Stellaris® LM3S9B96 RevB1 Errata

This document contains errata as of October 2013 for the Stellaris LM3S9B96 microcontroller. Additional errata added after October 2013 that apply to revision B1 can be found in the *Stellaris® LM3S Tempest- and Firestorm-Class Microcontrollers Errata* (SPMZ861). Read both documents for the complete list of errata for your device.

See also the ARM® Cortex™-M3 errata (SPMZ092).

Table 1. Revision History

Date	Revision	Description
October 2013	4.3	■ Added issue “Chip select operation is not correct when using dual chip selects in Host Bus Continuous Read mode” on page 15. ■ Removed issue "Flash Write Buffer does not function above 50 MHz" and added issue “Flash memory may be corrupted if programmed at system clock speeds above 50 MHz” on page 21. ■ Added issue “iRDY timing in General-Purpose mode is not as specified in the data sheet” on page 27. ■ Added issue “Simultaneous sampling on both ADC modules yields incorrect samples” on page 37. ■ Added issue “Phase offset does not delay as expected if sample sequencers are not triggered at the same time” on page 37.
October 2012	4.2	■ Added issue “Non-word-aligned write to SRAM can cause incorrect value to be loaded” on page 11. ■ Added issue “Internal reset supervisors may not prevent incorrect device operation during power transitions” on page 12. ■ Added issue “Watchdog clear mechanism described in the data sheet does not work for the Watchdog Timer 1 module” on page 32. ■ Added issue “Watchdog Timer 1 module asserts reset signal even if not programmed to reset” on page 33. ■ Added issue “WDTLOAD yields an incorrect value when read back” on page 33. ■ Added issue “Digital comparator in last step of sequence does not trigger or interrupt” on page 35. ■ Added issue “Digital comparator interrupts do not trigger or interrupt as expected” on page 36. ■ Added issue “Missing trigger or interrupt when multiple sequences configured for processor trigger and different trigger” on page 36. ■ Added issue “ADC sample sequencers priorities are different than expected” on page 36. ■ Added issue “When UART LIN or SIR mode is enabled, µDMA burst transfer does not occur” on page 40. ■ Added issue “UART transfers fail at certain system clock frequency and baud rate combinations” on page 40. ■ Added issue “Freescale SPI Mode at low SSIClk frequencies can yield data corruption” on page 41. ■ Added issue “First two ADC samples from the internal temperature sensor must be ignored” on page 54.
June 2012	4.1	■ Clarified how to read the date code on Stellaris devices.

Date	Revision	Description
March 2011	4.0	■ Added issue “In Host-Bus 16 mode, only one byte select is asserted if only 8 bits are read” on page 25. ■ Added issue “When non-blocking reads are pending, EPI accesses can cause the NBRFIFO counter to be incorrectly decremented” on page 26. ■ Added issue “In General-Purpose mode, the framing signal is output regardless of the state of the FRMPIN bit” on page 26. ■ Added issue “In General-Purpose mode, the maximum time to wait for the iRDY signal is derived from the system clock, not the EPI clock” on page 27.
September 2011	3.9	■ Added issue “Boundary scan is not functional” on page 9. ■ Added issue “At EPI clock speeds over 15 MHz, SDRAM initialization delay is not long enough” on page 25. ■ Added issue “LIN mode Sync Break does not have the correct length” on page 39.
August 2011	3.8	■ Clarified issue “PB1 has permanent internal pull-up resistance” on page 23. ■ Added additional details to issue “MCU may fail USB certification if the EPI module is operating” on page 46. ■ Clarified issue “Special considerations for PB1” on page 48. ■ Added issue “Cannot communicate with a low-speed Device through a hub” on page 48.
July 2011	3.7	■ Corrected the date code for issue “The PIOSC cannot be calibrated by the user” on page 10. ■ Corrected the read portion of the diagram in issue “Clock signal in EPI General-Purpose mode is inverted” on page 24. ■ Added issue “Retriggering a sample sequencer before it has completed the current sequence results in continuous sampling” on page 35. ■ Added issue “MCU may fail USB certification if the EPI module is operating” on page 46. ■ Added issue “Special considerations for PB1” on page 48.
March 2011	3.6	■ 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 22 and reworded description. ■ Removed Appendix A since information is now in the data sheet.
February 2011	3.5	■ Clarified issue “The PIOSC is not trimmed by the factory” on page 10. ■ Added issue “The PIOSC cannot be calibrated by the user” on page 10. ■ Added issue “Flash memory endurance cycle specification is 100 cycles” on page 21. ■ Added issue “The µDMA does not generate a completion interrupt when transferring to and from GPTM 2A and 2B” on page 22. ■ Added issue “PB1 has permanent internal pull-up resistance” on page 23. ■ Added issue “Differential pair encodings are incorrect” on page 34. ■ Added issue “PWM fault latch does not operate correctly” on page 51. ■ Added issue "PWM6 and PWM7 do not function". ■ Added Appendix A.
January 2011	3.4	■ Added issue “PB1 has permanent internal pull-up resistance” on page 23.

Date	Revision	Description
November 2010	3.2	■ Added clarification to issue "Flash Write Buffer does not function above 50 MHz".
October 2010	3.0	■ 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 45. ■ Added issue “USB compliance test issue: USB embedded host low-speed, far-end signal compliance tests fail” on page 45.
September 2010	2.9	■ Removed the "ROM_I2CMasterErr function is incorrect" issue because the data sheet has been changed such that the ERR0R bit no longer is set when the ARBLST bit is set. ■ Additional minor clarifications and corrections.
July 2010	2.8	■ Added issue “The RTRIS bit in the UARTRIS register is only set when the interrupt is enabled” on page 39.
June 2010	2.7	■ Minor edits.
April 2010	2.6	■ Based on further examination of the "I^2C 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 22. ■ Added issue “Encoding error in the Ethernet MAC LED Encoding (MACLED) register” on page 44. ■ Added information about items fixed on Rev C3.
March 2010	2.5	■ Added issue “The prescaler does not work correctly when counting up in periodic or one-shot mode” on page 31. ■ Added issue “Snapshot must be enabled in both Timer A and B when in 32-bit snapshot mode” on page 32. ■ Added issue “Phantom interrupts occur in Smart Card mode” on page 38. ■ Added issue "I^2C arbitration may be lost when operating as a master".
Mar 2010	2.4	■ Added issue “The option to force the ROM boot loader to execute at reset with an external pin does not function” on page 20. ■ Amended the workaround for issue “A spurious DMA request is generated when the timer rolls over in Input-Edge Time mode” on page 29. ■ Reworded description of issue “The value of the prescaler register is not readable in Edge-Count mode” on page 29. ■ 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 30. ■ Added issue “Wait-on-Trigger does not assert unless the TnOTE bit is set” on page 30 . ■ Added issue “Do not enable match and timeout interrupts in 16-bit PWM mode” on page 30. ■ Added issue “Do not use µDMA with 16-bit PWM mode” on page 31. ■ Added issue “Writing the GPTMTnV register does not change the timer value when counting up” on page 31.

