Stellaris® LM4F132C4QC RevB0/B1 Errata

This document contains known errata at the time of publication for the Stellaris LM4F132C4QC microcontroller. The table below summarizes the errata and lists the affected revisions. See the data sheet for more details.

See also the ARM® Cortex™-M4F errata, ARM publication number PRD40-PRDC-013029.

Table 1. Revision History

Date	Revision	Description
March 2013	1.1	■ Removed issue "Clearing the LDORDMIS interrupt status bit requires an extra write" as it does not apply to this device.
February 2013	1.0	■ Split RevA and RevB errata into separate documents. ■ Started tracking revision history for RevB errata document.

Table 2. List of Errata

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
1.1	The JTAG INTEST instruction does not properly capture data	JTAG	B0, B1
2.1	With a specific clock configuration, device may not wake from Deep-sleep mode	System Control	B0, B1
2.2	The MOSC verification circuit does not detect a loss of clock after the clock has been successfully operating	System Control	B0, B1
2.3	Device may not wake correctly from Sleep mode under certain circumstances	System Control	B0, B1
2.4	Resets fail while in Deep-sleep when using certain clock configurations	System Control	B0, B1
2.5	Deep-sleep clock frequency incorrect if a watchdog reset occurs upon entry	System Control	B0, B1
2.6	Some devices may not start properly during power up if V_{DDC} is decaying between 200 and 100 mV when power is reapplied	System Control	B0
2.7	Longer reset pulse needed if device is in Deep-Sleep mode with the LFIOSC as the clock source	System Control	B0, B1
3.1	Some Hibernation module registers may not have the correct value in two situations	Hibernation	B0, B1
3.2	Reading the HIBRTCC and HIBRTCSS registers may provide incorrect values	Hibernation	B0, B1
3.3	Device fails to wake from hibernation within a certain time after hibernation is requested	Hibernation	B0, B1
3.4	RTC match event is missed if it occurs in a certain window	Hibernation	B0, B1
3.5	GPIO pins may be released from retention while transitioning into or out of hibernation	Hibernation	B0

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
4.1	The START bit in the EEPROM Support Control and Status (EESUPP) register does not function	EEPROM	B0, B1
5.1	In three cases, two peripherals cannot both be programmed to use µDMA	µDMA	B0, B1
6.1	JTAG controller does not ignore transitions on PC0/TCK when it is configured as a GPIO	GPIO	B0, B1
6.2	GPIO Port B1 has a leakage path to ground when VDD is removed	GPIO	B0, B1
6.3	GPIO pins may glitch on power up	GPIO	B0
7.1	GPTMSYNC bits require manual clearing	General-Purpose Timers	B0, B1
7.2	The GPTMPP register does not correctly indicate 32/64-bit timer capability	General-Purpose Timers	B0, B1
7.3	Wait-for-Trigger mode is not available for PWM mode	General-Purpose Timers	B0, B1
7.4	The prescalar does not work properly when counting up in Input Edge-Time mode when the GPTM Timer n Interval Load (GPTMTnILR) register is written with 0xFFFF	General-Purpose Timers	B0, B1
8.1	Watchdog Timer 1 module cannot be used without enabling other peripherals first	Watchdog Timers	B0, B1
8.2	Watchdog clear mechanism described in the data sheet does not work for the Watchdog Timer 1 module	Watchdog Timers	B0, B1
8.3	Watchdog Timer 1 module asserts reset signal even if not programmed to reset	Watchdog Timers	B0, B1
8.4	WDTLOAD yields an incorrect value when read back	Watchdog Timers	B0, B1
8.5	WDTMIS register does not indicate an NMI interrupt from WDT0	Watchdog Timers	B0, B1
8.6	Watchdog timer reloads on any write to the Watchdog Interrupt Clear (WDTICR) register	Watchdog Timers	B0, B1
8.7	The Watchdog Test (WDTTEST) register can be changed even when the registers are locked	Watchdog Timers	B0, B1
8.8	The Watchdog Load (WDTLOAD) register cannot be changed when using a debugger while the STALL bit is set	Watchdog Timers	B0, B1
9.1	Retriggering a sample sequencer before it has completed the current sequence results in continuous sampling	ADC	B0, B1
9.2	Digital comparator in last step of sequence does not trigger or interrupt	ADC	B0, B1
9.3	Digital comparator interrupts do not trigger or interrupt as expected	ADC	B0, B1
9.4	ADC sample sequencers priorities are different than expected	ADC	B0, B1
9.5	ADC sample sequencer only samples when using certain clock configurations	ADC	B0, B1
9.6	First two ADC samples from the internal temperature sensor must be ignored	ADC	B0, B1
10.1	When UART SIR mode is enabled, µDMA burst transfer does not occur	UART	B0, B1

