Stellaris® LM4F120H5QR Rev A1/A3/B0 Errata

Note: The Stellaris LM4F120H5QR microcontroller has been replaced by the identical Tiva™ C Series TM4C1233H6PM microcontroller.

This document contains known errata at the time of publication for the Stellaris LM4F120H5QR microcontroller, revisions A1, A3, and B0. Additional errata that applies to revisions A1, A3, and B0, as well as revisions following B0 can be found in *Stellaris® LM4F Microcontrollers Silicon Revisions 6 and 7 Errata* (SPMZ856). Read both documents for a complete list of errata for your device.

See also the ARM® Cortex™-M4F errata (SPMZ637).

Table 1. Revision History

Date	Revision	Description
June 2014	2.6	■ Removed "The DITHER bit in the **ADC Control (ADCCTL)** Register does not Function" as this issue is duplicated in *Stellaris® LM4F Microcontrollers Silicon Revisions 6 and 7 Errata* (SPMZ856). ■ Clarified issue "Some devices may not start properly during power up".
August 2013	2.5	■ Moved errata that also affected later silicon revisions to *Stellaris® LM4F Microcontrollers Silicon Revisions 6 and 7 Errata (SPMZ856).* ■ Clarified issue "Missing trigger or interrupt when multiple sequences configured for processor trigger and different trigger" by replacing with“Data may not be present in the FIFO at the time of the sequence interrupt or trigger ” on page 21.
April 2013	2.4	■ Clarified issue “If the EEPROM is programmed, both the Flash memory and EEPROM will be unable to be programmed after a POR” on page 18 to indicate it was fixed in RevB1. ■ Corrected issue “GPIO Port B1 has a leakage path to ground when VDD is removed” on page 20 to indicate it has only been seen in RevA1. ■ Added issue "Writes to some General-Purpose Timers registers cause the counter to increment or decrement". ■ Clarified that issue "USB controller sends EOP at end of device remote wake-up" does not affect USB certification. ■ Deleted the following issues and made appropriate content changes to the data sheet: – The JTAG INTEST instruction does not properly capture data – Watchdog timer reloads on any write to the Watchdog Interrupt Clear (WDTICR) register – The Watchdog Test (WDTTEST) register can be changed even when the registers are locked – UART transfers fail at certain system clock frequency and baud rate combinations
April 2013	2.3	Combined Rev A1, A3 and B0 into one document, which added two Rev B0 issues to this document: ■ "Deep-sleep clock frequency incorrect if a watchdog reset occurs upon entry". ■ “If the EEPROM is programmed, both the Flash memory and EEPROM will be unable to be programmed after a POR” on page 18.
March 2013	2.2	Removed reference to LDORDMIS interrupt status bit in “Clearing the BORMIS interrupt status bit requires an extra write” on page 11 as it does not apply to this device.
February 2013	2.1	■ Split RevA and RevB errata into separate documents. ■ Added issue "The JTAG INTEST instruction does not properly capture data".

Date	Revision	Description
		■ Clarified “Internal reset supervisors may not prevent incorrect device operation during power transitions” on page 8. ■ Clarified that "Resets fail while in Deep-sleep when using certain clock configurations” on page 15" is not yet fixed. ■ Added issue “Brown-out operation functions differently than specified ” on page 12. ■ Added issue “Some devices may not start properly during power up ” on page 12. ■ Added issue "Longer reset pulse needed if device is in Deep-Sleep mode with the LFIOSC as the clock source". ■ Added issue “GPIO pins may be released from retention while transitioning into or out of hibernation ” on page 13. ■ Added issue “The KEY bit in the Boot Configuration (BOOTCFG) register does not function ” on page 15. ■ Added issue "In three cases, two peripherals cannot both be programmed to use µDMA". ■ Added issue “GPIO pins may glitch on power up ” on page 20. ■ Added issue "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". ■ Added issue "Watchdog timer reloads on any write to the Watchdog Interrupt Clear (WDTICR) register". ■ Added issue "The Watchdog Test (WDTTEST) register can be changed even when the registers are locked". ■ Added issue "The Watchdog Load (WDTLOAD) register cannot be changed when using a debugger while the STALL bit is set". ■ Added issue “The BUSY bit in the ADC Active Sample Sequencer (ADCACTSS) register does not function ” on page 21. ■ Added issue "The `DITHER` bit in the **ADC Control (ADCCTL)** register does not function". ■ Clarified "When UART SIR mode is enabled, µDMA burst transfer does not occur". ■ Removed issue "Clearing the RXRIS bit when configured for LIN mode causes the UART to not transfer data" as LIN content was removed from data sheet. ■ Clarified that issue “Freescale SPI Mode at low SSIClk frequencies can yield data corruption” on page 22 was fixed on revision B0. ■ Added issue “I^2C glitch filter not available on early revisions of the device ” on page 22. ■ Added issue "USB controller sends EOP at end of device remote wake-up".
October 2012	2.0	■ Added issue “Non-word-aligned write to SRAM can cause incorrect value to be loaded” on page 9. ■ Added issue "Resets fail while in Deep-sleep when using certain clock configurations". ■ Added issue “Deep-sleep clock configuration incorrect if certain resets occur upon entry” on page 11. ■ Added issue “Clearing the BORMIS interrupt status bit requires an extra write” on page 11. ■ Added issue "ADC sample sequencers priorities are different than expected". ■ Added issue "ADC sample sequencer only samples when using certain clock configurations". ■ Added issue "UART transfers fail at certain system clock frequency and baud rate combinations". ■ Added issue "Clearing the RXRIS bit when configured for LIN mode causes the UART to not transfer data."

