MHz.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

2.2 Device Capabilities registers may not accurately reflect available signals

Description:

Some of the Device Capabilities register bits reflect the presence of specific pins on the microcontroller. These bits do not always properly reflect the available signals. Bits affected include **DC3** [31:0], **DC4** [15:14], **DC5** [27:24] and [7:0], and **DC8** [31:0]. Do not rely on the value of these bits in system design.

Workaround:

None.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

2.3 The PIOSC is not trimmed by the factory

Description:

The PIOSC is not trimmed by the factory prior to shipment. This errata item affects any product with date codes prior to 0946 (work week 46, 2009). When the PIOSC is not trimmed, its accuracy is $\pm 18.75\%$.

Workaround:

For parts that have a Hibernation module, the PIOSC can be user calibrated. The PIOSC cannot be calibrated on parts without a Hibernation module. For more information, see the section entitled, "Precision Internal Oscillator Operation (PIOSC)" in the System Control chapter in the data sheet.

Silicon Revision Affected:

B1

Fixed:

Not fixed.

2.4 The PIOSC cannot be calibrated by the user

Description:

The PIOSC is trimmed by the factory, but cannot be user calibrated using the `UPDATE` bit in the **Precision Internal Oscillator Calibration (PIOSCCAL)** register.

Workaround:

None.

Silicon Revision Affected:

B1

Fixed:

Fixed on devices with date codes of 0x13 (March, 2011) or later.

3 ROM

3.1 Ethernet fails to connect when using the Boot Loader software in ROM

Description:

The Ethernet controller takes longer to connect than the Boot Loader software in ROM allows.

Workaround:

Download the Boot loader software in the on-chip Flash memory and ensure that the Ethernet connection uses MDI mode only.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

3.2 Some ROM functions are unsupported

Description:

The following functions are unsupported in ROM:

- ADCComparatorConfigure
- ADCComparatorRegionSet
- ADCComparatorReset
- ADCComparatorIntDisable
- ADCComparatorIntEnable
- ADCComparatorIntStatus
- ADCComparatorIntClear
- CANBitRateSet
- EPIntStatus
- EPIModeSet
- EPIDividerSet
- EPIConfigSDRAMSet
- EPIConfigGPMModeSet
- EPIConfigHB8Set
- EPIConfigHB16Set
- EPIAddressMapSet
- EPINonBlockingReadConfigure
- EPINonBlockingReadStart
- EPINonBlockingReadStop
- EPINonBlockingReadCount
- EPINonBlockingReadAvail

- EPINonBlockingReadGet32
- EPINonBlockingReadGet16
- EPINonBlockingReadGet8
- EPIFIFOConfig
- EPIWriteFIFOCountGet
- EPIIntEnable
- EPIIntDisable
- EPIIntErrorStatus
- EPIIntErrorClear
- GPIOPinConfigure
- GPIOPinTypeI2S
- GPIOPinTypeEthernetLED
- GPIOPinTypeUSBAnalog
- I2CSlaveIntClearEx
- I2CSlaveIntDisableEx
- I2CSlaveIntEnableEx
- I2CSlaveIntStatusEx
- I2SIntClear
- I2SIntDisable
- I2SIntEnable
- I2SIntStatus
- I2SMasterClockSelect
- I2SRxConfigSet
- I2SRxDataGet
- I2SRxDataGetNonBlocking
- I2SRxDisable
- I2SRxEnable
- I2SRxFIFOLevelGet
- I2SRxFIFOLimitGet
- I2SRxFIFOLimitSet
- I2STxConfigSet
- I2STxDataPut
- I2STxDataPutNonBlocking
- I2STxDisable
- I2STxEnable
- I2STxFIFOLevelGet
- I2STxFIFOLimitGet
- I2STxFIFOLimitSet
- I2STxRxConfigSet
- I2STxRxDisable
- I2STxRxEnable
- IntPendSet
- IntPendClear
- SSIBusy
- SysCtlDelay
- SysCtlI2SMClkSet
- UARTBusy
- UARTFIFODisable
- UARTFIFOEnable
- UARTRxErrorClear
- UARTRxErrorGet
- UARTTxIntModeGet
- UARTTxIntModeSet
- uDMAChannelSelectDefault

- uDMAChannelSelectSecondary
- USBDevEndpointConfigGet
- USBEndpointDataAvail
- USBEndpointDMACHannel
- USBEndpointDMADisable
- USBEndpointDMAEnable
- USBModeGet
- USBOTGHostRequest
- USBIntDisableControl
- USBIntEnableControl
- USBIntStatusControl
- USBIntDisableEndpoint
- USBIntEnableEndpoint
- USBIntStatusEndpoint
- USBHostMode

Workaround:

Code for these functions is included in the current version of StellarisWare, which can be downloaded from the website at http://www.ti.com/software_updates.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

3.3 ROM mapping check for the Boot loader does not function properly

Description:

Before the processor is released from the reset state, the System Control module is supposed to check offset 0x0000.0004 of Flash memory looking for a reset vector that is not 0xFFFF.FFFF. If an initialized reset vector is found, Flash memory is mapped to address 0x0000.0000, otherwise ROM is mapped to address 0x0000.0000. Currently, the System Control module errantly checks offset 0x0000.0008, which is the NMI vector. So, in situations where a valid reset vector (offset 0x0000.0004) has been programmed, but the NMI vector has not been programmed, the ROM is errantly mapped to zero preventing the application that is stored in Flash memory from being executed out of reset.

Workaround:

Ensure that the NMI vector is always programmed.

Silicon Revision Affected:

B1

Fixed:

Rev C3 implements the boot loader process outlined in the data sheet.