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
10.2	UART transfers fail at certain system clock frequency and baud rate combinations	UART	B0, B1
11.1	I^2C glitch filter not available on early revisions of the device	I2C	B0
12.1	USB Host controller may not be used to communicate with a low-speed Device when connected through a hub	USB	B0, B1
12.2	USB controller sends EOP at end of device remote wake-up	USB	B0, B1

1 JTAG

1.1 The JTAG INTEST instruction does not properly capture data

Description:

The INTEST instruction is used to test the microcontroller's internal logic and is not commonly used in customer applications. The EXTEST instruction functions correctly.

Workaround:

None.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

2 System Control

2.1 With a specific clock configuration, device may not wake from Deep-sleep mode

Description:

With the following specific clock configuration, the device fails to wake from Deep-sleep mode approximately 1 out of 1500 times. The configuration that may cause the issue is as follows:

- The PLL is using MOSC as the clock source, AND
- The PLL is the system clock source before going in to Deep-sleep mode, AND
- The Low-Frequency Internal Oscillator (LFIOSC) is the clock source during Deep-sleep

Workaround:

Either:

- Use the PIOSC as the clock source for the PLL, OR
- Manually disable the PLL before entering Deep-sleep mode, OR

- Use the PIOSC as the clock source during Deep-sleep

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

2.2 The MOSC verification circuit does not detect a loss of clock after the clock has been successfully operating

Description:

If the MOSC clock source has been powered up and operating correctly and is subsequently removed or flatlines, the MOSC verification circuit does not indicate an error condition.

Workaround:

Use Watchdog module 1, which runs off of PIOSC, to reset the system if the MOSC fails.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

2.3 Device may not wake correctly from Sleep mode under certain circumstances

Description:

With a certain configuration, the device may not wake correctly from Sleep mode because invalid data may be fetched from the prefetch buffer. The configuration that causes this issue is as follows:

- The system clock must be at least 40 MHz
- Interrupts must be disabled

Workaround:

Use following code instead of the ROM-based function `ROM_SysCtlSleep()` to put the device into Sleep mode:

```
__asm int
CPUwfi_safe(void) {
//
// Wait for the next interrupt.
//
wfi;
mov r0,#0 // force bx lr to not start until after clocks back on
bx lr
}
```

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

2.4 Resets fail while in Deep-sleep when using certain clock configurations

Description:

If a system reset occurs while in Deep-sleep mode when the MOSC is configured as the clock source for both Run mode and Deep-sleep mode and the PIOSC is configured to power down in Deep-sleep, the MOSC is immediately disabled. The system cannot be clocked because the PIOSC is configured to be off. A power-on reset (POR) is required to get the system out of this state.

Workaround:

Use the PIOSC during Deep-sleep or use a system clock other than the MOSC.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

2.5 Deep-sleep clock frequency incorrect if a watchdog reset occurs upon entry

Description:

If a watchdog reset occurs within 10 run-time clock cycles of entering Deep-sleep mode, the clocking configuration for Deep-sleep may be overlooked. If this occurs, the first time the device enters Deep-sleep after the reset, the Run mode parameters used for the system clock frequency are used instead.

The originally configured Deep-sleep clock configuration is reapplied after this first time entering Deep-sleep.

Workaround:

If the Run mode clock frequency does not have a significant impact to the user application, no additional steps are necessary. If the Run mode clock frequency is undesirable for Deep-sleep mode, the watchdog module should be powered down in Run mode before entering Deep-sleep to ensure that a watchdog event does not occur during the entry into Deep-sleep.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

2.6 Some devices may not start properly during power up if V_{DDC} is decaying between 200 and 100 mV when power is reapplied

Description:

The LDO may not start properly during power up if V_{DDC} is decaying between 200 and 100 mV when power is reapplied. If this situation occurs, the device does not begin operating, and V_{DDC} may not reach its specified levels. A power-on reset recovers the device.

Workaround:

Ensure that V_{DDC} drops below 100 mV before reapplying power to the device.

Silicon Revision Affected:

B0

Fixed:

Fixed on B1.

2.7 Longer reset pulse needed if device is in Deep-Sleep mode with the LFIOSC as the clock source

Description:

If the device is in Deep-Sleep mode with the LFIOSC as the clock source, the specified reset pulse is not sufficient to reset the part in all cases.