Date	Revision	Description
July 2012	1.9	■ Removed issue "Device does not wake from Deep-sleep mode if the hibernation oscillator is the clock source and a reset occurs". ■ Added issue "Wait-for-Trigger mode is not available for PWM mode". ■ Added issue "Watchdog Timer 1 module cannot be used without enabling other peripherals first". ■ Added issue issue "Watchdog clear mechanism described in the data sheet does not work for the Watchdog Timer 1 module". ■ Added issue "Watchdog Timer 1 module asserts reset signal even if not programmed to reset". ■ Added issue "WDTLOAD yields an incorrect value when read back". ■ Added issue "WDTMIS register does not indicate an NMI interrupt from WDT0". ■ Removed issue "A specific sequence is required when the MOSC is used to clock the ADC module". ■ Added issue "Digital comparator in last step of sequence does not trigger or interrupt". ■ Added issue "Digital comparator interrupts do not trigger or interrupt as expected". ■ Added issue "Missing trigger or interrupt when multiple sequences configured for processor trigger and different trigger". ■ Added issue "When UART SIR mode is enabled, µDMA burst transfer does not occur". ■ Added issue “Freescale SPI Mode at low SSIClk frequencies can yield data corruption” on page 22. ■ Added issue "First two ADC samples from the internal temperature sensor must be ignored".
May 2012	1.8	■ Added issue “Internal reset supervisors may not prevent incorrect device operation during power transitions” on page 8 which replaces "MCU executes code after BOR before proper power is restored" and "The POR and BOR threshold may vary from the specification". ■ Added issue “Hibernation write corruption on arbitrary power loss” on page 12.
March 2012	1.7	■ Added issue "Device does not wake from Deep-sleep mode if the hibernation oscillator is the clock source and a reset occurs". ■ Added issue "The START bit in the EEPROM Support Control and Status (EESUPP) register does not function". ■ Added additional information regarding PD7 and PF0 to issue “Some GPIO register bits default to the incorrect state” on page 18. ■ Added issue “GPIO Port B1 has a leakage path to ground when VDD is removed” on page 20. ■ Added issue "A specific sequence is required when the MOSC is used to clock the ADC module".
December 2011	1.6	■ Noted that issue “DID0 register shows revision A0 for revision A1 devices” on page 7 is fixed on revision A3. ■ Noted that issue “Device may not operate correctly at certain frequencies” on page 8 is fixed on revision A3. ■ Added issue "The MOSC verification circuit does not detect a loss of clock after the clock has been successfully operating". ■ Added issue "Device may not wake correctly from Sleep mode under certain circumstances". ■ Added issue "Device fails to wake from hibernation within a certain time after hibernation is requested".
November 2011	1.5	■ Clarified issue “Boundary scan does not function correctly” on page 6. ■ Clarified issue "MCU executes code after BOR before proper power is restored".