3.4 ROM_SSISetExpClk function is incorrect

Description:

If a non-Motorola format was specified in a call to the ROM_SSISetExpClk function, two lower bits of a clock divisor register could be corrupted. This corruption results in a small error in the actual clock rate.

Workaround:

Use the StellarisWare SSISetExpClk function in Flash memory.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

3.5 ROM_USBFIFOFlush function is incorrect

Description:

The ROM_USBFIFOFlush function improperly checks the state of the FIFO and does not allow the endpoint's FIFO to be flushed. This error affects all endpoints other than endpoint zero.

Workaround:

Use the StellarisWare USBFIFOFlush function in Flash memory.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

3.6 The option to force the ROM boot loader to execute at reset with an external pin does not function

Description:

The option to force the ROM boot loader to execute at reset with an external pin does not function. Changing the `PORT` and `PIN` fields of the **Boot Configuration (BOOTCFG)** register has no effect.

Workaround:

The ROM boot loader still executes if address 0x0000.0004 contains 0xFFFF.FFFF, indicating that the Flash memory has not been programmed.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C3.

4 Flash Memory

4.1 Cumulative page erases may introduce bit errors in Flash memory

Description:

Cumulative page erases anywhere in the Flash memory array may introduce bit errors. The bit error is not confined to the page being erased or the 4-KB block but could be in any page in the Flash memory. A page erase is used to erase a 1-KB page so it can be rewritten. A mass erase erases the entire Flash memory array (all pages). A bit error means that a bit may change from 0 to 1 or 1 to 0.

Workaround:

There are two possible workarounds for this issue:

1. Minimize total page erases to less than 3000 between mass erases for the lifetime of the product. After each mass erase, an additional 3000 page erase operations are allowed before bit errors may be introduced. At the rate of one page erase per week, this issue would not be seen over at least 17 years.
2. Perform CRC checks on all Flash memory after page erases to increase the chances of detecting the issue. The two CRC functions built into ROM can assist in this.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

4.2 Flash Write Buffer does not function above 50 MHz

Description:

The Flash Write Buffer does not successfully program the Flash memory at speeds above 50 MHz. In addition, the ROM boot loader cannot be used at speeds above 50 MHz because it uses the write buffer.

Workaround:

Lower the speed of the system clock to 50 MHz or less while programming the Flash memory or using the ROM boot loader.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

4.3 Flash memory endurance cycle specification is 100 cycles

Description:

The Flash memory endurance cycle specification (maximum program/erase cycles) is 100 cycles. Failure to adhere to the maximum number of program/erase cycles could result in corruption of the Flash memory contents and/or permanent damage to the device.

Workaround:

None. Because the failure mechanism is a function of the third-party Flash memory technology used in this device, there is no workaround. This third-party Flash memory technology is used only in the affected 130-nm Stellaris products and will not be used in any future devices. All other Stellaris products use Flash memory technology that exceeds industry quality and endurance cycle standards.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

5 μ DMA

5.1 The μ DMA controller fails to generate capture mode DMA requests from Timer A in the Timer modules

Description:

The μ DMA controller fails to generate DMA requests from Timer A in the General-Purpose Timer modules when in the Event Count and Event Time modes.

Workaround:

Use Timer B.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

5.2 The μ DMA does not generate a completion interrupt when transferring to and from GPTM 2A and 2B

Description:

The μ DMA module does not generate a completion interrupt on the Timer 2 interrupt vector when transferring data to and from Timers 2A and 2B. The μ DMA can successfully transfer data to and from Timers 2A and 2B; however, there is no interrupt to indicate that the transfer is complete.

Workaround:

If a completion interrupt is required, use an alternate GPTM.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

6 GPIO

6.1 Port B [1:0] pins require external pull-up resistors

Description:

The internal pull-up resistors are not effective for the Port B0 and B1 pins.

Workaround:

External pull-up resistors must be used on these two pins when they are used as GPIOs.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

6.2 Schmitt input feature does not function correctly

Description:

The Schmitt input on digital inputs may generate spurious transitions when connected to low slew-rate signal sources. If the input signal has a slew rate of less than $1\text{V}/\mu\text{s}$, a negative edge can generate several additional transitions into the microcontroller even though the input signal is still within the hysteresis band. Positive edges are not affected.

The additional transitions can cause anomalous operation in any peripherals or GPIOs that use digital inputs. Most at risk are peripherals that use pull-up resistors (I²C, GPIOs) or that typically involve slower signals (sensor inputs). This behavior can affect the noise immunity of digital inputs. As a result, arbitration may be lost during communication when the I²C module is the master.

Workaround:

Ensure that all signals connected to digital inputs have a slew rate of at least $1\text{V}/\mu\text{s}$. In some applications, reducing the resistance value of pull-up and pull-down resistors may be necessary. Note that R-C filters, such as low pass, on digital input signals should only be used if the slew-rate is still above $1\text{V}/\mu\text{s}$, or if additional transitions on the falling edge can be tolerated. Adding an external Schmitt-trigger circuit is a requirement for circuits where slow transitions are unavoidable and system noise levels are high.

Silicon Revision Affected:

B1

Fixed:

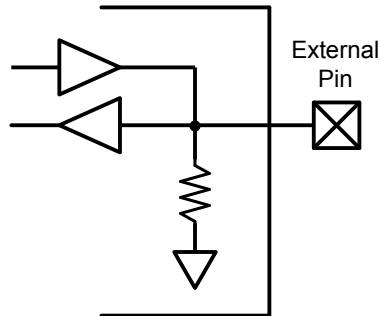
Fixed in Rev C3.

