

## 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.

| Date       | Revision | Description   |
|------------|----------|---|
| July 2010  | 2.8      | <ul style="list-style-type: none"> <li>Added issue "The RTRIS bit in the UARTRIS register is only set when the interrupt is enabled" on page 20.</li> </ul>   |
| June 2010  | 2.7      | <ul style="list-style-type: none"> <li>Minor edits.</li> </ul>  |
| April 2010 | 2.6      | <ul style="list-style-type: none"> <li>Based on further examination of the "I<sup>2</sup>C 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 12.</li> <li>Added issue "Encoding error in the Ethernet MAC LED Encoding (MACLED) register" on page 23.</li> <li>Added information about items fixed on Rev C3.</li> </ul>  |
| March 2010 | 2.5      | <ul style="list-style-type: none"> <li>Added issue "The prescaler does not work correctly when counting up in periodic or one-shot mode" on page 17.</li> <li>Added issue "Snapshot must be enabled in both Timer A and B when in 32-bit snapshot mode" on page 17.</li> <li>Added issue "Phantom interrupts occur in Smart Card mode" on page 19.</li> <li>Added issue "I<sup>2</sup>C arbitration may be lost when operating as a master".</li> </ul>   |
| Mar 2010   | 2.4      | <ul style="list-style-type: none"> <li>Added issue "The option to force the ROM boot loader to execute at reset with an external pin does not function" on page 11.</li> <li>Amended the workaround for issue "A spurious DMA request is generated when the timer rolls over in Input-Edge Time mode" on page 14.</li> <li>Reworded description of issue "The value of the prescaler register is not readable in Edge-Count mode" on page 15.</li> <li>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.</li> <li>Added issue "ADC trigger and Wait-on-Trigger may assert when the timer is disabled" on page 15.</li> <li>Added issue "Wait-on-Trigger does not assert unless the TnOTE bit is set" on page 16.</li> <li>Added issue "Do not enable match and timeout interrupts in 16-bit PWM mode" on page 16.</li> <li>Added issue "Do not use <math>\mu</math>DMA with 16-bit PWM mode" on page 16.</li> <li>Added issue "Writing the GPTMTnV register does not change the timer value when counting up" on page 17.</li> </ul> |

| Date     | Revision | Description  |
|----------|----------|--|
| Feb 2010 | 2.3      | <ul style="list-style-type: none"> <li>Added issue "A spurious DMA request is generated when the timer rolls over the 16-bit boundary" on page 15.</li> <li>Added issue "The value of the prescaler register is not readable in Edge-Count mode" on page 15.</li> <li>Added issue "Prescaler register must have a non-zero value in 16-bit Edge-Time mode."</li> <li>Added issue "The ADCSPC register does not function" on page 18.</li> </ul>  |
| Jan 2010 | 2.2      | <ul style="list-style-type: none"> <li>Modified description for "The General-Purpose Timer match register does not function correctly in 32-bit mode" on page 14 to include DMA operation.</li> <li>Added issue "A spurious DMA request is generated when the timer rolls over in Input-Edge Time mode" on page 14.</li> <li>Changed workaround for "Latch-up may occur if power is applied to the VBUS pin but not to VDD" on page 24 and changed status to "Fixed in Rev C."</li> </ul>  |
| Dec 2009 | 2.1      | <ul style="list-style-type: none"> <li>The status of "The Recover Locked Device sequence does not work as expected" on page 4 has been changed to "Fixed in Rev C."</li> <li>"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.</li> <li>Added additional APIs to "Some ROM functions are unsupported" on page 7.</li> <li>"The <math>\mu</math>DMA controller fails to generate capture mode DMA requests from Timer A in the Timer modules" on page 11 has been added.</li> <li>"Ethernet packet count decremented before the FCS is read" has been removed and the content added to the LM3S9B92 data sheet.</li> <li>The status of "Latch-up may occur if power is applied to the VBUS pin but not to VDD" on page 24 has been changed to "Not fixed in Rev C."</li> </ul> |
| Nov 2009 | 2.0      | Started tracking revision history.   |