Date	Revision	Description
		■ Clarified issue "The POR and BOR threshold may vary from the specification". ■ Clarified issue “Device may not operate correctly at certain frequencies” on page 8. ■ Added issue "With a specific clock configuration, device may not wake from Deep-sleep mode" . ■ Clarified issue "Some Hibernation module registers may not have the correct value in two situations". ■ Clarified issue “USB boot loader in ROM does not operate correctly” on page 13. ■ Added issue "Reading the HIBRTCC and HIBRTCSS registers may provide incorrect values". ■ Added issue "JTAG controller does not ignore transitions on PC0/TCK when it is configured as a GPIO".
November 2011	1.4	■ Updated issue “DID0 register shows revision A0 for revision A1 devices” on page 7. ■ Added issue “Precision Internal Oscillator (PIOSC) is untrimmed on devices with date codes prior to January 2012” on page 7. ■ Added issue “Device may not operate correctly at certain frequencies” on page 8. ■ Added issue "GPTMSYNC bits require manual clearing". ■ Added issue "The GPTMPP register does not correctly indicate 32/64-bit timer capability".
September 2011	1.3	■ Added issue “Boundary scan does not function correctly” on page 6. ■ Added issue "MCU executes code after BOR before proper power is restored". ■ Added issue "The POR and BOR threshold may vary from the specification". ■ Added issue “Flash memory page 0 and 1 may be erased if reset occurs during Flash memory erase operation” on page 14. ■ Added issue “EEPROM blocks must be accessed in alternate pairs to avoid corruption of data” on page 17. ■ Added issue “EEPROM blocks 0 through 3 may be erased if reset occurs during an EEPROM write” on page 17. ■ Added issue “Reset during Flash memory program or erase or an EEPROM write causes Suspend state” on page 15. ■ Added issue “PB1 has permanent internal pull-up resistance” on page 19. ■ Added issue "Retriggering a sample sequencer before it has completed the current sequence results in continuous sampling". ■ Added additional information to issue “Higher current than expected is consumed while V_{DD} ramps up until V_{DDC} crosses 1 V” on page 23. ■ Added issue “Nominal current consumption is 650 µA higher than specified” on page 23. ■ Added issue “V_{DD} inrush current of up to 500 mA is seen while V_{DD} ramps up” on page 23.

Date	Revision	Description
June 2011	1.2	■ Added issue “DID0 register shows revision A0 for revision A1 devices” on page 7. ■ Clarified issue "Some Hibernation module registers may not have the correct value in two situations". ■ Added issue “USB boot loader in ROM does not operate correctly” on page 13. ■ Added issue “ROM_SysCtlClockSet() does not operate correctly with fractional dividers” on page 14. ■ Added issue “When a 1-kB Flash page is erased, the adjacent page is also erased” on page 14. ■ Added issue “Higher current than expected is consumed while V_{DD} ramps up until V_{DDC} crosses 1 V” on page 23.
May 2011	1.1	■ Added issue “Some GPIO register bits default to the incorrect state” on page 18.
March 2011	1.0	Started tracking revision history.

Table 2. List of Errata

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
1.1	Boundary scan does not function correctly	JTAG	A1
2.1	DID0 register shows revision A0 for revision A1 devices	System Control	A1
2.2	Precision Internal Oscillator (PIOSC) is untrimmed on devices with date codes prior to January 2012	System Control	A1
2.3	Device may not operate correctly at certain frequencies	System Control	A1
2.4	Internal reset supervisors may not prevent incorrect device operation during power transitions	System Control	A1, A3
2.5	Non-word-aligned write to SRAM can cause incorrect value to be loaded	System Control	A1, A3
2.6	Deep-sleep clock configuration incorrect if certain resets occur upon entry	System Control	A1, A3
2.7	Clearing the BORMIS interrupt status bit requires an extra write	System Control	A1, A3
2.8	Brown-out operation functions differently than specified	System Control	A1, A3
2.9	Some devices may not start properly during power up	System Control	A1, A3, B0
3.1	Hibernation write corruption on arbitrary power loss	Hibernation	A1, A3
3.2	GPIO pins may be released from retention while transitioning into or out of hibernation	Hibernation	A1, A3, B0
4.1	USB boot loader in ROM does not operate correctly	ROM	A1
4.2	ROM_SysCtlClockSet() does not operate correctly with fractional dividers	ROM	A1
5.1	When a 1-kB Flash page is erased, the adjacent page is also erased	Flash memory	A1
5.2	Flash memory page 0 and 1 may be erased if reset occurs during Flash memory erase operation	Flash memory	A1
5.3	The KEY bit in the Boot Configuration (BOOTCFG) register does not function	Flash memory	A1, A3