6.3 PB0 and PB1 have permanent internal pull-down resistance

Description:

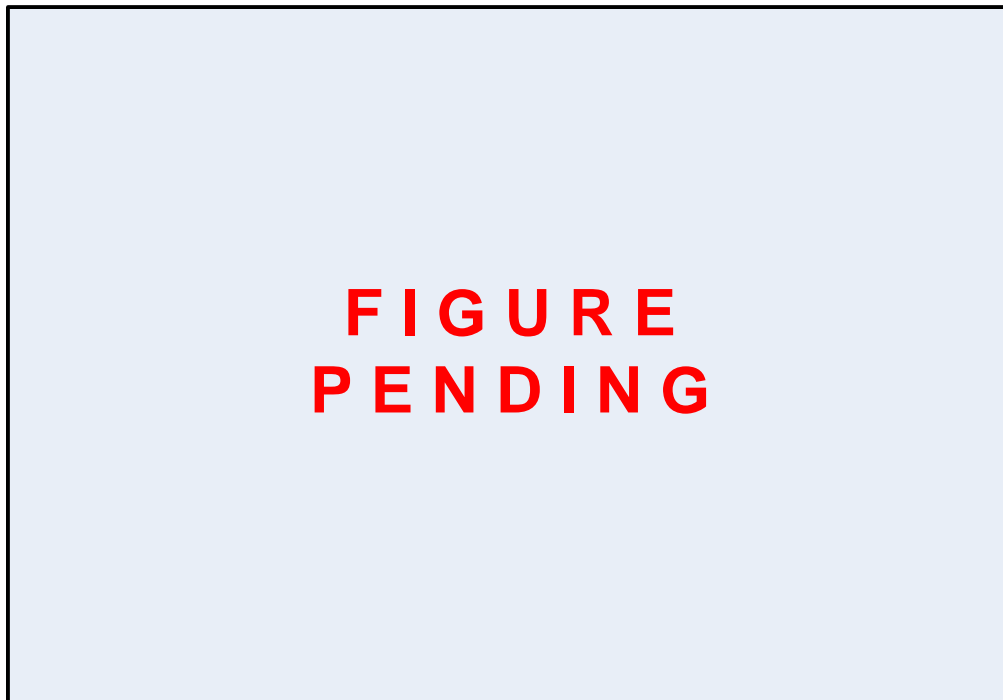
Regardless of their configuration, PB0 and PB1 have an internal pull-down resistance. The internal structure of these pins is shown in Figure 1 on page 15.

Figure 1. Internal Structure of PB0 and PB1

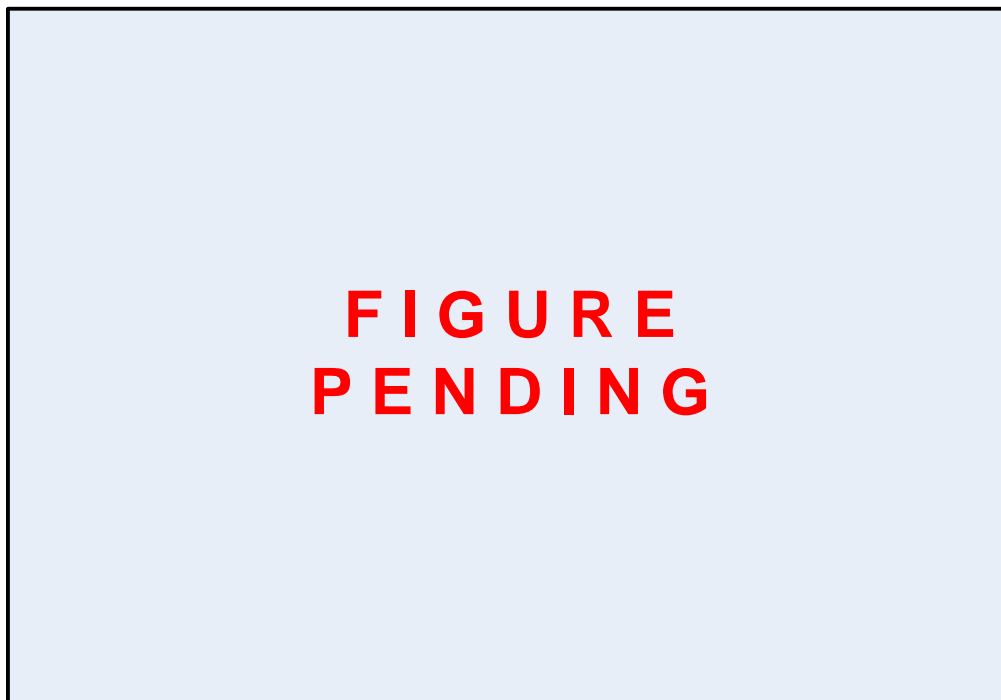


The characteristic of the pull-down for PB0 is shown in Figure 2 on page 15. The data labels for each data point show the external pull-up resistance in Ohms.

Figure 2. Voltage vs. Current for Various External Pull-up Resistors on PB0



The characteristic of the pull-down for PB1 is shown in Figure 3 on page 16. The data labels for each data point show the equivalent resistance in Ohms.

Figure 3. Voltage vs. Current for Various External Pull-up Resistors on PB1**Workaround:**

When either of these pins is configured as an input, the external circuit must provide enough drive strength to over-drive the internal pull-down and achieve the necessary V_{IH} voltage level. Analysis is ongoing to determine what value external resistors should have.

When either of these pins is configured as an output, the drive current needed to over-drive the internal pull-down resistance must be subtracted from the drive capabilities of the pin. In some applications, it may be necessary to select a higher drive strength (such as 4 mA instead of 2 mA) to achieve an acceptable output voltage on PB0.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

7 EPI

7.1 EPI dual-chip select function does not work

Description:

The Dual CSn Configuration mode ($CSCFG=0x2$) and the ALE with Dual CSn Configuration mode ($CSCFG=-x3$) controlled by the **EPI Host-Bus 8 Configuration 2 (EPIHB8CFG2)** register do not function. System designs should use ALE Configuration mode ($CSCFG=0x0$) or CSn Configuration mode ($CSCFG=0x1$).