Date	Revision	Description
Feb 2010	2.3	■ Added issue “A spurious DMA request is generated when the timer rolls over the 16-bit boundary” on page 29. ■ Added issue “The value of the prescaler register is not readable in Edge-Count mode” on page 29. ■ 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 34.
Jan 2010	2.2	■ Modified description for “The General-Purpose Timer match register does not function correctly in 32-bit mode” on page 28 to include DMA operation. ■ Added issue “A spurious DMA request is generated when the timer rolls over in Input-Edge Time mode” on page 29. ■ Changed workaround for “Latch-up may occur if power is applied to the VBUS pin but not to VDD” on page 44 and changed status to "Fixed in Rev C."
Dec 2009	2.1	■ The status of “The Recover Locked Device sequence does not work as expected” on page 8 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 LM3S9B96 data sheet. ■ Added additional APIs to “Some ROM functions are unsupported” on page 17. ■ “The µDMA controller fails to generate capture mode DMA requests from Timer A in the Timer modules” on page 21 has been added. ■ "Ethernet packet count decremented before the FCS is read" has been removed and the content added to the LM3S9B96 data sheet. ■ The status of “Latch-up may occur if power is applied to the VBUS pin but not to VDD” on page 44 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
1.3	Boundary scan is not functional	JTAG	B1, C3, C5
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, C3, C5
2.5	Non-word-aligned write to SRAM can cause incorrect value to be loaded	System Control	B1, C3, C5
2.6	Internal reset supervisors may not prevent incorrect device operation during power transitions	System Control	B1, C3, C5

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
2.7	Chip select operation is not correct when using dual chip selects in Host Bus Continuous Read mode	System Control	B1, C3, C5
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_SSIConfigSetExpClk 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 memory endurance cycle specification is 100 cycles	Flash Memory	B1, C3, C5
4.3	Flash memory may be corrupted if programmed at system clock speeds above 50 MHz	Flash Memory	B1, C3, C5
5.1	The µDMA controller fails to generate capture mode DMA requests from Timer A in the Timer modules	µDMA	B1, C3, C5
5.2	The µDMA does not generate a completion interrupt when transferring to and from GPTM 2A and 2B	µDMA	B1, C3, C5
6.1	Port B [1:0] pins require external pull-up resistors	GPIO	B1
6.2	Schmitt input feature does not function correctly	GPIO	B1
6.3	PB1 has permanent internal pull-up 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
7.4	At EPI clock speeds over 15 MHz, SDRAM initialization delay is not long enough	EPI	B1, C3, C5
7.5	In Host-Bus 16 mode, only one byte select is asserted if only 8 bits are read	EPI	B1, C3, C5
7.6	When non-blocking reads are pending, EPI accesses can cause the NBRFIFO counter to be incorrectly decremented	EPI	B1, C3, C5
7.7	In General-Purpose mode, the framing signal is output regardless of the state of the FRMPIN bit	EPI	B1, C3, C5
7.8	In General-Purpose mode, the read and write strobes are output regardless of the state of the RW bit	EPI	B1, C3, C5
7.9	In General-Purpose mode, the maximum time to wait for the iRDY signal is derived from the system clock, not the EPI clock	EPI	B1, C3, C5
7.10	iRDY timing in General-Purpose mode is not as specified in the data sheet	EPI	B1, C3, C5
8.1	The General-Purpose Timer match register does not function correctly in 32-bit mode	General-Purpose Timers	B1, C3, C5

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
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 µDMA 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
9.2	Watchdog clear mechanism described in the data sheet does not work for the Watchdog Timer 1 module	Watchdog Timers	B1, C3, C5
9.3	Watchdog Timer 1 module asserts reset signal even if not programmed to reset	Watchdog Timers	B1, C3, C5
9.4	WDTLOAD yields an incorrect value when read back	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
10.4	Retriggering a sample sequencer before it has completed the current sequence results in continuous sampling	ADC	B1, C3, C5
10.5	Digital comparator in last step of sequence does not trigger or interrupt	ADC	B1, C3, C5
10.6	Digital comparator interrupts do not trigger or interrupt as expected	ADC	B1, C3, C5
10.7	Missing trigger or interrupt when multiple sequences configured for processor trigger and different trigger	ADC	B1, C3, C5
10.8	ADC sample sequencers priorities are different than expected	ADC	B1, C3, C5
10.9	Simultaneous sampling on both ADC modules yields incorrect samples	ADC	B1, C3, C5
10.10	Phase offset does not delay as expected if sample sequencers are not triggered at the same time	ADC	B1, C3, C5
11.1	UART Smart Card (ISO 7816) mode does not function	UART	B1

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
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
11.5	LIN mode Sync Break does not have the correct length	UART	B1, C3, C5
11.6	When UART LIN or SIR mode is enabled, µDMA burst transfer does not occur	UART	B1, C3, C5
11.7	UART transfers fail at certain system clock frequency and baud rate combinations	UART	B1, C3, C5
12.1	An interrupt is not generated when using µDMA with the SSI module if the EOT bit is set	SSI	B1
12.2	Freescale SPI Mode at low SSIClk frequencies can yield data corruption	SSI	B1, C3, C5
13.1	Some bits in the I2SMCLKCFG register do not function	I2S	B1
13.2	I^2S 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
15.5	MCU may fail USB certification if the EPI module is operating	USB	B1, C3, C5
15.6	Special considerations for PB1	USB	B1, C3, C5
15.7	Cannot communicate with a low-speed Device through a hub	USB	B1, C3, C5
15.8	USB0DM may be driven after reset	USB	
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

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
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
17.3	First two ADC samples from the internal temperature sensor must be ignored	Electrical Characteristics	B1, C3, C5

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 okay.
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.

1.3 Boundary scan is not functional

Description:

The boundary scan is not functional on this device.

Workaround:

None.

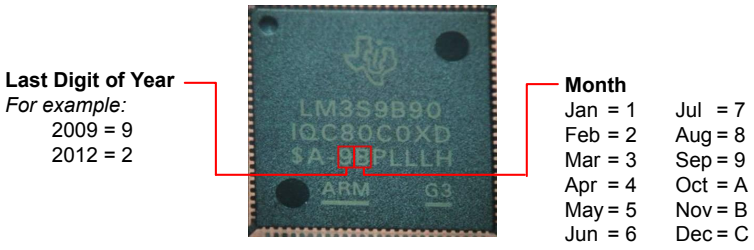
Silicon Revision Affected:

B1, C3, C5

Fixed:

Fixed on devices with date codes of 1A (October, 2011) or later.

Note: To determine the date code of your part, look at the first two characters following the dash on the third line of the part markings (highlighted in red in the following figure). The first number after the dash indicates the last decimal digit of the year. The second character indicates the month. Therefore, the following example shows a date code of 9B which indicates November 2009.



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. When the PIOSC is not trimmed, its accuracy is ±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.

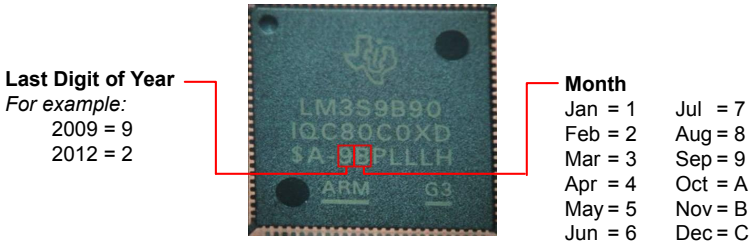
Silicon Revision Affected:

B1, C3, C5

Fixed:

Fixed on devices with date codes of 17 (July, 2011) or later.

Note: To determine the date code of your part, look at the first two characters following the dash on the third line of the part markings (highlighted in red in the following figure). The first number after the dash indicates the last decimal digit of the year. The second character indicates the month. Therefore, the following example shows a date code of 9B which indicates November 2009.



2.5 Non-word-aligned write to SRAM can cause incorrect value to be loaded

Description:

If a word-aligned value is loaded from an SRAM location into a core register, then altered by storing a byte or halfword at an unaligned offset, the altered word-aligned value is not correctly indicated when loaded into a core register. The loaded value from the SRAM location into a core register reflects the original value, not the modified value.

The following assembly sequence causes the altered value loaded into a core register to not load the correct value, even though the correct value is visible in the SRAM memory location.

```
//
// Load a word-aligned value from an SRAM location into a
// core register (such as R0)
//
LDR        R0, [SP, #+0];

//
// Store byte or halfword from the core register to
// the SRAM location at a non-word-aligned offset
//
STRB     R0, [SP, #+1];
      OR
STRB     R0, [SP, #+2];
      OR
STRB     R0, [SP, #+3];
      OR
STRH     R0, [SP, #+1];

//
// Load the same word-aligned value of the same SRAM location
// into a core register (such as R0)
```

```
//
LDR       R0, [SP, #+0];
```

This assembly sequence causes erroneous values only if these three instructions are executed in this order. However, the three instructions do not have to be consecutive, which means that other instructions can be placed in between the first and the second instructions, or the second and the third instructions, and the false value still occurs. Other instructions include, but are not limited to, branches in Flash, accesses to non-SRAM locations such as peripherals, and writes to other SRAM locations.