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
6.1	Reset during Flash memory program or erase or an EEPROM write causes Suspend state	Flash memory, EEPROM	A1
7.1	EEPROM blocks must be accessed in alternate pairs to avoid corruption of data	EEPROM	A1
7.2	EEPROM blocks 0 through 3 may be erased if reset occurs during an EEPROM write	EEPROM	A1
7.3	If the EEPROM is programmed, both the Flash memory and EEPROM will be unable to be programmed after a POR	EEPROM	B0
8.1	Some GPIO register bits default to the incorrect state	GPIO	A1
8.2	PB1 has permanent internal pull-up resistance	GPIO	A1
8.3	GPIO Port B1 has a leakage path to ground when VDD is removed	GPIO	A1
8.4	GPIO pins may glitch on power up	GPIO	A1, A3, B0
9.1	The first ADC sample when using differential mode is incorrect	ADC	A1, A3
9.2	Data may not be present in the FIFO at the time of the sequence interrupt or trigger	ADC	A1, A3
9.3	The BUSY bit in the ADC Active Sample Sequencer (ADCACTSS) register does not function	ADC	A1, A3
10.1	Freescale SPI Mode at low SSIClk frequencies can yield data corruption	SSI	A1, A3
11.1	I^2C glitch filter not available on early revisions of the device	I2C	A1, A3, B0
12.1	Higher current than expected is consumed while V_{DD} ramps up until V_{DDC} crosses 1 V	Electrical Characteristics	A1
12.2	Nominal current consumption is 650 µA higher than specified	Electrical Characteristics	A1
12.3	V_{DD} inrush current of up to 500 mA is seen while V_{DD} ramps up	Electrical Characteristics	A1

1 JTAG

1.1 Boundary scan does not function correctly

Description:

Boundary scan does not function correctly and should not be used. This issue does not affect the use of JTAG for programming Flash memory or debug.

Workaround:

None.

Silicon Revision Affected:

A1

Fixed:

Fixed on A3.

2 System Control

2.1 DID0 register shows revision A0 for revision A1 devices

Description:

The **Device Identification 0 (DID0)** register shows the revision of the device. The register should read 0x1005.0001 for A1, but instead it reads 0x1005.0000.

Workaround:

Read the ROM revision at address 0x0100.0010. This value is 0x1a9 on A1 silicon.

Silicon Revision Affected:

A1

Fixed:

Fixed on A3.

2.2 Precision Internal Oscillator (PIOSC) is untrimmed on devices with date codes prior to January 2012

Description:

The Precision Internal Oscillator (PIOSC) is untrimmed on some devices during factory test prior to shipment. The PIOSC on untrimmed devices has an error of up to ±10 %. Normally the PIOSC is trimmed to 16 MHz ± 1% at room temperature and 16 MHz ±3% across the operating temperature range.

In addition, the USB bootloader cannot operate if the PIOSC is not calibrated.

Workaround:

The PIOSC can be trimmed by the user in one of two ways: automatically with the Hibernation module, and manually with a user-defined calibration value based on another clock source.

By using the Hibernation module with a functioning 32.768-kHz clock source, the PIOSC can be automatically calibrated using the following method:

1. Set the CAL bit in the **Precision Internal Oscillator Calibration (PIOSCCAL)** register; the results of the calibration are shown in the RESULT field in the **Precision Internal Oscillator Statistic (PIOSCSTAT)** register.
2. After calibration is complete, the PIOSC is trimmed using the trimmed value returned in the CT field.

If the Hibernation module is not used in the system, the user must program a user-defined calibration value. The user can program the UT value in the PIOSCCAL register to adjust the PIOSC frequency. As the UT value increases, the generated period increases. To commit a new UT value, first set the UTEN bit, then program the UT field, and then set the UPDATE bit. The adjustment finishes within a few clock periods and is glitch free. For more information, see the section entitled, "Precision Internal Oscillator Operation (PIOSC)" in the System Control chapter in the data sheet.

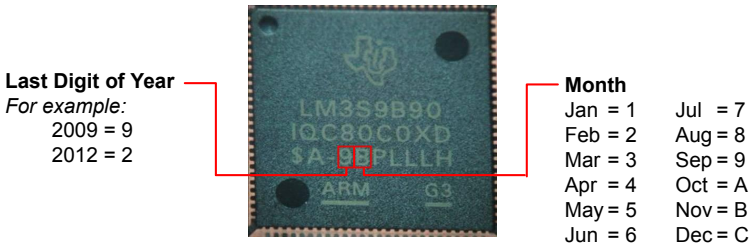
Silicon Revision Affected:

A1

Fixed:

Fixed on devices with date codes of 0x21 (January, 2012) or later. Fixed on all A3 devices.

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.



2.3 Device may not operate correctly at certain frequencies

Description:

When operating at system clock (SysClk) frequencies such that [35 MHz ≤ SysClk ≤ 45 MHz] or [70 MHz ≤ SysClk ≤ 80 MHz], an error in the digital control logic may result in inverted data causing incorrect program execution.

Workaround:

- When using the PLL, regardless of the clock source to the PLL, do not use SYSDIV values of 2.5, 4.5, 5, or 5.5.
- When not using the PLL and clocking from an external oscillator connected to MOSC, ensure that the system clock is below 35 MHz.