Workaround:

None.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

7.2 EPI Host-Bus 16 mode does not work

Description:

The Host-Bus 16 mode (`MODE=0x3`) controlled by the **EPI Configuration (EPICFG)** register do not function.

Workaround:

None.

Silicon Revision Affected:

B1

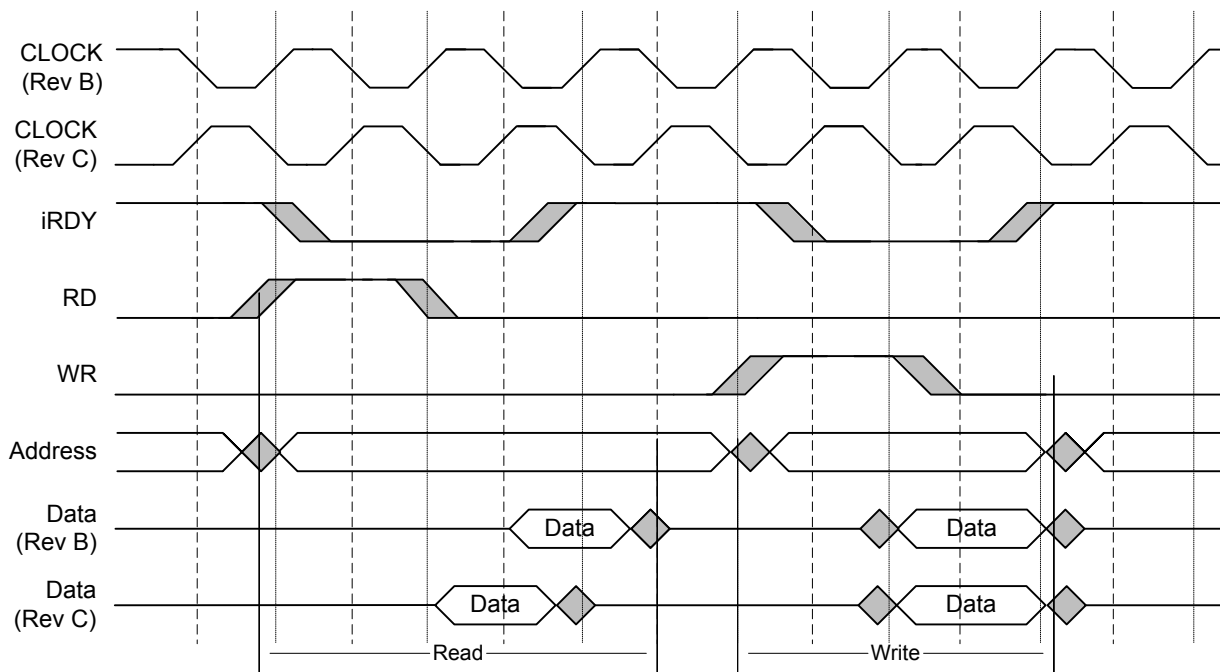
Fixed:

Fixed in Rev C.

7.3 Clock signal in EPI General-Purpose mode is inverted

Description:

The clock signal that is output on the EPI0S31 signal in General-Purpose mode is inverted. Figure 4 on page 18 shows the timing differences between Rev B parts and Rev C parts.

Figure 4. Timing Differences Between Rev B and Rev C Devices**Workaround:**

Use the opposite edge for timing when designing with this interface. During read cycles, ensure that the data meets set up and hold times for the appropriate edge as shown in the diagram above.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

8 General-Purpose Timers

8.1 The General-Purpose Timer match register does not function correctly in 32-bit mode

Description:

The **GPTM Timer A Match (GPTMTAMATCHR)** register triggers a match interrupt and a DMA request, if enabled, when the lower 16 bits match, regardless of the value of the upper 16 bits.

Workaround:

None.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

8.2 A spurious DMA request is generated when the timer rolls over in Input-Edge Time mode

Description:

When the timer is in Input-Edge Time mode and rolls over after the terminal count, a spurious DMA request is generated.

Workaround:

Either ignore the spurious interrupt, or capture the edge time into a buffer via DMA, then the spurious interrupt can be detected by noting that the captured value is the same as the previous capture value.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

8.3 A spurious DMA request is generated when the timer rolls over the 16-bit boundary

Description:

When the timer is in 32-bit periodic or one-shot mode and is enabled to generate periodic DMA requests, a spurious DMA request is generated when the timer rolls past 0x0000FFFF.

Workaround:

Only use DMA with a 16-bit periodic timer.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

8.4 The value of the prescaler register is not readable in Edge-Count mode

Description:

In Edge-Count mode, the prescaler is used as an 8-bit high order extension to the 16-bit counter. When reading the **GPTM Timer n (GPTMTnR)** register as a 32-bit value, the bits [23:16] always contain the initial value of the **GPTM Timer n Prescale (GPTMTnPR)** register, that is, the "load" value of the 8-bit extension.

Workaround:

None.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

8.5 ADC trigger and Wait-on-Trigger may assert when the timer is disabled

Description:

If the value in the **GPTM Timer n Match (GPTMTnMATCHR)** register is equal to the value of the timer counter and the **TnOTE** bit in the **GPTM Control (GPTMCTL)** register is set, enabling the ADC trigger, the trigger fires even when the timer is disabled (the **TnEN** bit in the **GPTMCTL** register is clear). Similarly, if the value in the **GPTMTnMATCHR** register is equal to the value of the timer counter and the **TnWOT** bit in the **GPTM Timer n Mode (GPTMTnMR)** register is set, enabling the Wait-on-Trigger mode, the trigger fires even when the timer is disabled.