| Erratum Number | Erratum Title  | Revision(s) Affected |
|----------------|--|----------------------|
| 1.1            | JTAG INTEST instruction does not work  | B1                   |
| 1.2            | The Recover Locked Device sequence does not work as expected                           | B1                   |
| 2.1            | Sleep and Deep-Sleep mode not usable at higher speeds when ISRs reside in Flash memory | B1                   |
| 2.2            | Device Capabilities registers may not accurately reflect available signals             | B1                   |
| 2.3            | The PIOSC is not trimmed by the factory  | B1                   |
| 3.1            | Cumulative page erases may introduce bit errors in Flash memory                        | B1                   |
| 3.2            | Flash Write Buffer does not function above 50 MHz                                      | B1                   |
| 4.1            | Ethernet fails to connect when using the Boot Loader software in ROM                   | B1                   |
| 4.2            | Some ROM functions are unsupported   | B1                   |
| 4.3            | ROM mapping check for the Boot loader does not function properly                       | B1                   |

| Erratum Number | Erratum Title  | Revision(s) Affected |
|----------------|--|----------------------|
| 4.4            | ROM_I2CMasterErr function is incorrect   | B1                   |
| 4.5            | ROM_SSIConfigSetExpClk function is incorrect   | B1                   |
| 4.6            | ROM_USBFIFOFlush function is incorrect   | B1                   |
| 4.7            | The option to force the ROM boot loader to execute at reset with an external pin does not function     | B1, C1               |
| 5.1            | The $\mu$ DMA controller fails to generate capture mode DMA requests from Timer A in the Timer modules | B1, C1, C3           |
| 6.1            | Port B [1:0] pins require external pull-up resistors   | B1                   |
| 6.2            | Schmitt input feature does not function correctly  | B1, C1               |
| 7.1            | EPI dual-chip select function does not work  | B1                   |
| 7.2            | EPI Host-Bus 16 mode does not work   | B1                   |
| 7.3            | Clock signal in EPI General-Purpose mode is inverted   | B1                   |
| 8.1            | The General-Purpose Timer match register does not function correctly in 32-bit mode                    | B1, C1, C3           |
| 8.2            | A spurious DMA request is generated when the timer rolls over in Input-Edge Time mode                  | B1, C1, C3           |
| 8.3            | A spurious DMA request is generated when the timer rolls over the 16-bit boundary                      | B1, C1, C3           |
| 8.4            | The value of the prescaler register is not readable in Edge-Count mode                                 | B1, C1, C3           |
| 8.5            | ADC trigger and Wait-on-Trigger may assert when the timer is disabled                                  | B1, C1, C3           |
| 8.6            | Wait-on-Trigger does not assert unless the TnOTE bit is set  | B1, C1, C3           |
| 8.7            | Do not enable match and timeout interrupts in 16-bit PWM mode  | B1, C1, C3           |
| 8.8            | Do not use $\mu$ DMA with 16-bit PWM mode  | B1, C1, C3           |
| 8.9            | Writing the GPTMTnV register does not change the timer value when counting up                          | B1, C1, C3           |
| 8.10           | The prescaler does not work correctly when counting up in periodic or one-shot mode                    | B1, C1, C3           |
| 8.11           | Snapshot must be enabled in both Timer A and B when in 32-bit snapshot mode                            | B1, C1, C3           |
| 9.1            | Writes to Watchdog Timer 1 module WDTLOAD register sometimes fail                                      | B1, C1, C3           |
| 10.1           | ADC hardware averaging produces erroneous results in differential mode                                 | B1, C1, C3           |
| 10.2           | The ADCSPC register does not function  | B1                   |
| 11.1           | UART Smart Card (ISO 7816) mode does not function  | B1                   |
| 11.2           | When in IrDA mode, the UnRx signal requires configuration even if not used                             | B1                   |
| 11.3           | Phantom interrupts occur in Smart Card mode  | B1, C1               |
| 11.4           | The RTRIS bit in the UARTRIS register is only set when the interrupt is enabled                        | B1, C1, C3           |
| 12.1           | An interrupt is not generated when using $\mu$ DMA with the SSI module if the EOT bit is set           | B1                   |
| 13.1           | Some bits in the I2SMCLKCFG register do not function   | B1                   |
| 13.2           | I <sup>2</sup> S SCLK signal is inverted in certain modes  | B1                   |
| 14.1           | Ethernet receive packet corruption may occur when using optional auto-clock gating                     | B1                   |