Pointers, structures, and unions are common C code methods that can be found in user code that may generate this assembly sequence and, therefore, result in incorrect values for variables. If using interrupts, it is possible to continue the assembly sequence in the interrupt handler, which could also return incorrect data.

For more information about this erratum as well as C code examples that may generate this assembly sequence, refer to the document, *Non-Word-Aligned Write to SRAM Additional Information* (SPMA047).

Workaround:

The type of compiler and optimization settings used in your application affects whether the problematic assembly code is generated from your user code. Each compiler behaves a little differently with respect to this erratum. The behavior for each compiler is not guaranteed due to the large number of compiler and tool version combinations.

At the assembly level, loading a volatile 32-bit-aligned word value from a different address in SRAM after storing and before loading in the assembly instruction sequence yields a correct value. A dummy SRAM load of a volatile 32-bit-aligned word from a different SRAM memory location should be inserted after the second assembly instruction (storing a byte or halfword from the core register to the desired SRAM location at a non-word-aligned offset) and before the third assembly instruction (loading the same word-aligned value of the desired SRAM location into a core register). This also means that a dummy SRAM load of a volatile 32-bit-aligned word from a different SRAM memory location should also be placed at the beginning of any interrupt routine, in case the third assembly instruction is executed before leaving the handler.

For more information about this erratum as well as C code examples that may generate this assembly sequence, refer to the document, *Non-Word-Aligned Write to SRAM Additional Information* (SPMA047).

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

2.6 Internal reset supervisors may not prevent incorrect device operation during power transitions

Description:

This microcontroller incorporates internal Power-On Reset (POR) and Brown-Out Reset (BOR) supervisors to ensure that code only executes when power to the device is within specification. However, gaps in the voltage and timing thresholds of the internal supervisors result in a risk of incorrect operation during VDD power transitions.

Unexpected operation may occur that can include brief execution of random sections of user code including ROM functions and random instructions, as well as incorrect power-up initialization. The uncontrolled brief execution of random instructions may result in the undesired erasing or writing of non-volatile memories and GPIO state changes. There is also the possibility that the device may be left in a state where it does not operate correctly until a clean power cycle has been completed.

The Power-On Reset gap occurs because the supervisor can release internal state machine operation as soon as 6.0 ms after the VDD supply reaches 1.9 V. If VDD is still below the minimum operating voltage of 3.0 V after 6.0 ms, the power-up state machine may not function correctly, resulting in the effects described above. The $\overline{\text{RST}}$ pin of the device has no effect on the initialization state machine, therefore, a complete power-cycle is required to restore the initialization state machine.

The Brown-Out Reset threshold (V_{BTH}) gap occurs because the brown-out supervisor has a threshold as low as 2.85 V, which is less than the minimum operating voltage on VDD, and also because it can take several microseconds to respond. BOR gaps can be encountered after power up, during steady state operation power-on, if the VDD rail has glitches, and also during power-down.

Workaround:

After initial power-up, any processor operation with VDD below 3.0 V may result in unexpected code execution resulting in the effects described above. The processor must be halted or the $\overline{\text{RST}}$ signal must be driven Low prior to VDD dropping below 3.0 V and stay in that state until VDD is above 3.0 V.

If VDD falls below 2.1 V, it must continue to fall until it reaches 1.5 V. VDD must stay below 1.5 V for at least 36 µs to ensure that a POR is triggered correctly. Additionally, the VDD power-up time between 1.9 V and 3.0 V must be at most 6.0 ms. If VDD falls below 3.0 V but stays above 2.1 V, it is not necessary for the voltage to continue falling below 2.1 V. VDD can come back up to 3.0 V without any additional timing requirements.

The system designer must ensure they meet the requirements listed below for power-up, steady state, and power-down:

1. The VDD power-up, steady state, and power-down waveform meets the timing requirements shown in Figure 1 on page 14.
2. The power-up transition of VDD between 1.9 V and 3.0 V must not have any points where it decreases in voltage (must be monotonic).
3. The power-down transition of VDD between 3.0 V and 1.5 V must not have any points where it increases in voltage (must be monotonic).
4. Once steady-state operation between 3.0 V and 3.6 V is achieved, $\overline{\text{RST}}$ must go Low or the CPU execution must be halted prior to VDD falling below 3.0 V.
5. The **Brown-Out Reset Control (PBORCTL)** register must be set so that a brown-out event causes a reset.

Depending on the system environment requirement, items 3, 4, and 5 in the above list may be met by using a voltage supervisor, such as the TLV803M, to monitor a higher voltage rail from which the VDD supply is regulated. Figure 2 on page 14 shows this implementation with a voltage supervisor monitoring the 5-V rail and a voltage trip point of 4.38 V. A voltage supervisor with a lower voltage trip point can be used to monitor the VDD (3.3-V) rail, however this supervisor must assert reset before VDD reaches 3.0 V. Regardless of the implemented voltage supervisor circuit, the system designer must ensure that there is enough time to assert $\overline{\text{RST}}$ Low prior to VDD falling below 3.0 V. Figure 3 on page 15 shows the resulting waveform of the circuit shown in Figure 2 on page 14.