Workaround:

Enable the timer before setting the **TnOTE** bit. Also, for the Wait-on-Trigger mode, ensure that the timers are configured in the order in which they will be triggered.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

8.6 Wait-on-Trigger does not assert unless the TnOTE bit is set

Description:

Wait-on-Trigger does not assert unless the **TnOTE** bit is set in the **GPTMCTL** register.

Workaround:

If the **TnWOT** bit in the **GPTM Timer n Mode (GPTMTnMR)** register is set, enabling the Wait-on-Trigger mode, the **TnOTE** bit must also be set in the **GPTMCTL** register in order for the Wait-on-Trigger to fire. Note that when the **TnOTE** bit is set, the ADC trigger is also enabled.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

8.7 Do not enable match and timeout interrupts in 16-bit PWM mode

Description:

16-bit PWM mode generates match and timeout interrupts in the same manner as periodic mode.

Workaround:

Ensure that any unwanted interrupts are masked in the **GPTMTnMR** and **GPTMIMR** registers.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

8.8 Do not use μ DMA with 16-bit PWM mode

Description:

16-bit PWM mode generates match and timeout μ DMA triggers in the same manner as periodic mode.

Workaround:

Do not use μ DMA to transfer data when the timer is in 16-bit PWM mode.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

8.9 Writing the GPTMTnV register does not change the timer value when counting up

Description:

When counting up, writes to the **GPTM Timer n Value (GPTMTnV)** register do not change the timer value.

Workaround:

None.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

8.10 The prescaler does not work correctly when counting up in periodic or one-shot mode

Description:

When counting up, the prescaler does not work correctly in 16-bit periodic or snap-shot mode.

Workaround:

Do not use the prescaler when counting up in 16-bit periodic or snap-shot mode.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

8.11 Snapshot must be enabled in both Timer A and B when in 32-bit snapshot mode

Description:

When a periodic snapshot occurs in 32-bit periodic mode, only the lower 16-bit are stored into the **GPTM Timer A (GPTMTAR)** register.

Workaround:

If both the **TASNAPS** and **TBSNAPS** bits are set in the **GPTM Timer A Mode (GPTMTAMR)** register, the entire 32-bit snapshot value is stored in the **GPTMTAR** register.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

9 Watchdog Timers

9.1 Writes to Watchdog Timer 1 module WDTLOAD register sometimes fail

Description:

Due to the independent clock domain of the Watchdog Timer 1 module, writes to the **Watchdog Load (WDTLOAD)** register may sometimes fail, even though the **WRC** bit in the **WDTCTL1** register is set after the write occurs.

Workaround:

After performing a write to the **WDTLOAD** register, read the contents back and verify that they are correct. If they are incorrect, perform the write operation again.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

10 ADC

10.1 ADC hardware averaging produces erroneous results in differential mode

Description:

The implementation of the ADC averaging circuit does not work correctly when the ADC is sampling in differential mode and the difference between the voltages is approximately 0.0V.

Workaround:

Do not use hardware averaging in differential mode. Instead, use the FIFO to store results and average them in software.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

10.2 The ADCSPC register does not function

Description:

The **ADC Sample Phase Control (ADCSPC)** register does not function and cannot be used.

Workaround:

None.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

10.3 Differential pair encodings are incorrect

Description:

When using differential mode, the **MUX_n** fields in the **ADCSSMUX_n** registers should be configured to be "i" where the paired inputs are "2i" and "2i + 1". This encoding does not work for AIN8 - AIN15.

Workaround:

Use the encodings shown in the following table:

Adjacent Channels	i	MUX _n Encoding
AIN0 and AIN1	0	0x0
AIN2 and AIN3	1	0x1
AIN4 and AIN5	2	0x2
AIN6 and AIN7	3	0x3
AIN8 and AIN9	4	0x8

Adjacent Channels	i	MUXn Encoding
AIN10 and AIN11	5	0x9
AIN12 and AIN13	6	0xA
AIN14 and AIN15	7	0xB

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

11 UART

11.1 UART Smart Card (ISO 7816) mode does not function

Description:

The `UnTX` signal does not function correctly as the bit clock in Smart Card mode.

Workaround:

None.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

11.2 When in IrDA mode, the `UnRx` signal requires configuration even if not used

Description:

When in IrDA mode, the transmitter may not function correctly if the `UnRx` signal is not used.

Workaround:

When in IrDA mode, if the application does not require the use of the `UnRx` signal, the GPIO pin that has the `UnRx` signal as an alternate function must be configured as the `UnRx` signal and pulmac High.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

11.3 Phantom interrupts occur in Smart Card mode

Description:

In Smart Card mode, after receiving a valid TX interrupt, phantom parity error interrupts occur, even though all **UARTRIS** and **UARTMIS** bits are clear.

Workaround:

Make sure to always clear the parity error interrupt in the interrupt handler, even when the **PERIS** and **PEMIS** bits are clear.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C3.

11.4 The RTRIS bit in the UARTRIS register is only set when the interrupt is enabled

Description:

The **RTRIS** (UART Receive Time-Out Raw Interrupt Status) bit in the **UART Raw Interrupt Status (UARTRIS)** register should be set when a receive time out occurs, regardless of the state of the **RTIM** enable bit in the **UART Interrupt Mask (UARTIM)** register. However, currently the **RTIM** bit must be set in order for the **RTRIS** bit to be set when a receive time out occurs.

Workaround:

For applications that require polled operation, the **RTIM** bit can be set while the UART interrupt is disabled in the NVIC using the `IntDisable(n)` function in the StellarisWare Peripheral Driver Library, where *n* is 21, 22, or 49 depending on whether UART0, UART1 or UART2 is used. With this configuration, software can poll the **RTRIS** bit, but the interrupt is not reported to the NVIC.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

12 SSI

12.1 An interrupt is not generated when using μ DMA with the SSI module if the EOT bit is set

Description:

When using the primary μ DMA channels with the SSI module, an interrupt is not generated on transmit μ DMA completion if the **EOT** bit (bit 4 of the **SSICR1** register) is enabled.