Workaround:

Ensure that the reset pulse is at least 30 ms if the part may be in Deep-Sleep mode with the LFIOSC as the clock source.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

3 Hibernation

3.1 Some Hibernation module registers may not have the correct value in two situations

Description:

Some Hibernation module registers may not have the correct value in two different situations:

1. After enabling the hibernation 32-kHz oscillator by setting the CLK32EN bit in the **Hibernation Control (HIBCTL)** register.
2. When the CLK32EN bit is set, both the RTCEN and PINWEN bits in the **HIBCTL** register are clear, and any kind of reset occurs.

The following Hibernation module registers are affected:

- **HIBRTCLD**
- **HIBRTCM0**
- **HIBRTCSS**
- **HIBRTCT**
- **HIBIM**

Note that the register values may or may not be correct, but software cannot assume that these registers have any specific values following the occurrence of the situations described above.

Workaround:

Ensure that every bit in these registers is correctly initialized in application software following the occurrence of the situations described above.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

3.2 Reading the HIBRTCC and HIBRTCSS registers may provide incorrect values

Description:

Reads from the **Hibernation RTC Counter (HIBRTCC)** and **Hibernation RTC Sub Seconds (HIBRTCSS)** registers may not be correct.

Workaround:

Use the following code sequence to read from the **HIBRTCC** and **HIBRTCSS** registers:

```
//
// Disable Interrupts
//
IntMasterDisable();

//
// A) For HIB_RTCC or HIB_RTCSS individual register reads
//

do
{
    ulRTC = HWREG(HIB_RTCC);
}   while (ulRTC != HIBREG(HIB_RTCC));

//
// B) For synchronized reads of both the HIB_RTCC and HIB_RTCSS
//
```

```
do {
    ulRTC    = HWREG(HIB_RTCC);
    ulRTCSS  = HWREG(HIB_RTCSS);
    ulRTCSS2 = HWREG(HIB_RTCSS);
    ulRTC1   = HWREG(HIB_RTCC);
}   while ((ulRTC != ulRTC1) || (ulRTCSS != ulRTCSS2));

//
// Re-enable interrupts
//
IntMasterEnable();
```

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

3.3 Device fails to wake from hibernation within a certain time after hibernation is requested

Description:

If a wake event occurs during a small window after the device enters Hibernate mode, the device cannot wake from hibernation. The window in which this issue occurs extends from 31 µs before the $\overline{\text{HIB}}$ signal is asserted until V_{DD} drops below the BOR threshold, if BOR is enabled, or the POR falling edge threshold. Note that this erratum does not apply when using the VDD3ON mode because V_{DD} does not drop in this mode.

Workaround:

Add a StellarisWare `SysCtlReset()` function after the hibernation request in the following manner:

```
HibernateRequest();

//
// Wait till the isolation has been applied
//

while ((HWREG(HIB_CTL) & HIB_CTL_CLK32EN) == HIB_CTL_CLK32EN)
{

}

SysCtlReset();
```

In addition, add the following code to the reset handler

```
//
// Halt code execution if in Hibernate as supplies decay
//

while( HWREG(HIBCTL) == 0x80000000)
```

```
{
}
```

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

3.4 RTC match event is missed if it occurs in a certain window

Description:

An RTC match event is missed if the match occurs within three 32.768-kHz clocks (92 µs) after setting the HIBREQ bit in the **Hibernation Control (HIBCTL)** register.

Workaround:

Compare the RTC counter value before going into hibernation with the RTC match value and if the match is within three counts of the RTC sub seconds counter, hold off entering into hibernation until the match has occurred.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

3.5 GPIO pins may be released from retention while transitioning into or out of hibernation

Description:

The GPIO pins may be released from retention when in VDD3ON mode while transitioning into or out of hibernation. When this occurs, the GPIOs return to their default POR state.

Workaround:

None.

Silicon Revision Affected:

B0

Fixed:

Fixed on B1.

4 EEPROM

4.1 The START bit in the EEPROM Support Control and Status (EESUPP) register does not function

Description:

Setting the START bit should begin error recovery if the PRETRY or ERETRY bit in the **EESUPP** register is set. However, setting this bit does not perform any function.