| Erratum Number | Erratum Title   | Revision(s) Affected |
|----------------|---|----------------------|
| 14.2           | Ethernet packet loss with cables longer than 50 meters                      | B1                   |
| 14.3           | Ethernet PHY interrupts do not function correctly                           | B1                   |
| 14.4           | Encoding error in the Ethernet MAC LED Encoding (MACLED) register           | B1, C1, C3           |
| 15.1           | USB0ID and USB0VBUS signals are required to be connected regardless of mode | B1                   |
| 15.2           | Latch-up may occur if power is applied to the VBUS pin but not to VDD       | B1                   |
| 16.1           | PWM generation is incorrect with extreme duty cycles                        | B1                   |
| 16.2           | Sync of PWM does not trigger "zero" action                                  | B1                   |
| 16.3           | PWM "zero" action occurs when the PWM module is disabled                    | B1                   |
| 16.4           | PWM Enable Update register bits do not function                             | B1                   |
| 17.1           | Momentarily exceeding $V_{IN}$ ratings on any pin can cause latch-up        | B1                   |
| 17.2           | Power-on event may disrupt operation  | 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 C1.

## 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 C1.

## 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 C1.

### 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 C1.

## 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.

**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.

**Silicon Revision Affected:**

B1

**Fixed:**

Fixed for devices with date codes beginning 0946.

## 3 Internal Memory

### 3.1 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 C1.

## 3.2 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.

**Workaround:**

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

**Silicon Revision Affected:**

B1

**Fixed:**

Fixed in Rev C1.

## 4 ROM

### 4.1 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 C1.

### 4.2 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
- EPIConfigGPModeSet
- 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](http://www.ti.com/software_updates).

**Silicon Revision Affected:**

B1

**Fixed:**

Fixed in Rev C1.

## 4.3 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:**

ROM boot loader does not function in Rev C1. Address 0x0000.0000 is checked in Rev C3.

## 4.4 ROM\_I2CMasterErr function is incorrect

**Description:**

The ROM\_I2CMasterErr function currently assumes that bit 2 of the **I2CMCS** register is set in all error conditions and, if this bit is clear, assumes no error has occurred. Unfortunately, this bit only indicates an ACK error so the function returns I2C\_MASTER\_ERR\_NONE if the controller loses arbitration. I2C\_MASTER\_ERR\_ARB\_LOST is expected in this case.

**Workaround:**

Use the StellarisWare I2CMasterErr function in Flash memory.

**Silicon Revision Affected:**

B1

**Fixed:**

Fixed in Rev C1.

## 4.5 ROM\_SSIConfigSetExpClk function is incorrect

**Description:**

If a non-Motorola format was specified in a call to the ROM\_SSIConfigSetExpClk 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 SSIConfigSetExpClk function in Flash memory.

**Silicon Revision Affected:**

B1

**Fixed:**

Fixed in Rev C1.

## 4.6 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 C1.

## 4.7 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, C1

**Fixed:**

Fixed in Rev C3.

## 5 $\mu$ DMA

### 5.1 The $\mu$ DMA controller fails to generate capture mode DMA requests from Timer A in the Timer modules

**Description:**

The  $\mu$ 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, C1, C3

**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 C1.

## 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 1V/μs, 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 (I<sup>2</sup>C, 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 I<sup>2</sup>C module is the master.

**Workaround:**

Ensure that all signals connected to digital inputs have a slew rate of at least 1V/μs. 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 1V/μs, 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, C1

**Fixed:**

Fixed in Rev C3.