Workaround:

Use the alternate μ DMA channels for the SSI module.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

13 I2S

13.1 Some bits in the I2SMCLKCFG register do not function

Description:

The top 2 bits of the `RXI` and `TXI` bit fields in the **I2SMCLKCFG** register do not function (bits [29:28] of `RXI` and bits [13:12] of `TXI`). The `RXI` and `TXI` fields contain the 10-bit integer input for the receive and transmit clock generator, respectively. The remaining 8 bits in each field function correctly, so most of the possible integer input choices can be used in system design.

Workaround:

None.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

13.2 I²S SCLK signal is inverted in certain modes

Description:

When the I²S controller is operating as a receiver in SCLK Master mode, the WS signal is latched on the rising edge of SCLK, not the falling edge. In addition, when the controller is operating as a transmitter in SCLK Slave mode, the data is launched on the rising edge of SCLK, not the falling edge.

Workaround:

For the transmitter, there are two possible workarounds for this issue:

1. Ensure that the `I2S0TXSCK` signal leads the `I2S0TXWS` signal by at least 4 ns.
2. Configure as I²S mode with DAC in Left-Justified audio format.

For the receiver, ensure that the CODEC is configured as the SCLK master, and the I²S receive module is configured as the SCLK slave.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

14 Ethernet Controller

14.1 Ethernet receive packet corruption may occur when using optional auto-clock gating

Description:

Ethernet receive packets may be corrupted if the `ACG` bit in the **Run-Mode Clock Configuration (RCC)** register is set.

Workaround:

Do not set the `ACG` bit in the **RCC** register.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

14.2 Ethernet packet loss with cables longer than 50 meters

Description:

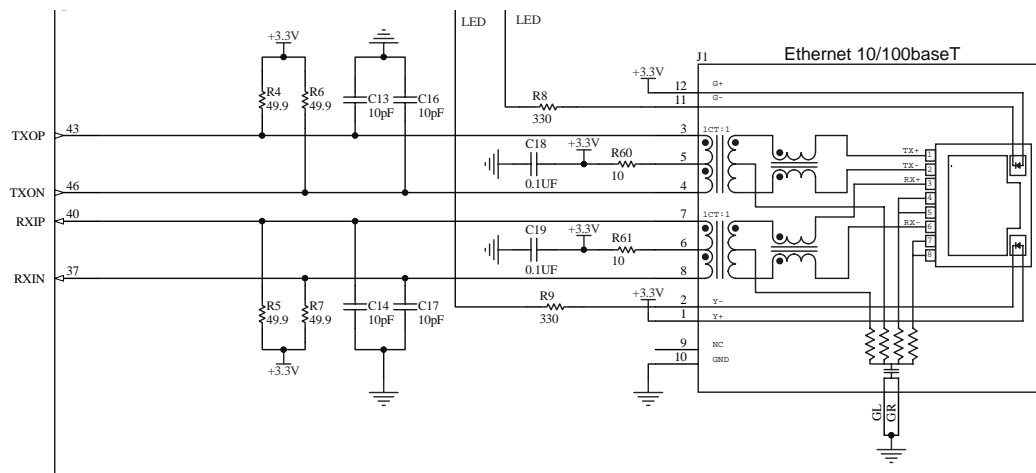
The microcontroller experiences some packet loss with Ethernet cables longer than 50 meters in normal operating conditions.

Workaround:

There are two possible workarounds for this issue:

1. Add 10 Ω resistor to the center-tap of the transformer as shown in the figure. These resistors should be replaced by a direct connection for silicon that has this item fixed.
2. Continue using the recommended circuit, but limit cable lengths to 50 meters.

Figure 5. Recommended Center-Tap Connections



Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

14.3 Ethernet PHY interrupts do not function correctly

Description:

The Ethernet PHY interrupts are not functional. Ethernet PHY interrupts are not necessary for normal Ethernet operation. MAC interrupts are all functional and provide necessary operation.

Workaround:

None.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

14.4 Encoding error in the Ethernet MAC LED Encoding (MACLED) register

Description:

Configuring the LED0 or LED1 field of the **Ethernet MAC LED Encoding (MACLED)** register to 0x8 should cause the corresponding LED to report a combined link + activity status. However, it instead only reports activity status (i.e. exactly the same as encoding 0x1).

Workaround:

None.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

15 USB

15.1 USB0ID and USB0VBUS signals are required to be connected regardless of mode

Description:

The DEVMODOTG bit in the **USB General-Purpose Control and Status (USBGPCS)** register does not function correctly.

Workaround:

Connect the `USB0VBUS` input to `VBUS` in all modes. In addition, connect the `USB0ID` pin to ground for Host mode operation and to `VDD` for Device mode operation using the `DEVMOD` bit in the **USB General-Purpose Control and Status (USBGPCS)** register.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

15.2 Latch-up may occur if power is applied to the `VBUS` pin but not to `VDD`

Description:

If power is applied to the `VBUS` pin but not to `VDD`, the microcontroller may latch up and or draw excessive current. This condition can occur if the microcontroller is unpowered and is connected as a USB device or OTG B.