Note that this issue is not a concern when using the PIOSC or an external crystal of any allowed frequency connected to MOSC as the system clock, with the PLL bypassed.

Silicon Revision Affected:

A1

Fixed:

Fixed on A3.

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

Description:

The 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. This also results in a change of the minimum operating voltage to 3.05 V.

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 part may be left in a state where it will 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 118.5 µs after the VDD supply reaches 2.70 V. If VDD is still below the minimum operating voltage of 3.05 V after 118.5 µs, 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:

This issue is resolved in B0 silicon. If designing for B0 silicon, design for a VDD (Min) specification of 3.15 V.

Silicon Revision Affected:

A1, A3

Fixed:

Fixed on B0.

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:

A1, A3

Fixed:

Fixed on B0.

2.6 Deep-sleep clock configuration incorrect if certain resets occur upon entry

Description:

If an external reset, a brown-out reset, or a watchdog reset occurs when entering Deep-sleep mode with a system clock of any frequency, the clocking configuration for Deep-sleep may be overlooked.

If one of these resets occurs within 10 run-time clock cycles of entering Deep-sleep mode, the first time the device enters Deep-sleep after the reset, the Run mode parameters used for the system clock frequency is used instead of the originally configured Deep-sleep parameters. If the PIOSC was on in Run mode and configured to power-down in Deep-sleep, this is ignored and stays on. The **DCGC** register is used for the peripheral clock enables.

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

Workaround:

If the Run mode configurations do not have a significant impact to the user application, no additional steps are necessary.

If the Run mode configurations are undesirable for Deep-sleep mode, an external reset, a brown-out reset, or a watchdog reset that occurs when entering Deep-sleep should be followed by entering then exiting Deep-sleep mode. This allows the next entry to Deep-sleep to use the originally configured Deep-sleep clocking conditions.

Silicon Revision Affected:

A1, A3

Fixed:

Fixed on B0 for external and brown-out resets.

2.7 Clearing the BORMIS interrupt status bit requires an extra write

Description:

Writing a 1 to the BORMIS bit in the **Masked Interrupt Status and Clear (MISC)** register does not immediately clear the BORRIS raw interrupt bit in the **Raw Interrupt Status (RIS)** register.

Workaround:

Write a 1 twice to the BORMIS bit to successfully clear the BORRIS raw interrupt bit.

Silicon Revision Affected:

A1, A3

Fixed:

Fixed in RevB.

2.8 Brown-out operation functions differently than specified

Description:

The BOR0 bit does not function in the **Brown-Out Reset Control (PBORCTL)** register. In addition, the brown-out reset triggered by the BOR1 bit is centered at 2.92 V for the rising edge and 2.90 V for the falling edge with a minimum of 2.8 V and a maximum of 3.1 V in both cases.

Also, the BOR0RIS and VDDARIS bits in the **Raw Interrupt Status (RIS)** register do not function. The BOR1RIS bit does properly indicates a BOR caused by the conditions in the above paragraph. The BOR0IMC and VDDAIMC in the **Interrupt Mask Control (IMC)** register, as well as the BOR0MISC and VDDAMISC bits in the **Masked Interrupt Status and Clear (MISC)** register are similarly affected.

Workaround:

None.

Silicon Revision Affected:

A1, A3

Fixed:

Fixed on B0.

2.9 Some devices may not start properly during power up

Description:

In very rare cases, the internal LDOs may not start properly during power up. If the LDOs do not start properly, the device may not begin operating, and V_{DDC} may not reach its specified levels.

Workaround:

Power cycle the device until the device starts up correctly. This issue has not been seen on devices when the VDD rise time from 0 V to 3.3 V is less than 100 µs. However, meeting this condition does not guarantee that the issue will not occur.

Silicon Revision Affected:

A1, A3, B0

Fixed:

Fixed on B1.

3 Hibernation

3.1 Hibernation write corruption on arbitrary power loss

Description:

A write to any configuration register in the Hibernation Module can be corrupted if the V_{DD} supply falls below the minimum operating voltage of 3.05 V while a write is in progress.

Workaround:

Use a voltage supervisor to assert an external reset at 3.05 V. The power-down transition between 3.05 V and 2.70 V must be at least 93 µs and must not have any points where it increases in voltage (must be monotonic).

Silicon Revision Affected:

A1, A3

Fixed:

Fixed on B0.

3.2 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:

A1, A3, B0

Fixed:

Fixed on B1.