Figure 1. VDD Waveform Signature Limits

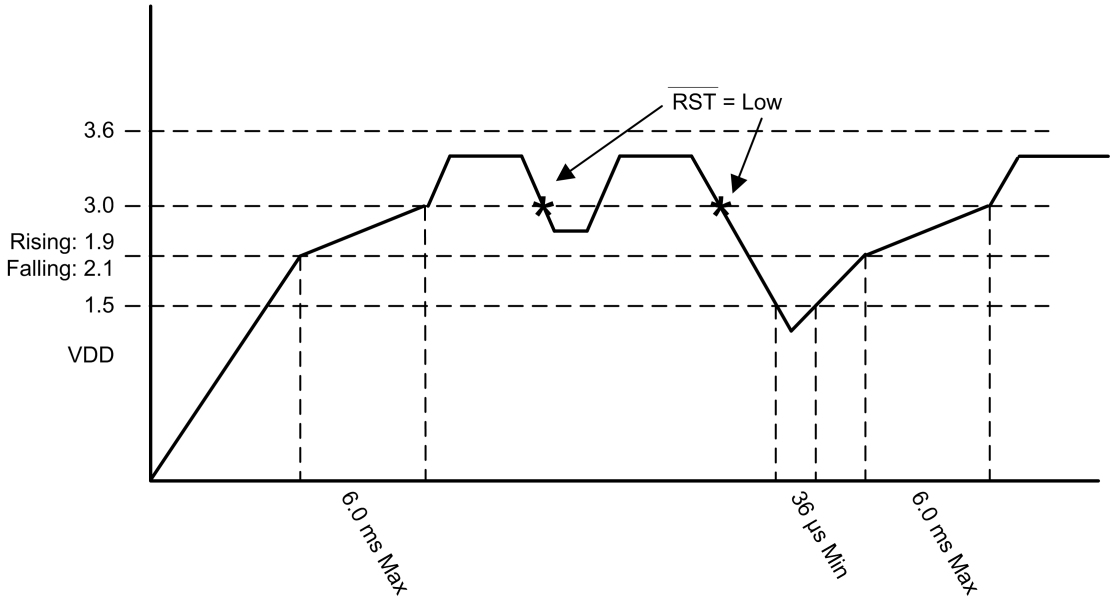


Figure 2. Using a Voltage Supervisor to Monitor the Voltage Rail

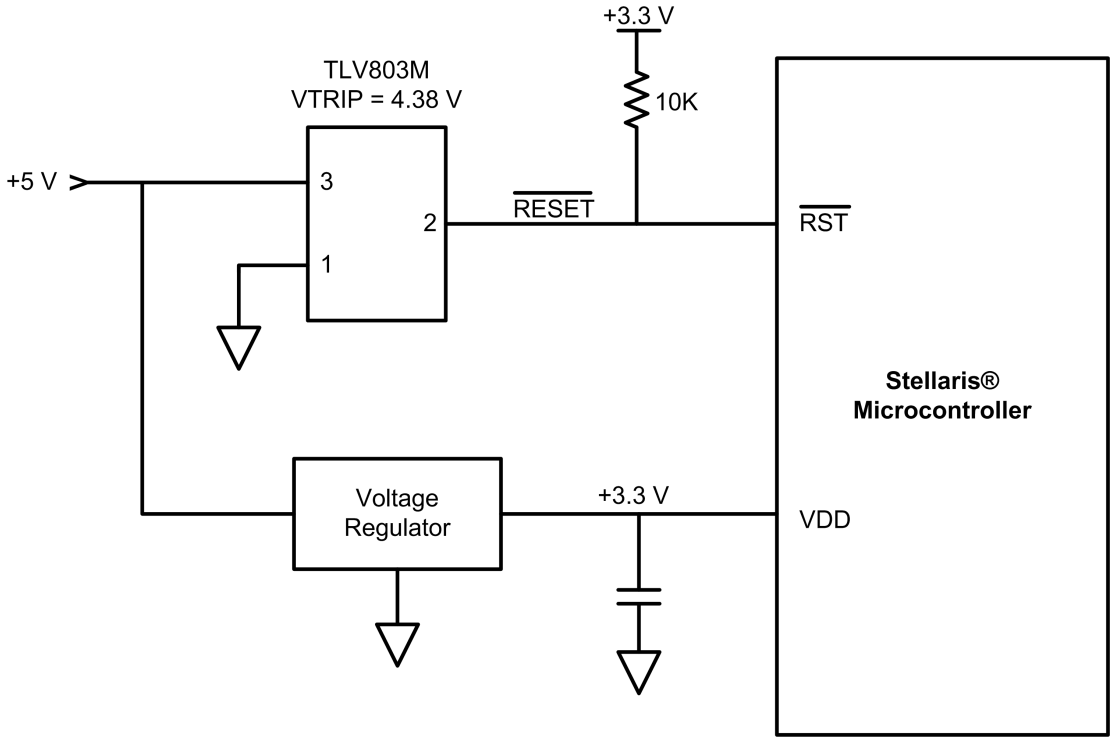
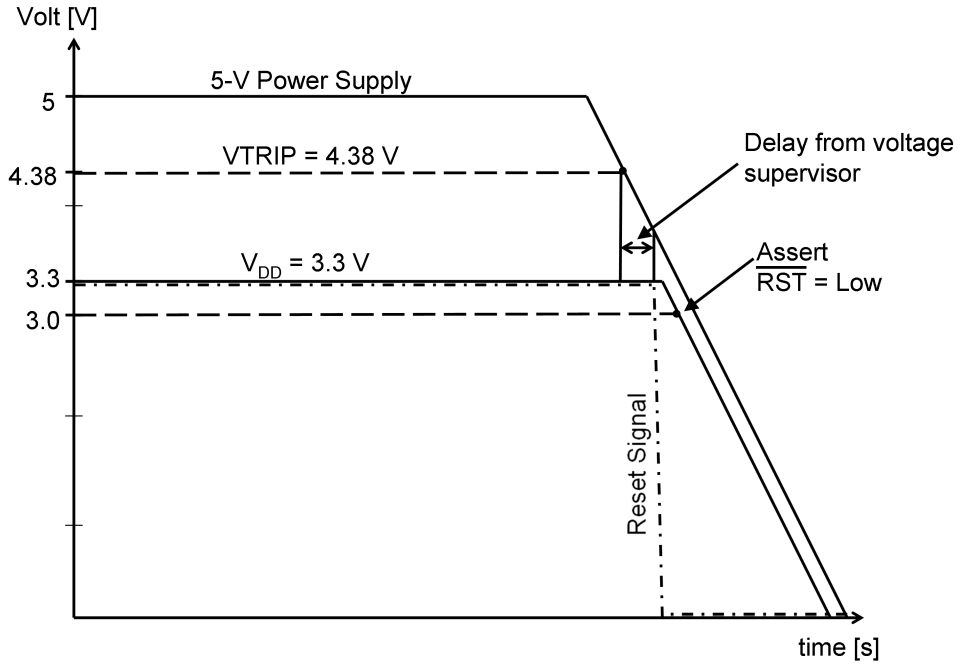


Figure 3. Resulting Waveform Using the Voltage Supervisor Circuit



Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

2.7 Chip select operation is not correct when using dual chip selects in Host Bus Continuous Read mode

Description:

Chip select operation for the first read is not correct when the EPI module is in Host Bus mode and is configured to use dual chip selects (the CSCFG field in the **EPIHBnCFG2** register is 0x2) and Continuous Read mode (the MODE field in the **EPIHBnCFG** register is 0x2). When accessing a memory region assigned to one chip select, the other chip select is asserted first along with the RD strobe. This incorrect chip select is de-asserted before the next EPI clock edge and the correct chip select is asserted on that EPI clock edge. The RD strobe remains asserted, but the number of cycles that it is asserted includes the time that it is asserted with the incorrect chip select.

For example, if the RD strobe is programmed to be asserted for two clocks (the RDWS field in the **EPIHBnCFG** register is 0x0), the signal is asserted for one clock cycle with the incorrect chip select and one clock cycle with the correct chip select. As long as the width of the RD strobe is adjusted for this one clock difference, data is read correctly in this mode. Figure 4 on page 16 shows the read and chip select timing of this example when accessing a memory region assigned to CS0n.

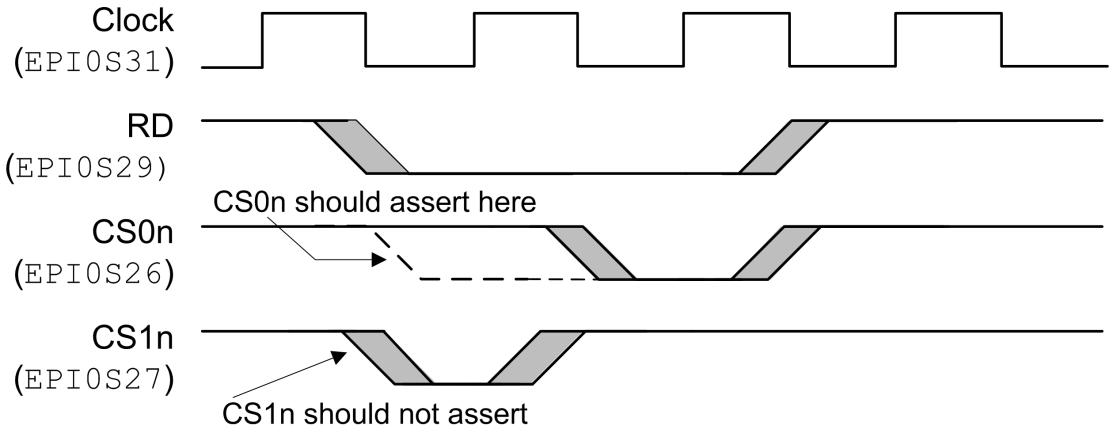
Subsequent reads while the OE signal is asserted operate as expected.

Workaround:

Use a value in the RDWS field in the **EPIHBnCFG** register that is 1 more than required for the peripheral in the system.

For example, if the peripheral requires a read strobe that is 4 EPI clocks wide, set the MODE field to be 0x2 (6 clocks) to account for the 1 clock difference in the strobe width.

Figure 4. Chip Selects



Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

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
- EPIIntStatus
- 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.

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_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 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 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.

4.3 Flash memory may be corrupted if programmed at system clock speeds above 50 MHz

Description:

Flash memory may occasionally be corrupted during programming if the system clock speed is above 50 MHz.