Workaround:

Add a 100 Ω resistor (the tolerance is not critical) in series with the microcontroller's `USB0VBUS` signal. This resistor changes the USB `VBUS` signalling thresholds by approximately 8 mV which addresses the latch-up issue with no impact on USB performance.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

15.3 USB compliance test issue: USB full-speed, far-end signal compliance tests fail with 5 m cable

Description:

While USB packet loss has not been observed, the device is unable to pass the following USB compliance tests:

- USB Host Test B.3.3.2 Full-speed Downstream Signal Quality Test
- USB Device Test B.6.3.1 Signal Integrity Test – Upstream Signal test (full speed)

Compliance testing is based on the "USB Implementers Forum Full and Low Speed Electrical and Interoperability Compliance Test Procedure" Revision 1.3 available from usb.org website. The compliance testing is performed using a 5 m USB certified cable between the host or device under test and the test SQiDD which is then connected to a USB compliant hub chain to the root hub. Under compliance test conditions, the rising edges of the USB `D+/D-` signals begin to violate the lower right corner of the full-speed eye diagram defined by the USB specification. USB certification cannot be obtained because of this erratum.

A full report on this issue, "USB Far End Signal Integrity Test Results," is available from your local TI FAE.

Workaround:

If a cable with a length of 1 m is used instead of a 5 m cable, the Eye diagram compliance tests all pass with adequate margin across the voltage and temperature range of the part. Under nominal voltage and temperature conditions, a cable of up to 3 m can be used and passes the eye diagram compliance tests.

Silicon Revision Affected:

B1, C3

Fixed:

Fixed in Rev C5.

15.4 USB compliance test issue: USB embedded host low-speed, far-end signal compliance tests fail

Description:

While USB packet loss has not been observed, the device is unable to pass the following USB compliance test:

- USB Host Test B.3.3.1 Low-Speed Downstream Signal Quality Test

USB Compliance testing is based on the "USB Implementers Forum Full and Low Speed Electrical and Interoperability Compliance Test Procedure" Revision 1.3 available from usb.org website. The rising and falling edges of the USB D+/D- signals violate the lower half of the low-speed eye diagram defined by the USB specification. This erratum applies only to systems defined as a USB embedded host that support low-speed devices. USB embedded host and OTG systems that support full-speed devices only are not affected by this erratum. USB device systems are full-speed only and thus are not affected by this erratum.

A full report on this issue, "USB Far End Signal Integrity Test Results," is available from your local TI FAE.

Workaround:

None.

Silicon Revision Affected:

B1, C3

Fixed:

Fixed in Rev C5.

16 PWM

16.1 PWM generation is incorrect with extreme duty cycles

Description:

If a PWM generator is configured for Count-Up/Down mode, and the **PWM Load (PWMnLOAD)** register is set to a value N, setting the compare to a value of 1 or N-1 results in steady state signals instead of a PWM signal. For example, if the user configures PWM0 as follows:

- PWMENABLE = 0x00000001

- PWM0 Enabled
- PWM0CTL = 0x00000007
 - Debug mode enabled
 - Count-Up/Down mode
 - Generator enabled
- PWM0LOAD = 0x00000063
 - Load is 99 (decimal), so in Count-Up/Down mode the counter counts from zero to 99 and back down to zero (200 clocks per period)
- PWM0GENA = 0x000000b0
 - Output High when the counter matches comparator A while counting up
 - Output Low when the counter matches comparator A while counting down
- PWM0DBCTL = 0x00000000
 - Dead-band generator is disabled

If the **PWM0 Compare A (PWM0CMPA)** value is set to 0x00000062 (N-1), PWM0 should output a 2-clock-cycle long High pulse. Instead, the PWM0 output is a constant High value.

If the **PWM0CMPA** value is set to 0x00000001, PWM0 should output a 2-clock-cycle long negative (Low) pulse. Instead, the PWM0 output is a constant Low value.

Workaround:

User software must ensure that when using the PWM Count-Up/Down mode, the compare values must never be 1 or the **PWMnLOAD** value minus one (N-1).

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

16.2 Sync of PWM does not trigger "zero" action

Description:

If the **PWM Generator Control (PWM0GENA)** register has the **ActZero** field set to 0x2, then the output is set to 0 when the counter reaches 0, as expected. However, if the counter is cleared by setting the appropriate bit in the **PWM Time Base Sync (PWMSYNC)** register, then the "zero" action is not triggered, and the output is not set to 0.

Workaround:

None.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

16.3 PWM "zero" action occurs when the PWM module is disabled**Description:**

The zero pulse may be asserted when the PWM module is disabled.

Workaround:

None.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

16.4 PWM Enable Update register bits do not function**Description:**

The `ENUPDn` bits in the **PWM Enable Update (PWMENUPD)** register do not function. As a result, enabling the PWM modules can't be synchronized.

Workaround:

None.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

16.5 PWM fault latch does not operate correctly**Description:**

If the `LATCH` bit is set in the **PWMnCTL** register, the PWM fault condition should be latched until the `INTFAULTn` bit in the **PWMISC** register is cleared. However, the PWM fault signal is not correctly latched and the PWM resumes programmed signalling after the fault condition is removed, regardless of whether the `INTFAULTn` bit is cleared.

Workaround:

Software can effectively address this issue with the addition of a few register writes in the ISR.

1. The **PWMnMINFLTPER** register can be used to ensure that the fault is asserted for a long enough period such that the ISR can be called to implement the workaround.
2. The PWM output can be disabled manually using the `PWMnEN` bit in the **PWMENABLE** register.
3. Software can perform computations to determine if the PWM can be restarted.

4. The `INTFAULTn` bit in the **PWMISC** is cleared by writing a 1 to it.
5. The PWM output can be manually re-enabled using the `PWMnEN` bit in the **PWMENABLE** register.