## 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 C1.

## 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

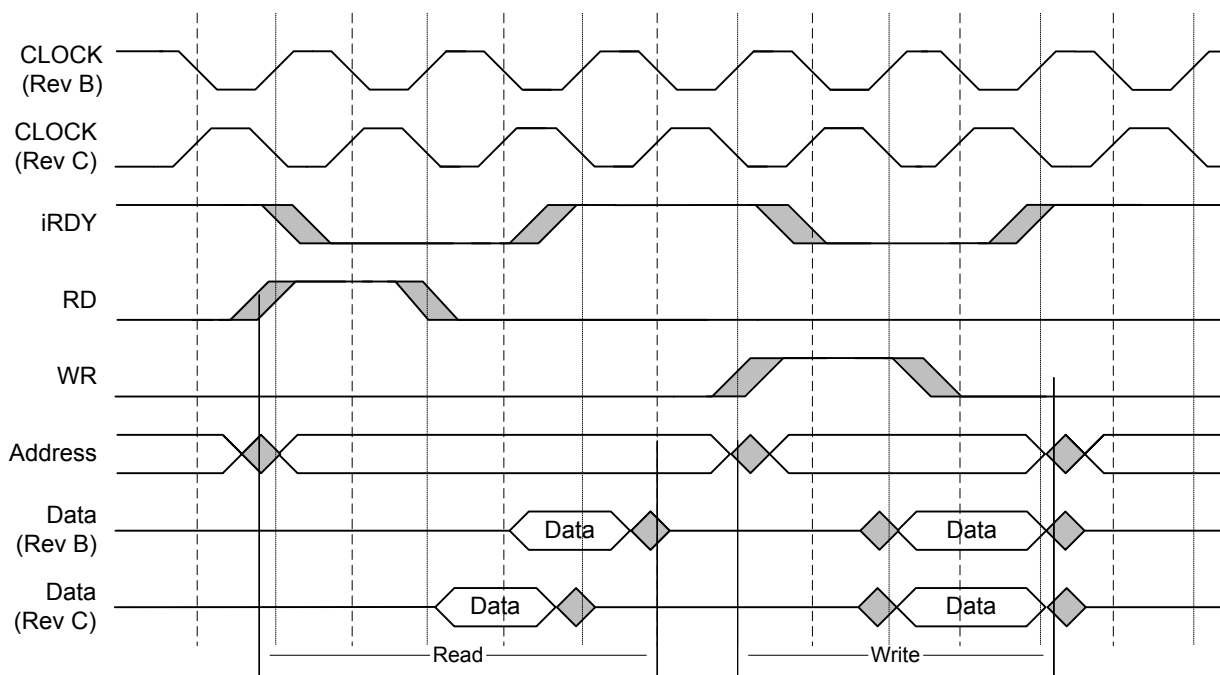
**Fixed:**

Fixed in Rev C1.

## 7.3 Clock signal in EPI General-Purpose mode is inverted

**Description:**

The clock signal that is output on the EPIO31 signal in General-Purpose mode is inverted. Figure 1 on page 13 shows the timing differences between Rev B parts and Rev C parts.

**Figure 1. 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 C1.

## 8 General-Purpose Timer

### 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, C1, C3

**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, C1, C3

**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, C1, C3

**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, C1, C3

**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, C1, C3

**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, C1, C3

**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, C1, C3

**Fixed:**

Not yet fixed.

## 8.8 Do not use $\mu$ DMA with 16-bit PWM mode

**Description:**

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

**Workaround:**

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

**Silicon Revision Affected:**

B1, C1, C3



**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, C1, C3

**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, C1, C3

**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, C1, C3

**Fixed:**

Not yet fixed.

## 9 Watchdog Timer 1

### 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, C1, C3

**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, C1, C3

**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 C1.

## 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 C1.

### 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 C1.

### 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, C1

**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, C1, C3

**Fixed:**

Not yet fixed.

## 12 SSI

### 12.1 An interrupt is not generated when using $\mu$ DMA with the SSI module if the EOT bit is set

**Description:**

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

**Workaround:**

Use the alternate  $\mu$ DMA channels for the SSI module.

**Silicon Revision Affected:**

B1

**Fixed:**

Fixed in Rev C1.

## 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 C1.

### 13.2 I<sup>2</sup>S SCLK signal is inverted in certain modes

**Description:**

When the I<sup>2</sup>S 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 I<sup>2</sup>S mode with DAC in Left-Justified audio format.

For the receiver, ensure that the CODEC is configured as the SCLK master, and the I<sup>2</sup>S receive module is configured as the SCLK slave.

**Silicon Revision Affected:**

B1

**Fixed:**

Fixed in Rev C1.

## 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  $\overline{\text{ACG}}$  bit in the **Run-Mode Clock Configuration (RCC)** register is set.

#### Workaround:

Do not set the  $\overline{\text{ACG}}$  bit in the **RCC** register.

#### Silicon Revision Affected:

B1

#### Fixed:

Fixed in Rev C1.