4 ROM

4.1 USB boot loader in ROM does not operate correctly

Description:

The USB boot loader in ROM does not operate correctly.

Workaround:

To use the USB boot loader, load the StellarisWare version of the USB boot loader into Flash memory.

Silicon Revision Affected:

A1

Fixed:

Fixed on A3.

4.2 ROM_SysCtlClockSet() does not operate correctly with fractional dividers

Description:

The `ROM_SysCtlClockSet()` function in ROM does not operate correctly when using fractional dividers (such as SYSDIV_2_5). The function does work correctly with integer dividers.

Workaround:

If fractional clock dividers are used, load the StellarisWare version 8049 or later of `SysCtlClockSet()` into Flash memory and use that version of the function.

Silicon Revision Affected:

A1

Fixed:

Fixed on A3.

5 Flash memory

5.1 When a 1-kB Flash page is erased, the adjacent page is also erased

Description:

When a 1-kB Flash page is erased, the adjacent page in the even/odd pair is also erased. For example, if page 0 is erased, then page 1 is also erased. Similarly, if page 1 is erased, then page 0 is also erased.

Workaround:

None.

Silicon Revision Affected:

A1

Fixed:

Fixed on A3.

5.2 Flash memory page 0 and 1 may be erased if reset occurs during Flash memory erase operation

Description:

If a page erase command is issued to Flash memory and any type of system reset occurs before the erase operation starts, page 0 and 1 may be erased instead of the specified page.

Workaround:

None.

Silicon Revision Affected:

A1

Fixed:

Fixed on A3.

5.3 The KEY bit in the Boot Configuration (BOOTCFG) register does not function

Description:

The KEY bit in the **Boot Configuration (BOOTCFG)** register does not function, so the default value of 0xA442 must be used.

Workaround:

None.

Silicon Revision Affected:

A1, A3

Fixed:

Fixed on B0.

6 Flash memory, EEPROM

6.1 Reset during Flash memory program or erase or an EEPROM write causes Suspend state

Description:

If a non-POR reset ($\overline{\text{RST}}$ signal, brown out, software, watchdog, or MOSC failure) occurs when the Flash memory is being programmed or erased, or when the EEPROM is being written, any subsequent attempts to program or erase Flash memory or write to EEPROM fail. When this situation occurs, the Flash memory or the EEPROM is in the Suspend state. It is possible that a POR does not clear this condition.

Workaround:

The following code checks to see if the Flash memory or the EEPROM is in the Suspend state and clears it if necessary. This code should be run by an application during initialization and before any attempt to program or erase Flash memory or write to EEPROM.

```
tBoolean
FlashClearSuspend(void)
{
    unsigned long ulVal, ulSave;
    tBoolean bRetcode;

    //
    // Wait a while.
    //
    ROM_SysCtlDelay(10);
    ulSave = HWREG(0x400FD0FC);
    HWREG(0x400FD0FC) = 0x01000003;
    ROM_SysCtlDelay(10);
```

```
        //
        // Read flash controller status.
        //
        ulVal = HWREG(0x400AE054);

        //
        // Is the controller in the suspended state?
        //
        if(ulVal & 0x06)
        {
            //
            // Yes - clear the state.
            //
            HWREG(0x400AE288) = 0x05;
            HWREG(0x400AE20C) = 0x18;
            HWREG(0x400AE110) = 0;
            HWREG(0x400AE2B4) = 0x15;

            do
            {
                //
                // Poll for completion.
                //
                ulVal = HWREG(0x400AE054);
            }
            while(ulVal & 0x100);

    // NEW CODE
            HWREG(0x400AE050) = 0;
    // END NEW CODE
            HWREG(0x400AE2A4) = 0;
            HWREG(0x400AE2C0) = 0;
            HWREG(0x400AE2C4) = 0;
            HWREG(0x400AE2C8) = 0;
            HWREG(0x400AE2CC) = 0;
            HWREG(0x400AE2D0) = 0;
            HWREG(0x400AE2D4) = 0;
            HWREG(0x400AE2D8) = 0;
            HWREG(0x400AE2DC) = 0;
    // NEW CODE
            HWREG(0x400AE050) = 1;
            HWREG(0x400AE2C0) = 0;
            HWREG(0x400AE050) = 0;
    // END NEW CODE

            //
            // Tell the caller that we needed to clean up.
            //
            bRetcode = true;
        }
        else
        {
            //
            // No cleaning up was necessary.
```

```
        //
        bRetcode = false;
    }

    ROM_SysCtlDelay(10);
    HWREG(0x400FD0FC) = ulSave;
    ROM_SysCtlDelay(10);

    return(bRetcode);
}
```

Silicon Revision Affected:

A1

Fixed:

Fixed on A3.