Workaround:

Always program Flash memory with system clock speeds of 50 MHz and below. In addition, it is always a good practice to verify that programming was successful by comparing the Flash memory contents with the expected contents.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

5 µDMA

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

Description:

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

Description:

The µDMA module does not generate a completion interrupt on the Timer 2 interrupt vector when transferring data to and from Timers 2A and 2B. The µDMA 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 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^2C, 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^2C 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

Fixed:

Fixed in Rev C3.

6.3 PB1 has permanent internal pull-up resistance

Description:

Regardless of its configuration (GPIO or alternate digital function), PB1 has a maximum internal pull-up resistance of 800 ohms that turns on when the voltage on the pin is approximately 1.2 V. Due to this internal resistance, up to 3 mA of current may be sourced during the transition from 1.2 V to 3.3 V.

Workaround:

When this pin is configured as an input, the external circuit must drive with an impedance less than or equal to 300 Ω to provide enough drive strength to over-drive the internal pull-up and achieve the necessary V_{IL} voltage level. Ensure that the driver can sink the temporary current. In addition, do not use PB1 in open-drain mode.

if this pin is configured as an output, be aware that if the output was driven high and a non-POR reset occurs, the output may be driven high after reset unless it has a 300-Ω resistor on it. Once the pin is configured as an output, the pin drives the programmed level.

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:

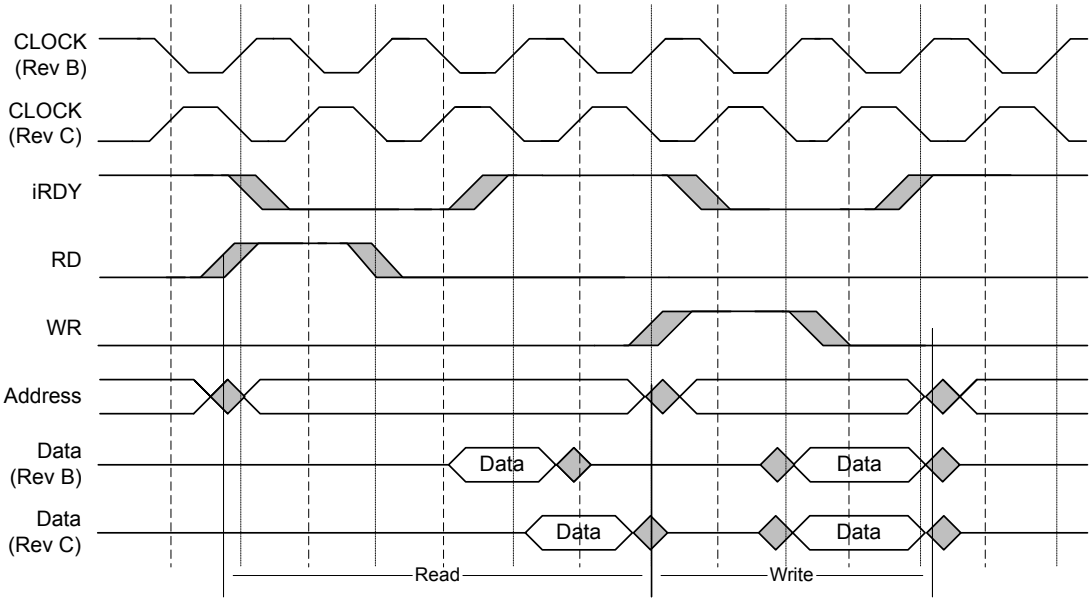
Fixed in Rev C.

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 Figure 5 on page 24 shows the timing differences between Rev B parts and Rev C parts.

Figure 5. 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.

7.4 At EPI clock speeds over 15 MHz, SDRAM initialization delay is not long enough

Description:

After enabling the EPI SDRAM interface via the **EPISDRAMCFG** register, the EPI SDRAM controller should hold off any SDRAM accesses for 100 µs. When an EPI clock speed greater than 15 MHz is used, it is possible for an access to start before the 100 µs has elapsed.

Workaround:

After enabling the EPI SDRAM interface in the **EPISDRAMCFG** register, wait 100 µs before performing any accesses to SDRAM.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

7.5 In Host-Bus 16 mode, only one byte select is asserted if only 8 bits are read

Description:

When reading from most 16-bit memories in Host-Bus 16 mode, both byte selects should be asserted for every access, even if only 8 bits are being read. The EPI controller only asserts the byte select for the required 8 bits, violating this specification.

Workaround:

For standard memory configurations, the memory operation proceeds normally. Although this behavior violates the memory specifications, the EPI reads the proper data from memory.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

7.6 When non-blocking reads are pending, EPI accesses can cause the NBRFIFO counter to be incorrectly decremented

Description:

When non-blocking reads are pending and the MCU performs a blocking read or a write from any EPI address with bits [11:4] equal to 0x07 or 0x08, the NBRFIFO counter is incorrectly decremented.

Workaround:

When a non-blocking read operation has been initiated, wait for the completion interrupt before performing any additional CPU or µDMA accesses to or from the EPI registers or the EPI memory space.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

7.7 In General-Purpose mode, the framing signal is output regardless of the state of the FRMPIN bit

Description:

In General-Purpose mode, the FRMPIN bit in the **EPI General-Purpose Configuration (EPIGPCFG)** register should control whether or not the framing signal, FRAME, is output on EPIS030. However, the FRAME signal is output regardless of the state of the FRMPIN bit.

Workaround:

If the FRMPIN signal is not required, configure the port pin to an alternate function or a GPIO.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

7.8 In General-Purpose mode, the read and write strobes are output regardless of the state of the RW bit

Description:

In General-Purpose mode, the RW bit in the **EPI General-Purpose Configuration (EPIGPCFG)** register should control whether or not the read and write strobes, RD and WR, are output on EPIS029 and EPIS028. However, the RD and WR signals are output regardless of the state of the RW bit.

Workaround:

If the read and write strobes are not required, configure the port pins to an alternate function or GPIOs.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

7.9 In General-Purpose mode, the maximum time to wait for the iRDY signal is derived from the system clock, not the EPI clock

Description:

In General-Purpose mode, the MAXWAIT field in the **EPI General-Purpose Configuration (EPIGPCFG)** register specifies the number of EPI clocks to wait for the iRDY signal to be deasserted. However, the hardware counts system clocks and not EPI clocks, so the wait may be shorter than expected.

Workaround:

Adjust the MAXWAIT configuration to account for the difference in frequency between the system clock and the EPI clock.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

7.10 iRDY timing in General-Purpose mode is not as specified in the data sheet

Description:

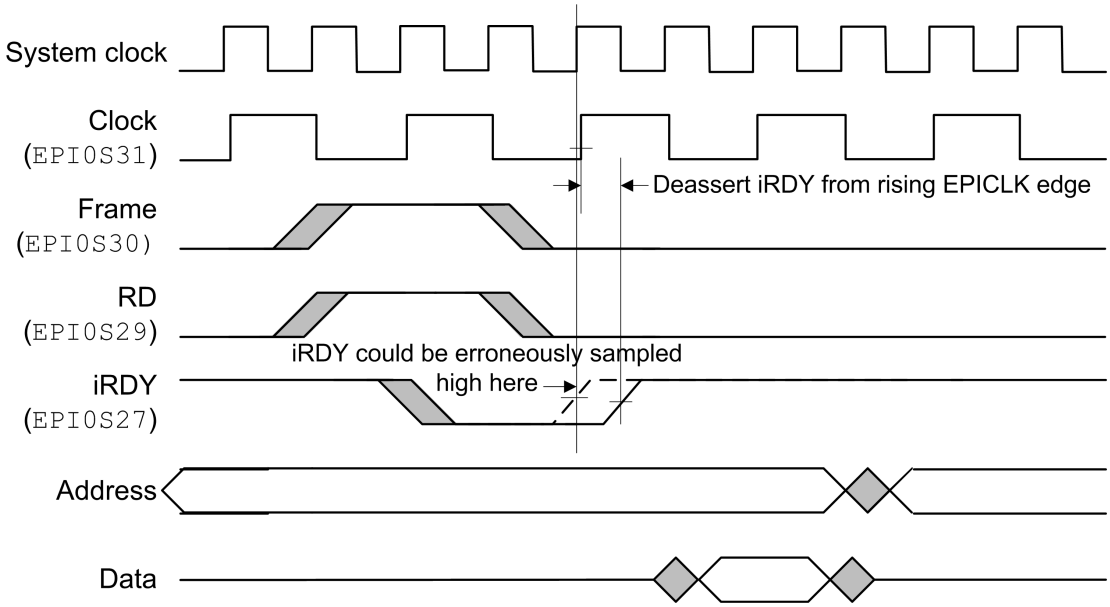
The data sheet specifies that ready input (iRDY) is sampled on the falling edge of the EPI clock, when it is actually sampled on the rising edge of system clock when configured for General-Purpose mode. The iRDY signal may be seen earlier than expected.