Note that when using this workaround, the PWM output is disabled manually, which means it does not go to the "pre-programmed" state from various fault registers but instead goes to 0.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

17 Electrical Characteristics

17.1 Momentarily exceeding V_{IN} ratings on any pin can cause latch-up

Description:

To avoid latch-up, the maximum DC ratings of the part must be strictly enforced. The most common violation of the V_{IN} electrical specification can occur when a mechanical switch or contact is connected directly to a GPIO or special function (`RST`, `WAKE`, ...) pin. The circuit shown in Figure 6 on page 33 typically has stray inductance and capacitance that can cause a voltage glitch when the switch transitions, as shown in Figure 7 on page 33. The magnitude of the glitch may exceed the V_{IN} in the maximum DC ratings table in the Electrical Characteristics chapter. Figure 8 on page 34 shows an improved circuit that eliminates the glitch.

Figure 6. Incorrect Reset Circuitry

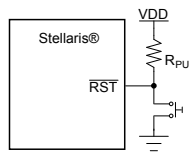
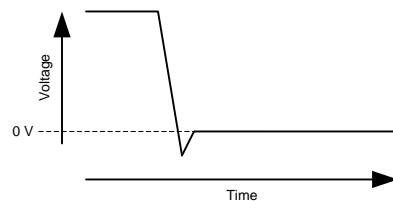


Figure 7. Excessive Undershoot Voltage on Reset

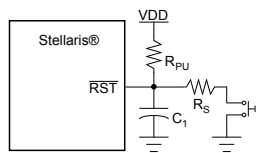


Workaround:

Use a circuit as shown in Figure 8 on page 34. In this circuit, R_S should be less than or equal to $R_{PU}/10$. C_1 should be matched to R_{PU} to achieve a suitable t_{RC} for the application. Typical values are:

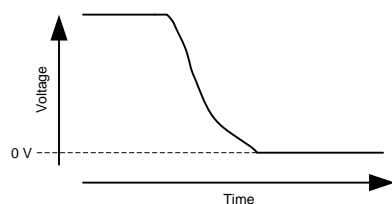
- $R_{PU} = 10\text{ k}\Omega$
- $R_S = 470\ \Omega$
- $C_1 = 0.01\ \mu\text{F}$

Figure 8. Recommended Reset Circuitry



After implementing the circuit shown in Figure 8 on page 34, confirm that the voltage on the $\overline{\text{RST}}$ input has a curve similar to the one in Figure 9 on page 34, and that the V_{IN} specification is not exceeded.

Figure 9. Recommended Voltage on Reset



Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

17.2 Power-on event may disrupt operation

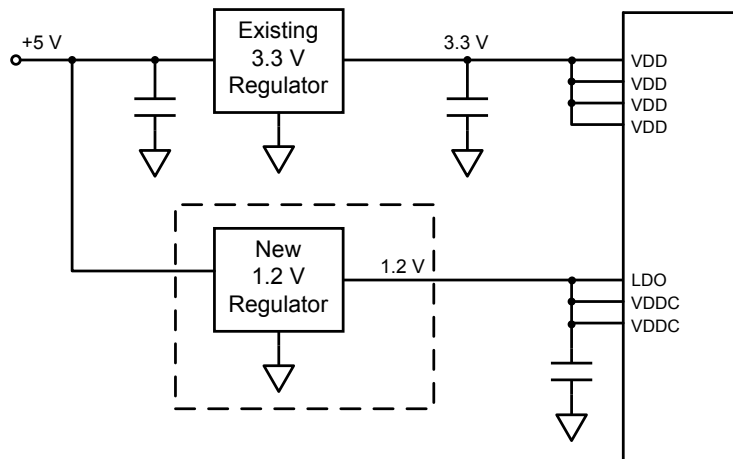
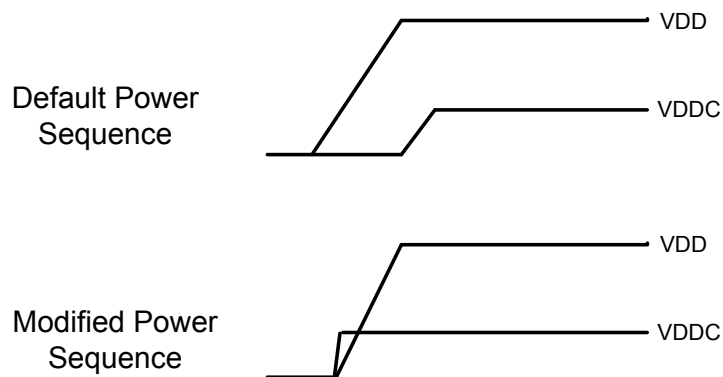
Description:

Incorrect power sequencing during power up can disrupt operation and potentially cause device failure.

Workaround:

V_{DDC} must be applied approximately 50 μs before V_{DD} . Normally V_{DDC} is controlled by the part's internal LDO voltage regulator. The workaround requires the addition of an external regulator (see Figure 10) to ensure that V_{DDC} sequencing requirements are met (see Figure 11). A recommended regulator is the TI TPS73101DBVR.

This fix mitigates the on-chip power issue, but does not solve it completely. During development, the Flash memory should also be reprogrammed (using LMFlash or another programming tool) at least once a week.

Figure 10. Configuration of External Regulator**Figure 11. VDDC Sequencing Requirements**

Detailed characterization is ongoing. Contact the Applications Support Team for the latest information.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

Copyright © 2008-2011 Texas Instruments Incorporated All rights reserved. Stellaris and StellarisWare are registered trademarks of Texas Instruments Incorporated. ARM and Thumb are registered trademarks and Cortex is a trademark of ARM Limited. Other names and brands may be claimed as the property of others.

Texas Instruments Incorporated
108 Wild Basin, Suite 350
Austin, TX 78746

<http://www.ti.com/stellaris>

<http://www-k.ext.ti.com/sc/technical-support/product-information-centers.htm>



**TEXAS
INSTRUMENTS**



Cortex
Intelligent Processors by ARM®