7 EEPROM

7.1 EEPROM blocks must be accessed in alternate pairs to avoid corruption of data

Description:

When the words in a pair of EEPROM blocks are repetitively written, the words of the next pair of blocks get corrupted. In a given group of four blocks of EEPROM, for example, 0, 1, 2 and 3, repeated writes to either block 0 or block 1 cause the data in blocks 2 and 3 to be corrupted.

Workaround:

The EEPROM should be used only in alternate pairs of blocks 0,1,4,5,8,9, and so on, or 2,3,6,7,10,11, and so on.

Silicon Revision Affected:

A1

Fixed:

Fixed on A3.

7.2 EEPROM blocks 0 through 3 may be erased if reset occurs during an EEPROM write

Description:

If a write is issued to EEPROM and any type of system reset occurs before the write starts, blocks 0 through 3 may be erased.

Workaround:

Do not use blocks 0 through 3 in the EEPROM. Blocks 4 through 31 are available for EEPROM use.

Silicon Revision Affected:

A1

Fixed:

Fixed on A3.

7.3 If the EEPROM is programmed, both the Flash memory and EEPROM will be unable to be programmed after a POR

Description:

Due to a factory configuration error, if the EEPROM has been programmed and then a POR is executed, both the Flash memory and EEPROM will be unable to be programmed. Do not use the EEPROM on RevB0 devices.

Workaround:

None.

Silicon Revision Affected:

B0

Fixed:

Fixed on B1.

8 GPIO

8.1 Some GPIO register bits default to the incorrect state

Description:

The AFSEL bits for the following pins are set at reset, resulting in the pins defaulting to their alternate function:

- Port A[1:0]
- Port A[5:2]
- Port B[3:2]

This error in pin functionality may create pin conflict during any type of reset with the following signals:

Signal	Function	I/O	Level
PA0	U0Rx	Input	Tristate
PA1	U0Tx	Output	High
PA2	SSI0Clk	Output	Low
PA3	SSI0Fss	Output	High
PA4	SSI0Rx	Input	Tristate
PA5	SSI0Tx	Output	Low
PB2	I2C0SCL	Indeterminate[a]	Indeterminate[a]

Signal	Function	I/O	Level
PB3	I2C0SDA	Input	Tristate

a. While the pin is in an indeterminate state, it may be driving High or Low. When powering up, this pin is in an indeterminate state for 100 µs after V_{DD} reaches 3.0 V, at which point, PB2 is configured as an input and the level is tristate. If the pin has been operating in I^2C mode and any type of reset occurs, this pin holds its last driven state for 1 PIOSC clock after reset asserts, at which point, PB2 is configured as an input and the level is tristate.

In addition, the PMCx fields in the **GPIOPCTL** register for PD7 and PF0 default to 0x3.

Workaround:

To reconfigure the pins to their intended reset state (GPIO Input, GPIODEN =0), software must clear the corresponding bits in the **GPIOAFSEL** and **GPIODEN** registers for the associated pins. For pins PD7 and PF0, software must clear the corresponding AFSEL bits using the register commit control procedures described in the Commit Control section in the General-Purpose Input/Outputs chapter in the data sheet.

Note that PD7 and PF0 should be grounded, if possible, to prevent triggering an NMI. If that is not possible, an NMI handler must be implemented in case a High level is applied to PD7 or PF0 before they can be reconfigured.

For PD7, the PMC7 assignment of 0x3 is not valid, so it does not cause any issues. However, for PF0, the PMC0 assignment of 0x3 specifies CAN0Rx. If the system design requires CAN0Rx to be on another pin, the PMC0 field for Port F must be assigned to another function or cleared.

Silicon Revision Affected:

A1

Fixed:

Fixed on A3.

8.2 PB1 has permanent internal pull-up resistance

Description:

Regardless of its configuration, PB1 has an internal pull-up resistance that turns on when the voltage on the pin reaches approximately 3.3 V. Once turned on, the resistance remains in place even if the pin is driven Low.

Workaround:

When this pin is configured as an input, the external circuit must drive with an impedance less than or equal to 20 kΩ to provide enough drive strength to over-drive the internal pull-up and achieve the necessary V_{IL} voltage level.

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 instead of defaulting to an input. If a logic Low level is required after reset, a pull-down resistor of 20-kΩ or less should be connected. After reset, once the pin is re-configured as an output, the pin drives the programmed level.

Silicon Revision Affected:

A1

Fixed:

Fixed on A3.

8.3 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:

A1

Fixed:

Fixed on A3.