Workaround:

If trying to stall the address phase, this is not an issue. When trying to stall the data phase, you must ensure that iRDY is not deasserted until after the next rising edge of EPICLK. If iRDY misses the set up for the next rising system clock edge, it will be captured on the subsequent rising system clock edge.

For example, if the system clock frequency is configured for 80 MHz (12.5 ns period) and the EPI clock frequency is configured for 40 MHz (25 ns period), there is a 2.5 ns window ((25 ns – 12.5 ns)) – 10 ns, where 10 ns is the iRDY assertion or deassertion set up time (TRDYSU)) where the iRDY signal may be asserted. Figure 6 on page 28 shows where iRDY is actually sampled when trying to stall the data phase. iRDY should be deasserted after the next falling edge of EPICLK. iRDY could be erroneously sampled high if it is deasserted at the dotted line.

Figure 6. iRDY Timing



Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

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 µDMA with 16-bit PWM mode

Description:

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

Workaround:

Do not use µDMA 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.

9.2 Watchdog clear mechanism described in the data sheet does not work for the Watchdog Timer 1 module

Description:

Periodically reloading the count value into the **Watchdog Timer Load (WDTLOAD)** register of the Watchdog Timer 1 module will not restart the count, as specified in the data sheet.

Workaround:

Disable the Watchdog Timer 1 module before reprogramming the counter. Alternatively, clear the watchdog interrupt status periodically outside of the interrupt handler by writing any value to the **Watchdog Interrupt Clear (WDTICR)** register.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

9.3 Watchdog Timer 1 module asserts reset signal even if not programmed to reset

Description:

Even if the reset signal is not enabled (the RESEN bit of the **Watchdog Control (WDTCTL)** register is clear), the Watchdog Timer 1 module will assert a reset signal to the system when the time-out value is reached for a second time.

Workaround:

Clear the Watchdog Timer 1 interrupt once the time-out value is reached for the first time by writing any value to the **Watchdog Interrupt Clear (WDTICR)** register.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

9.4 WDTLOAD yields an incorrect value when read back

Description:

If the Watchdog Timer 1 module is enabled and configured to run off the PIOSC, writes to the **Watchdog Load (WDTLOAD)** register yield an incorrect value when read back.

Workaround:

None.

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 MUXn fields in the **ADCSSMUXn** 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	MUXn 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.

10.4 Retriggering a sample sequencer before it has completed the current sequence results in continuous sampling

Description:

Re-triggering a sample sequencer before it has completed its programmed conversion sequence causes the sample sequencer to continuously sample. If interrupts have been enabled, interrupts are generated at the appropriate place in the sample sequence. This problem only occurs when the new trigger is the same type as the current trigger.

Workaround:

Ensure that a sample sequence has completed before triggering a new sequence using the same type of trigger.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

10.5 Digital comparator in last step of sequence does not trigger or interrupt

Description:

If a digital comparator that is expected to trigger or interrupt is configured for the last step of a sample sequence with sequence trigger TRIGGER_PROCESSOR, TRIGGER_COMPn, TRIGGER_EXTERNAL, TRIGGER_TIMER, or TRIGGER_PWMn, the trigger or interrupt does not occur. These sequence trigger parameters should not be used when using a sample sequencer configured with only one step and a digital comparator that is expected to trigger or interrupt.

Note: Sample Sequencer 3 can only be configured for a total of one step.

Workaround:

If an extra sequence step is available in a sample sequencer, a dummy sequence step and a dummy digital comparator can be configured as the last step in the sample sequencer.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

10.6 Digital comparator interrupts do not trigger or interrupt as expected

Description:

The digital comparator configured for the ADC sample sequence step (n+1) is triggered if the voltage on the AINx input specified for step (n) meets the conditions that trigger the digital comparator for step (n+1). In this case, the conversion results are sent to the digital comparator specified by step (n+1).

Workaround:

Adjust user code or hardware to account for the fact that the voltage seen at the AINx input specified for sequence step (n) will be handled by sequence step (n+1)'s digital comparator using sequence step (n+1)'s configurations.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

10.7 Missing trigger or interrupt when multiple sequences configured for processor trigger and different trigger

Description:

If a sample sequence is configured to trigger or interrupt using a processor event and a different, consecutive sample sequence is configured to trigger or interrupt using any other event, the interrupt or trigger for the processor-triggered sample sequence will occasionally not occur, even if the processor-triggered sample sequence is configured with a higher priority.

Workaround:

None.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

10.8 ADC sample sequencers priorities are different than expected

Description:

If sample sequencer 2 (SS2) and sample sequencer 3 (SS3) have been triggered, and sample sequencer 0 (SS0) and sample sequencer 1 (SS1) have not been triggered or have already been triggered, the priority control logic compares the priorities of SS1 and SS2 rather than SS2 and SS3. For example, if SS1's priority is the highest (such as 0) and SS3's priority is higher than SS2's priority (such as SS3 = 1, SS2 = 2), SS2 is incorrectly selected to initiate the sampling conversion after

SS1. If SS1's priority is the lowest (such as 3) and SS3's priority is lower than SS2's (such as SS3 = 2, SS2 = 1), SS3 is incorrectly selected as the next sample sequencer, then SS2, then SS1.

Workaround:

If only three of the four ADC sample sequencers are needed, SS0 and SS1 can be used with either SS2 or SS3. This ensures that the execution order is as expected. If all four ADC sample sequencers are needed, the highest priority conversions should be programmed into SS0 and SS1. The sequences programmed into SS2 and SS3 occur, but not necessarily in the programmed priority order.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

10.9 Simultaneous sampling on both ADC modules yields incorrect samples

Description:

The input impedance of the analog input channel is altered if both ADC modules are used to sample the same pin at the same time. The altered input impedance results in incorrect samples.

Workaround:

Avoid incorrect samples by performing one of the following:

- Configure the ADC modules to sample at different times.
- Wait twice as long for the sample to settle.
- Halve the input impedance.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

10.10 Phase offset does not delay as expected if sample sequencers are not triggered at the same time

Description:

The phase difference set in the **ADC Sample Phase Control (ADCSPC)** register does not reference the same starting point in time if the sequencers are configured for a phase offset and are not triggered at the same time.

Workaround:

Use the same trigger to ensure that the sample sequencers will trigger at the same time. If using processor trigger and both ADC modules with phase offset, use the `GSYNC` and `SYNCWAIT` bits in

the **ADC Processor Sample Sequence Initiate (ADCPSSI)** register to ensure that the trigger occurs simultaneously. The phase offsets will not align if triggering using Trigger Always mode.

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.

11.5 LIN mode Sync Break does not have the correct length

Description:

When operating as a LIN master, the microcontroller provides a Sync Break of the length that is programmed in the BLEN field in the **UART LIN Control (UARTLCTL)** register. However, the actual Sync Break length is 1 less than what is programmed in the BLEN field as shown in Table 3 on page 39.