Workaround:

Execute the `EEPROMInit()` function and then manually retry the failed operation.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

5 µDMA

5.1 In three cases, two peripherals cannot both be programmed to use µDMA

Description:

For the following pairs of peripherals, both peripherals cannot both be configured to use µDMA:

- SSI0 and SSI1
- UART2 and USBEP1
- UART0 and UART2

Workaround:

Configure peripherals such that the combinations of peripherals listed above are not both using µDMA.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

6 GPIO

6.1 JTAG controller does not ignore transitions on PC0/TCK when it is configured as a GPIO

Description:

When PC0/TCK is configured as a GPIO, toggling on the pin may cause the device to execute unexpected JTAG instructions.

Workaround:

Only use PC0/TCK as a JTAG pin. Do not use it as a GPIO. Ensure that this pin is connected to a pull-up to VDD.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

6.2 GPIO Port B1 has a leakage path to ground when VDD is removed

Description:

When the device is unpowered and a voltage is applied to PB1, there is a leakage path to ground that results in 45 µA of leakage current. Note that this leakage can also occur during hibernation when not using the VDD3ON mode.

Workaround:

None.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

6.3 GPIO pins may glitch on power up

Description:

The following circumstances could result in GPIOs glitching Low during power up.

- When V_{DD} rises to around 0.8 V, the device drives the GPIOs Low to ~400 mV above the V_{DD} rail. The voltage on the GPIOs rises with V_{DD} until V_{DD} reaches ~2.9 V, at which point the GPIOs go into their default configuration.
- Some devices may drive the GPIOs to ground during power up for less than 5 µs when V_{DDC} is ~ 400-500 mV.

Workaround:

None.

Silicon Revision Affected:

B0

Fixed:

Fixed on B1.

7 General-Purpose Timers

7.1 GPTMSYNC bits require manual clearing

Description:

The **GPTM Synchronize (GPTMSYNC)** register allows software to synchronize a number of timers. The bits in this register should be self-clearing after setting bits to synchronize selected timers, but they are not.

Workaround:

When bits in the **GPTMSYNC** register are set, software must clear the bits prior to setting them for a subsequent update. When using StellarisWare APIs, instead of just calling the `TimerSynchronize()` function once, software should call the function a second time with 0 as a parameter, as shown below :

```
TimerSynchronize(TIMER0_BASE, TIMER_0A_SYNC | TIMER_1A_SYNC);

TimerSynchronize(TIMER0_BASE, 0);
```

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

7.2 The GPTMPP register does not correctly indicate 32/64-bit timer capability

Description:

The **GPTM Peripheral Properties (GPTMPP)** register reads as 0x0 on the 32/64-bit wide timers, which indicates that the timer is a 16/32-bit timer. It should read as 0x1 on these timers, indicating a 32/64-bit wide timer.

Workaround:

In situations where code is required to dynamically determine the capabilities of a specific timer, create a look-up table based on the `CLASS` field of the **Device Identification 0 (DID0)** register.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

7.3 Wait-for-Trigger mode is not available for PWM mode

Description:

Daisy chaining functionality of the general-purpose timers is only valid for One-shot and Periodic modes. If the TnWOT bit of the **GPTM Timer n Mode (GPTMTnMR)** register is set, and the nth timer is configured for PWM mode, the nth timer will not wait for the (n-1)th timer to trigger it and will begin counting immediately when enabled. If, instead, the nth timer is configured for One-shot or Periodic mode and the (n-1)th timer is configured for PWM mode, the nth timer would never begin counting as it will never receive a trigger from the (n-1)th timer in the daisy chain.

Workaround:

None.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

7.4 The prescalar does not work properly when counting up in Input Edge-Time mode when the GPTM Timer n Interval Load (GPTMTnILR) register is written with 0xFFFF

Description:

If the GPTM is configured in Input Edge-Time count-up mode with the **GPTM Timer n Interval Load (GPTMTnILR)** register equal to 0xFFFF, the prescaler does not work properly.

Workaround:

Do not load 0xFFFF into the **GPTMTnILR** register when counting up in Input Edge-Time mode.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

8 Watchdog Timers

8.1 Watchdog Timer 1 module cannot be used without enabling other peripherals first

Description:

The Watchdog Timer 1 module is not fully enabled by setting the WDT1 bit in the **Run Mode Clock Gating Control Register 0 (RCGC0n)** register and, therefore, the module cannot be used unless a different peripheral is enabled first.

Workaround:

Enable at least one of the following peripherals before enabling the Watchdog Timer 1 module—UARTn, SSIn, or ADC—by setting the respective bit(s) in the **RCGUART**, **RCGCSSI**, or **RCGCADC** registers.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

8.2 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:

B0, B1

Fixed:

Not yet fixed.

8.3 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:

B0, B1

Fixed:

Not yet fixed.