### 14.2 Ethernet packet loss with cables longer than 50 meters

#### Description:

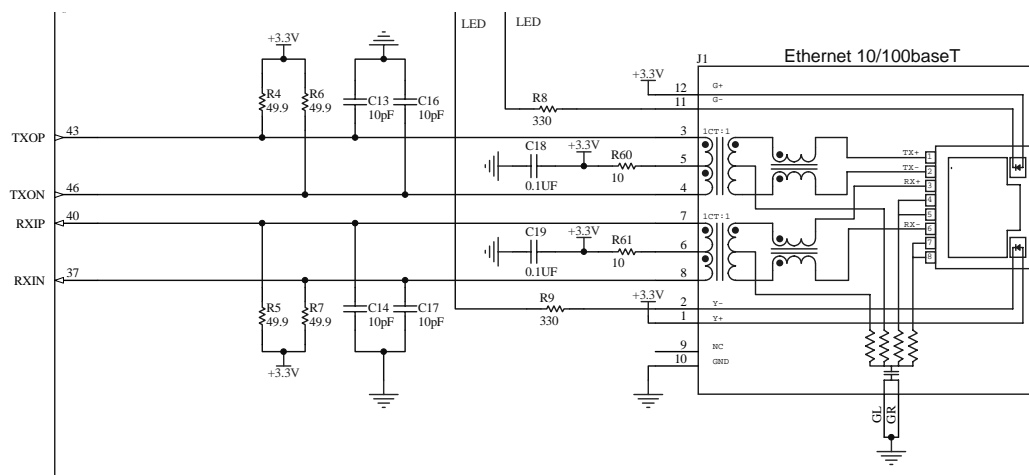
The microcontroller experiences some packet loss with Ethernet cables longer than 50 meters in normal operating conditions.

#### Workaround:

There are two possible workarounds for this issue:

1. Add 10  $\Omega$  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 2. Recommended Center-Tap Connections**



**Silicon Revision Affected:**

B1

**Fixed:**

Fixed in Rev C1.

## 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 C1.

## 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, C1, C3

**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 C1.

## 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  $\Omega$  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 C1.

## 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 C1.

## 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 C1.

## 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 C1.

## 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:**

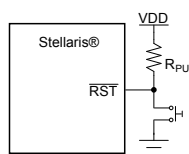
Fixed in Rev C1.

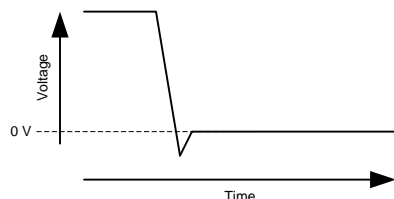
## 17 Electrical Characteristics

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

**Description:**

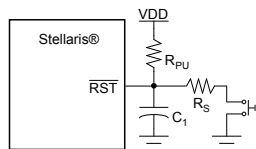
To avoid latch-up, the maximum DC ratings of the part must be strictly enforced. The most common violation of the  $V_{IN}$  electrical specification can occur when a mechanical switch or contact is connected directly to a GPIO or special function (`RST`, `WAKE`, ...) pin. The circuit shown in Figure 3 on page 26 typically has stray inductance and capacitance that can cause a voltage glitch when the switch transitions, as shown in Figure 4 on page 27. The magnitude of the glitch may exceed the  $V_{IN}$  in the maximum DC ratings table in the Electrical Characteristics chapter. Figure 5 on page 27 shows an improved circuit that eliminates the glitch.

**Figure 3. Incorrect Reset Circuitry**

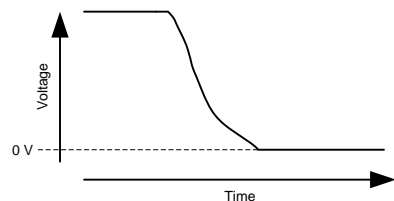
**Figure 4. Excessive Undershoot Voltage on Reset****Workaround:**

Use a circuit as shown in Figure 5 on page 27. 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 5. Recommended Reset Circuitry**

After implementing the circuit shown in Figure 5 on page 27, confirm that the voltage on the  $\overline{\text{RST}}$  input has a curve similar to the one in Figure 6 on page 27, and that the  $V_{IN}$  specification is not exceeded.

**Figure 6. Recommended Voltage on Reset****Silicon Revision Affected:**

B1

**Fixed:**

Fixed in Rev C1.

## 17.2 Power-on event may disrupt operation

**Description:**

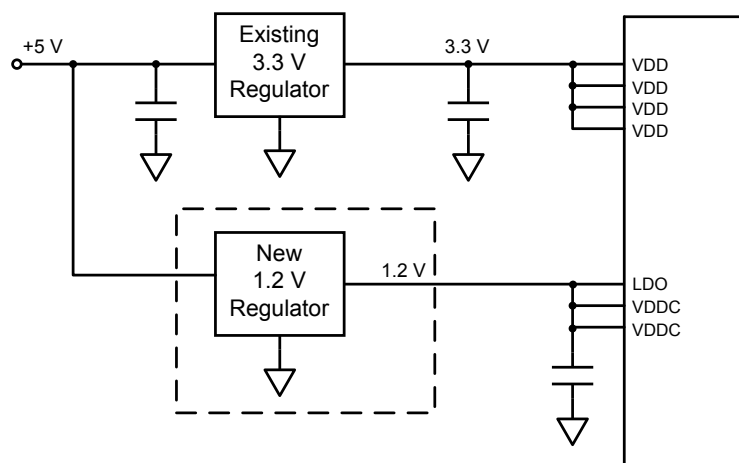
Incorrect power sequencing during power up can disrupt operation and potentially cause device failure.