Table 3. SyncBreak Length

BLEN Encoding	Data Sheet Value	Actual Value
0x0	13T bits	12T bits
0x1	14T bits	13T bits
0x2	15T bits	14T bits
0x3	16T bits	15T bits

Workaround:

Adjust the BLEN encoding to correspond to the actual Sync Break required.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

11.6 When UART LIN or SIR mode is enabled, µDMA burst transfer does not occur

Description:

If the LIN or the IrDA Serial Infrared (SIR) mode is enabled in the UART peripheral and the uDMA UARTn RX or UARTn TX channel is configured to do a burst transfer, the burst data transfer does not occur.

Workaround:

Clear the SETn bit in the **DMA Channel Useburst Set (DMAUSEBURSTSET)** register to have the uDMA UART channel respond to single or burst requests to ensure that the data transfer occurs.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

11.7 UART transfers fail at certain system clock frequency and baud rate combinations

Description:

UART data transfers using the TXRIS and RXRIS interrupt bits and FIFOs fail for certain combinations of the system clock frequency and baud rate.

System Clock Freq [MHz]	32	24	16	10	8	5	4	2	1
Failing Baud Rate [bps]	<460800	<460800	<230400	<460800	<115200	<230400	<57600	<38400	<19200

Workaround:

Use a system clock frequency above 32MHz if using the UART with the raw interrupt status bits or use µDMA UART data transfers instead of the TXRIS and RXRIS bits. When using µDMA UART data transfers, there are no system clock frequency and baud rate conflicts.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

12 SSI

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

Description:

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

Workaround:

Use the alternate µDMA channels for the SSI module.

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

12.2 Freescale SPI Mode at low SSIClk frequencies can yield data corruption

Description:

Data transmitted by the SPI slave may be corrupted when using Freescale SPI Mode 0 at an SSIClk frequency between 0.5 MHz to 1.1 MHz and a system clock frequency of 33 MHz or lower.

Workaround:

Operate the Freescale SPI Mode 0 at an SSIClk frequency above 1.1 MHz and use a system clock frequency above 33 MHz or use a different mode.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

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:

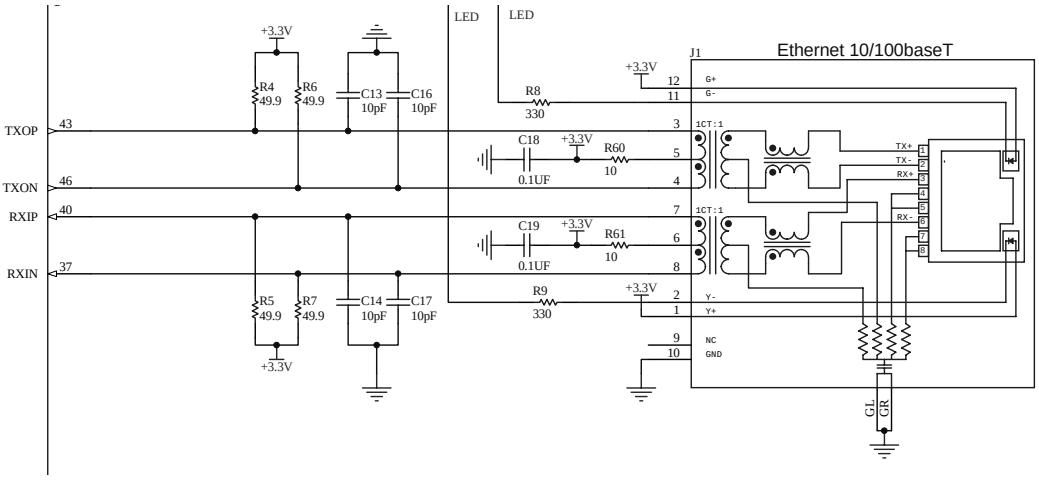
The microcontroller experiences some packet loss with Ethernet cables longer than 50 meters in normal operating conditions.

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 7. 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.

15.5 MCU may fail USB certification if the EPI module is operating

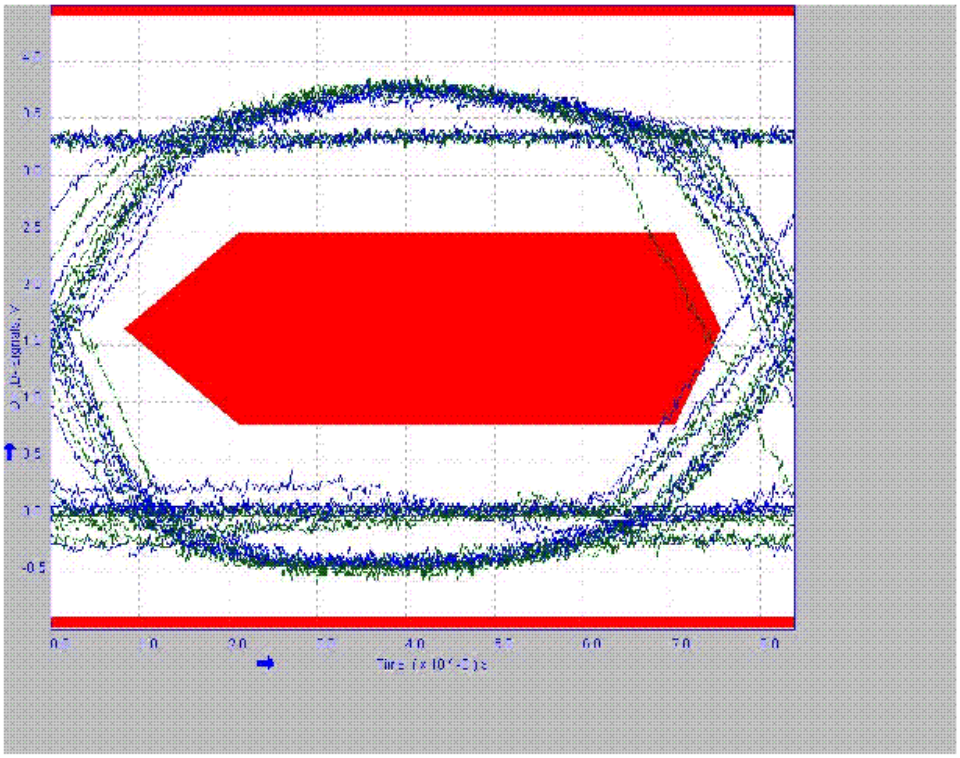
Description:

If the EPI module is operating, the USB interface may fail USB certification. Failures have been seen for the eye diagram and jitter (see Figure 8 on page 47). Because of these issues, there is a potential for data corruption on the USB interface when using the USB and EPI at the same time. The risk of data corruption is small, but is dependent on the system design.

The USB hardware senses bit errors in the CRC check for control, interrupt, and bulk transfers. If an error occurs, the host hardware requests a resend of the packet up to three times. The application software could be written to address this error further if needed.

Figure 8. Example of USB Certification Failure

- Signal eye:
  Eye Diagram Test fails



- EOP width: 169.0995ns
  EOP width passes
- Receivers: reliable operation on tier Tier 6
  Receivers pass
- Measured Signalling Rate: 11.92707Mbps
  signal rate conditionally passes
- Crossover voltage range: 1.546667 V to 1.680000 V
  Mean crossover = 1.615889 V
  First crossover at 1.660000 V(11 other differential crossovers checked)
  crossover voltages pass
- Consecutive jitter range: -165.1885ns to 1.811468ns, RMS jitter 52.27256ns
  Paired JK jitter range: 752.3810ps to 2.871429ns, RMS jitter 1.821542ns
  Paired KJ jitter range: 1.185714ns to 2.168571ns, RMS jitter 1.695397ns
  jitter fails

USB certification can be attained if an application does not require the USB and the EPI to be operating simultaneously. Customers who do not intend to pursue USB certification should determine if their application can handle the resulting small amount of error.

Workaround:

None.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

15.6 Special considerations for PB1

Description:

When using PB1 as a GPIO or digital alternate function, special considerations are required due to issue “PB1 has permanent internal pull-up resistance” on page 23.