8.4 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:

A1, A3, B0

Fixed:

Fixed on B1.

9 ADC

9.1 The first ADC sample when using differential mode is incorrect

Description:

The first sample taken after the ADC is configured to operate in differential mode is incorrect. When using the continuous trigger, only the first sample is incorrect. When using other trigger sources, the first value after every trigger is incorrect.

Workaround:

When using the continuous or processor trigger, there is no workaround.

When using other trigger sources, configure Sample Sequencer 3 and an alternate Sample Sequencer in the same manner, but set the priority for SS3 to a higher level. In this configuration, SS3 captures the first, erroneous sample, and the alternate sample sequencer captures correct data for the sequence.

Silicon Revision Affected:

A1, A3

Fixed:

Fixed on B0.

9.2 Data may not be present in the FIFO at the time of the sequence interrupt or trigger

Description:

The interrupt or trigger for a sample sequence may occur before data is placed in the ADC sample sequence FIFO.

Workaround:

Insert a delay after receiving the interrupt or trigger and before reading the data in the FIFO. The minimum length of the delay is given by the following equation, where H is the number of samples to be averaged if hardware averaging is enabled (H=1 if hardware averaging is not used), and S is the sample rate:

```
Delay = H / S
```

For example, if sampling at a rate of 1 MSPS and 4 samples are to be hardware averaged, delay at least 4 µs before reading the data in the FIFO. The TivaWare API SysCtlDelay() can be used to add a delay based on your system clock frequency.

Silicon Revision Affected:

A1, A3

Fixed:

Fixed on B0.

9.3 The BUSY bit in the ADC Active Sample Sequencer (ADCACTSS) register does not function

Description:

Because the BUSY bit in the **ADC Active Sample Sequencer (ADCACTSS)** register does not function, if the ADC clock is gated or the system enters Sleep or Deep-Sleep mode after the ADC is triggered to start a conversion, the data in the sample sequencer may be corrupted.

Workaround:

Ensure that the wait for interrupt (WFI) instruction is used to enter sleep or deep-sleep mode, and not the wait for event (WFE) instruction or the SLEEPEXIT bit. Configure the last sample of the last sequence as the End of Sequence in the **ADC Sample Sequence Control n (ADCSSCTLn)** register and enable a sample interrupt for that sample. Before executing the WFI instruction, check to see if the End of Sequence interrupt is received, then disable all the configured sample sequencers by

clearing the corresponding ASENn bit(s) in the **ADC Active Sample Sequencer (ADCACTSS)** register. Delay for a number of clock cycles given by the following equation:

```
((# of previously configured samples) * (16 ADC clock cycles)) + 5 ADC clock cycles
```

This delay should be inserted to wait for any sampling activity to be completed. The ADC module will continue to be active in sleep or deep-sleep mode if configured in the **SCGCADC** or the **DCGCADC** registers.

Silicon Revision Affected:

A1, A3

Fixed:

Fixed on B0.

10 SSI

10.1 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:

A1, A3

Fixed:

Fixed on B0.

11 I2C

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

Description:

A glitch filter was added on Revision B1 to the I^2C 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^2C 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^2C signals.

Silicon Revision Affected:

A1, A3, B0

Fixed:

Fixed on B1.

12 Electrical Characteristics

12.1 Higher current than expected is consumed while V_{DD} ramps up until V_{DDC} crosses 1 V

Description:

While V_{DD} is ramping up, an excess 50 mA of current is consumed until V_{DDC} crosses 1 V. During this time, the output voltage on GPIO pins can go as high as 0.7 V.

Workaround:

None.

Silicon Revision Affected:

A1

Fixed:

Fixed on A3.

12.2 Nominal current consumption is 650 µA higher than specified

Description:

The POR oscillator is always enabled, causing higher current consumption than specified. This issue is noticeable primarily in Deep-sleep operation.

Workaround:

None.

Silicon Revision Affected:

A1

Fixed:

Fixed on A3.

12.3 V_{DD} inrush current of up to 500 mA is seen while V_{DD} ramps up

Description:

V_{DD} inrush current of up to 500 mA is seen while V_{DD} ramps up due to the on-chip LDO regulator charging the LDO and V_{DDC} capacitors. Expected inrush current should be between 50 and 250 mA.

Workaround:

Ensure that the V_{DD} power supply has sufficient output capacitance to supply up to 500 mA for approximately 100 µs.

Silicon Revision Affected:

A1

Fixed:

Fixed on A3.

Texas Instruments Incorporated
108 Wild Basin, Suite 350
Austin, TX 78746
http://www.ti.com/tiva-c
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