**Workaround:**

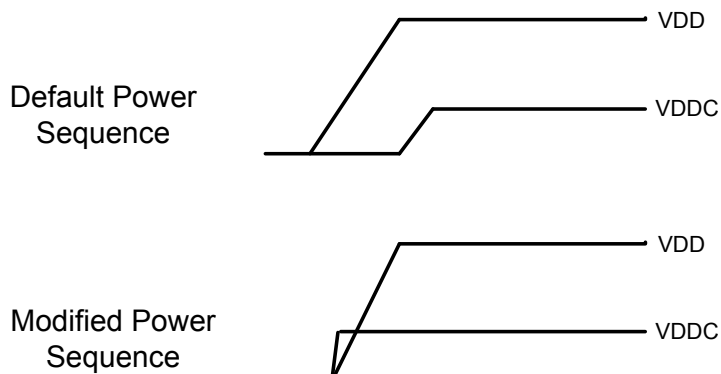
$V_{DDC}$  must be applied approximately 50  $\mu$ s before  $V_{DD}$ . Normally  $V_{DDC}$  is controlled by the part's internal LDO voltage regulator. The workaround requires the addition of an external regulator (see Figure 7) to ensure that  $V_{DDC}$  sequencing requirements are met (see Figure 8). 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 7. Configuration of External Regulator**



**Figure 8. VDDC Sequencing Requirements**



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

**Silicon Revision Affected:**

B1

**Fixed:**

Fixed in Rev C1.

---

Copyright © 2008-2010 Texas Instruments Inc. All rights reserved. Stellaris and StellarisWare are registered trademarks of Texas Instruments. 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  
108 Wild Basin, Suite 350  
Austin, TX 78746  
<http://www.ti.com/stellaris>



**TEXAS  
INSTRUMENTS**



**Cortex**  
Intelligent Processors by ARM®

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

| Products                    |  | Applications               |  |
|-----------------------------|--|----------------------------|--|
| Amplifiers                  | <a href="http://amplifier.ti.com">amplifier.ti.com</a>             | Audio                      | <a href="http://www.ti.com/audio">www.ti.com/audio</a>                                   |
| Data Converters             | <a href="http://dataconverter.ti.com">dataconverter.ti.com</a>     | Automotive                 | <a href="http://www.ti.com/automotive">www.ti.com/automotive</a>                         |
| DLP® Products               | <a href="http://www.dlp.com">www.dlp.com</a>                       | Communications and Telecom | <a href="http://www.ti.com/communications">www.ti.com/communications</a>                 |
| DSP                         | <a href="http://dsp.ti.com">dsp.ti.com</a>                         | Computers and Peripherals  | <a href="http://www.ti.com/computers">www.ti.com/computers</a>                           |
| Clocks and Timers           | <a href="http://www.ti.com/clocks">www.ti.com/clocks</a>           | Consumer Electronics       | <a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>                   |
| Interface                   | <a href="http://interface.ti.com">interface.ti.com</a>             | Energy                     | <a href="http://www.ti.com/energy">www.ti.com/energy</a>                                 |
| Logic                       | <a href="http://logic.ti.com">logic.ti.com</a>                     | Industrial                 | <a href="http://www.ti.com/industrial">www.ti.com/industrial</a>                         |
| Power Mgmt                  | <a href="http://power.ti.com">power.ti.com</a>                     | Medical                    | <a href="http://www.ti.com/medical">www.ti.com/medical</a>                               |
| Microcontrollers            | <a href="http://microcontroller.ti.com">microcontroller.ti.com</a> | Security                   | <a href="http://www.ti.com/security">www.ti.com/security</a>                             |
| RFID                        | <a href="http://www.ti-rfid.com">www.ti-rfid.com</a>               | Space, Avionics & Defense  | <a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a> |
| RF/IF and ZigBee® Solutions | <a href="http://www.ti.com/lprf">www.ti.com/lprf</a>               | Video and Imaging          | <a href="http://www.ti.com/video">www.ti.com/video</a>                                   |
|                             |  | Wireless                   | <a href="http://www.ti.com/wireless-apps">www.ti.com/wireless-apps</a>                   |