8.4 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:

B0, B1

Fixed:

Not yet fixed.

8.5 WDTMIS register does not indicate an NMI interrupt from WDT0

Description:

The `WDTMIS` bit of the **Watchdog Masked Interrupt Status (WDTMIS)** register does not get set if a watchdog time-out NMI interrupt from Watchdog Timer Module 0 has been signaled to the interrupt controller. A watchdog interrupt can be programmed to be a non-maskable interrupt (NMI) by setting the `INTTYPE` bit in the **Watchdog Control (WDTCTL)** register.

Workaround:

None.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

8.6 Watchdog timer reloads on any write to the Watchdog Interrupt Clear (WDTICR) register

Description:

Any write to the **Watchdog Interrupt Clear (WDTICR)** register reloads the watchdog timer counter when the timeout interrupt is enabled, regardless of whether the timeout interrupt has been asserted.

Workaround:

Do not write to the **Watchdog Interrupt Clear (WDTICR)** register unless an interrupt has triggered.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

8.7 The Watchdog Test (WDTTEST) register can be changed even when the registers are locked

Description:

When the **Watchdog Lock (WDTLOCK)** register is 0x1, all registers in the Watchdog Timer module should be unable to be written. However, even in this situation, the **Watchdog Test (WDTTEST)** register can be modified.

Workaround:

Ensure that software does not write the **Watchdog Test (WDTTEST)** register unless necessary.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

8.8 The Watchdog Load (WDTLOAD) register cannot be changed when using a debugger while the STALL bit is set

Description:

The **Watchdog Load (WDTLOAD)** register cannot be changed when using a debugger with the STALL bit in the **Watchdog Test (WDTTEST)** register set.

Workaround:

Avoid changing the **Watchdog Load (WDTLOAD)** register with the debugger connected when the STALL bit is set.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

9 ADC

9.1 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:

B0, B1

Fixed:

Not yet fixed.

9.2 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 that 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:

B0, B1

Fixed:

Not yet fixed.

9.3 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:

B0, B1

Fixed:

Not yet fixed.

9.4 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:

B0, B1

Fixed:

Not yet fixed.

9.5 ADC sample sequencer only samples when using certain clock configurations

Description:

The ADC sample sequencer does not sample if using either the MOSC or the PIOSC as both the system clock source and the ADC clock source.

Workaround:

There are three possible workarounds:

- Enable the PLL and use it as the system clock source.
- Configure the MOSC as the system clock source and the PIOSC as the ADC clock source.
- Enable the PLL, configure the PIOSC as the ADC clock source and as the system clock source, then subsequently disable the PLL using HWREG(0x400fe060) != 0x00000200.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

9.6 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:

B0, B1

Fixed:

Not yet fixed.

10 UART

10.1 When UART SIR mode is enabled, µDMA burst transfer does not occur

Description:

If the IrDA Serial Infrared (SIR) mode is enabled in the UART peripheral and the µDMA is mapped to either UARTn RX or UARTn TX and 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 µDMA channel mapped to the UART to respond to single or burst requests to ensure that the data transfer occurs.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

10.2 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:

B0, B1

Fixed:

Not yet fixed.

11 I2C

11.1 I^{2}C glitch filter not available on early revisions of the device

Description:

A glitch filter was added on Revision B1 to the I^{2}C signals to improve immunity to noise. This filter is enabled in the **I2C Master Configuration (I2CMCR)** register . Devices that are earlier revisions do not have this capability, and as a result, when the I^{2}C SCL or SDA signal is rising and noise on the signal causes it to cross back below the VIL threshold, data loss or corruption can occur. Arbitration is lost and the module must be reset to resume operation.

Workaround:

Minimize noise on the I^{2}C signals.

Silicon Revision Affected:

B0

Fixed:

Fixed on B1.

12 USB

12.1 USB Host controller may not be used to communicate with a low-speed Device when connected through a hub

Description:

Occasionally when the USB controller is operating as a Host and a low-speed packet is sent to a Device when connected through a hub, the subsequent Start-of-Frame will be 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:

B0, B1

Fixed:

Not yet fixed.

12.2 USB controller sends EOP at end of device remote wake-up

Description:

When the USB controller is operating as a Device and is suspended by the Host, and the USB controller issues a remote wake-up, an end of packet (EOP) is sent to the Host at the end of the Device's remote wake-up signal. Although this EOP is not expected, issues related to remote wake-up have not been witnessed.

Workaround:

None.

Silicon Revision Affected:

B0, B1

Fixed:

Not yet fixed.

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

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