Workaround:

The DEVMODOTG and DEVMOD bits in the **USB General-Purpose Control and Status (USBGPCS)** register can be used to configure the USB controller to operate only in Host mode or Device mode and allowing PB0 and PB1 to be used as GPIOs or digital alternate functions. If both the DEVMODOTG and DEVMOD bits are set, indicating Device mode, the USB0VBUS signal is not driven, therefore the USB VBUS signal must be monitored using a GPIO as an input to detect connect and disconnect. This monitoring must be done with a GPIO other than PB1, because PB1 is not 5-V tolerant. Note that this erratum does not affect devices operating in OTG mode. The USB0VBUS signal operates as specified.

In addition, if the USB functionality is not used on the device, in order to be able to use PB1 as a GPIO or digital alternate function, the user application must enable the USB module in the **RCGC2** register, set the DEVMODOTG bit, and then disable the USB module again. The restrictions detailed in issue “PB1 has permanent internal pull-up resistance” on page 23 still apply.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

15.7 Cannot communicate with a low-speed Device through a hub

Description:

When the USB controller is operating as a Host and a low-speed packet is sent to a Device through a hub, the subsequent Start-of-Frame is corrupted. After a period of time, this corruption causes the USB controller to lose synchronization with the hub, resulting in data corruption.

Workaround:

None.

Silicon Revision Affected:

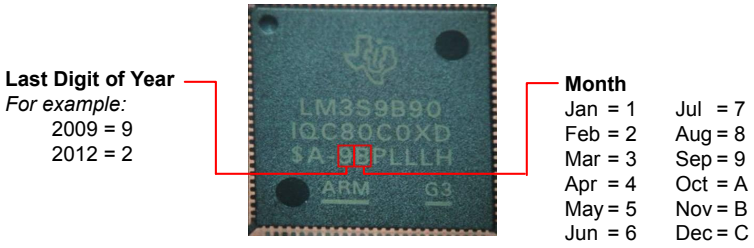
B1, C3, C5

Fixed:

Fixed on devices with date codes of 1A (October 2011) or later. In addition, the system clock on the MCU must be at least 30 MHz.

Note: To determine the date code of your part, look at the first two characters following the dash on the third line of the part markings (highlighted in red in the following figure). The first number after the dash indicates the last decimal digit of the year. The second character

indicates the month. Therefore, the following example shows a date code of 9B which indicates November 2009.



15.8 USB0DM may be driven after reset

Description:

If the microcontroller is reset while the USB device is connected to an upstream port with the SOFTCONN bit set in the **USB Power (USBPOWER)** register, the USB0DM signal is driven to 2 V for 66 µs after the microcontroller comes out of reset. This activity can appear to be unsolicited traffic to the upstream port. This traffic is generally ignored, but may cause unexpected behavior from the upstream host controller.

Workaround:

If the system can determine that a reset is about to occur, disconnect the USB peripheral by clearing the SOFTCONN bit in the **USB Power (USBPOWER)** register prior to resetting the device. If the microcontroller reset is asynchronous, there is no workaround.

Silicon Revision Affected:

Fixed:

Not yet fixed.

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 th **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 ($\overline{RST}$, $\overline{WAKE}$, ...) pin. The circuit shown in Figure 9 on page 52 typically has stray inductance and capacitance that can cause a voltage glitch when the switch transitions, as shown in Figure 10 on page 52. The magnitude of the glitch may exceed the V_{IN} in the maximum DC ratings table in the Electrical Characteristics chapter. Figure 11 on page 53 shows an improved circuit that eliminates the glitch.

Figure 9. Incorrect Reset Circuitry

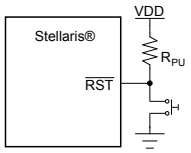
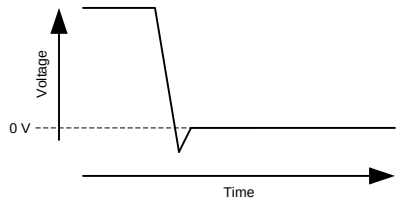


Figure 10. Excessive Undershoot Voltage on Reset

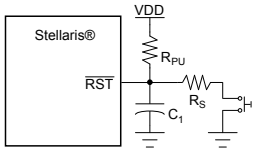


Workaround:

Use a circuit as shown in Figure 11 on page 53. 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:

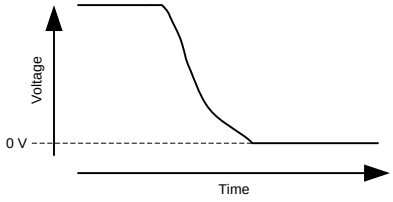
- R_{PU} = 10 kΩ
- R_S = 470 Ω
- C_1 = 0.01 µF

Figure 11. Recommended Reset Circuitry



After implementing the circuit shown in Figure 11 on page 53, confirm that the voltage on the $\overline{RST}$ input has a curve similar to the one in Figure 12 on page 53, and that the V_{IN} specification is not exceeded.

Figure 12. 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 13) to ensure that V_{DDC} sequencing requirements are met (see Figure 14). 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 13. Configuration of External Regulator

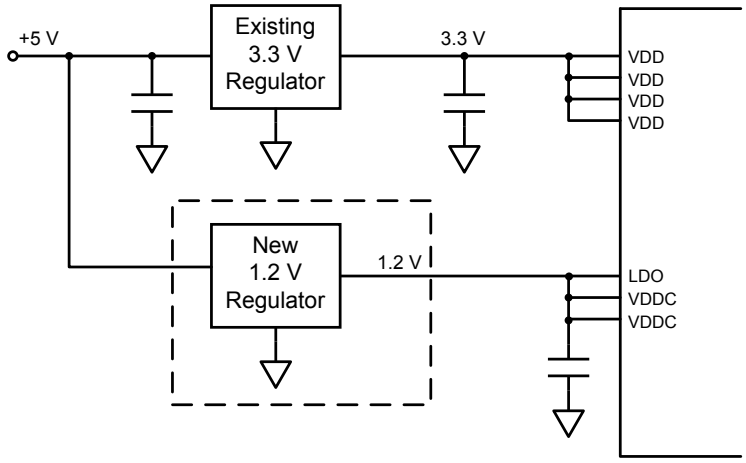
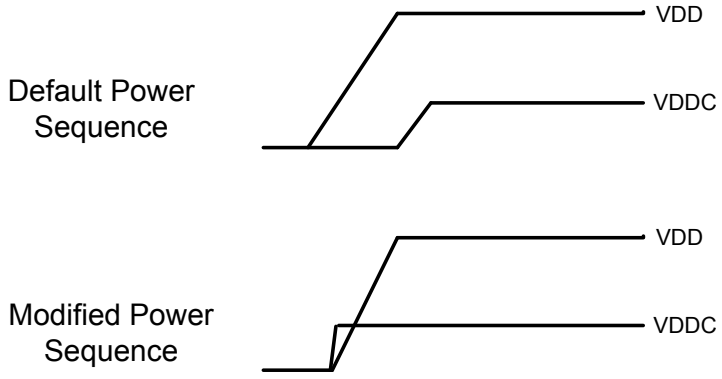


Figure 14. VDDC Sequencing Requirements



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

Silicon Revision Affected:

B1

Fixed:

Fixed in Rev C.

17.3 First two ADC samples from the internal temperature sensor must be ignored

Description:

The analog source resistance (Rs) to the ADC from the internal temperature sensor exceeds the specified amount of 500Ω. This causes a settling time requirement that is longer than the sampling interval to the converter.

Workaround:

Three consecutive samples from the same channel must be taken to accurately sample the internal temperature sensor using the ADC. The first two consecutive samples should be discarded and the third sample can be kept. These consecutive samples cannot be interrupted by sampling another channel.

Silicon Revision Affected:

B1, C3, C5

Fixed:

Not yet fixed.

Texas Instruments Incorporated
108 Wild Basin, Suite 350
Austin, TX 78746
http://www.ti.com/lm3s
http://www-k.ext.ti.com/sc/technical-support/customer-support-centers.htm

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license 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 significant portions 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. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265