Stellaris® LM3S Tempest- and Firestorm-Class Microcontrollers Errata

Silicon Errata

Literature Number: SPMZ861
July 2014

Contents

Stellaris® LM3S Tempest- and Firestorm-class Microcontrollers Errata

1 Introduction

This document describes known exceptions to the functional specifications for the Stellaris Tempest-class Rev C5 and C3, and the Firestorm-class Rev A2 and A1 microcontrollers. See also the ARM® Cortex™-M3 errata, SPMZ092.

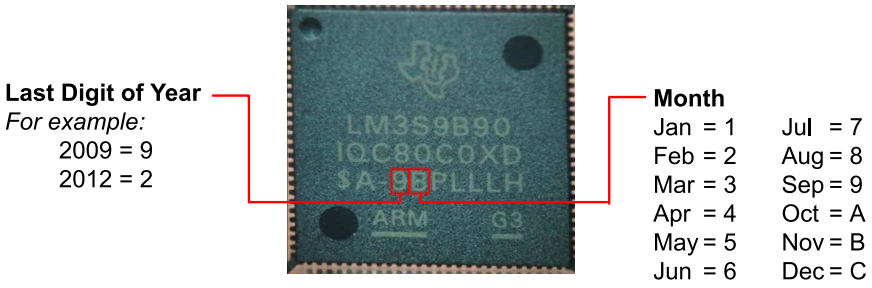
Note that each Stellaris microcontroller may not have all peripherals so not all errata may apply to your device.

For product details on the microcontrollers, see:

- Tempest-Class MCUs
- Firestorm-Class MCUs

2 Device Date Code

To determine the date code of your part, look at the third line in the part markings, at the fourth and fifth characters following the dash (outlined in red below). The first number after the dash indicates the last decimal digit of the year. The next character indicates the month, in hexadecimal. So, in the below example, the 9B indicates a date code of November 2009.



3 Advisory to Silicon Revision Correlation

Table 1. Advisory to Silicon Revision Matrix

Advisory Number	Advisory Title	Silicon Revision(s) Affected			
		Tempest Rev C3	Tempest Rev C5	Firestorm Rev A1	Firestorm Rev A2
ADC					
LM3ADC#03	ADC hardware averaging produces erroneous results in differential mode	X	X		
LM3ADC#04	Retriggering a sample sequencer before it has completed the current sequence results in continuous sampling	X	X	X	X
LM3ADC#08	Differential pair encodings are incorrect	X	X		
LM3ADC#09	Digital comparator in last step of sequence does not trigger or interrupt	X	X	X	X
LM3ADC#10	Digital comparator interrupts do not trigger or interrupt as expected	X	X	X	X
LM3ADC#11	Missing trigger or interrupt when multiple sequences configured for processor trigger and different trigger	X	X	X	X
LM3ADC#12	ADC sample sequencers priorities are different than expected	X	X	X	X
LM3ADC#13	Simultaneous sampling on both ADC modules yields incorrect samples	X	X	X	X
LM3ADC#14	Phase offset does not delay as expected if sample sequencers are not triggered at the same time	X	X	X	X
LM3ADC#15	The encoding of the VREF field in the ADCCTL register is incorrect			X	X
LM3ADC#16	The ADC external reference pin does not function	X	X		
LM3ADC#17	ADC errors exceed specifications			X	
LM3ADC#18	Data may not be present in the FIFO at the time of the sequence interrupt or trigger	X	X	X	X
LM3ADC#19	The first two ADC samples may be incorrect	X	X	X	X
DMA					
LM3DMA#02	The µDMA controller fails to generate capture mode DMA requests from Timer A in the Timer modules	X	X	X	X
LM3DMA#03	The µDMA does not generate a completion interrupt when transferring to and from GPTM 2A and 2B	X	X	X	X
Electrical Characteristics					
LM3ELEC#02	First two ADC samples from the internal temperature sensor must be ignored	X	X	X	X
EPI					
LM3EPI#01	At EPI clock speeds over 15 MHz, SDRAM initialization delay is not long enough	X	X	X	X
LM3EPI#02	In Host-Bus 16 mode, only one byte select is asserted if only 8 bits are read	X	X	X	X
LM3EPI#03	When non-blocking reads are pending, EPI accesses can cause the NBRFIFO counter to be incorrectly decremented	X	X	X	X
LM3EPI#04	In General-Purpose mode, the framing signal is output regardless of the state of the FRMPIN bit	X	X	X	X
LM3EPI#05	In General-Purpose mode, the read and write strobes are output regardless of the state of the RW bit	X	X	X	X
LM3EPI#06	In General-Purpose mode, the maximum time to wait for the iRDY signal is derived from the system clock, not the EPI clock	X	X	X	X

Table 1. Advisory to Silicon Revision Matrix (continued)

Advisory Number	Advisory Title	Silicon Revision(s) Affected			
		Tempest Rev C3	Tempest Rev C5	Firestorm Rev A1	Firestorm Rev A2
LM3EPI#07	iRDY timing in General-Purpose mode is not as specified in the data sheet	X	X	X	X
LM3EPI#08	Chip select operation is not correct when using dual chip selects in Host Bus Continuous Read mode	X	X	X	X
Ethernet Controller					
LM3ETH#01	Encoding error in the Ethernet MAC LED Encoding (MACLED) register	X	X	X	X
GPIO					
LM3GPIO#05	PB1 has permanent internal pull-up resistance	X	X	X	X
General-Purpose Timers					
LM3GPTM#03	The General-purpose timer match register does not function correctly in 32-bit mode	X	X	X	X
LM3GPTM#04	A spurious DMA request is generated when the timer rolls over in Input-Edge Time mode	X	X	X	X
LM3GPTM#05	A spurious DMA request is generated when the timer rolls over the 16-bit boundary	X	X	X	X
LM3GPTM#06	The value of the prescalar register is not readable in Edge-Count mode	X	X	X	X
LM3GPTM#07	ADC trigger and Wait-on-Trigger may assert when the timer is disabled	X	X	X	X
LM3GPTM#08	Wait-on-Trigger does not assert unless the TnOTE bit is set	X	X	X	X
LM3GPTM#09	Do not enable match and timeout interrupts in 16-bit PWM mode	X	X	X	X
LM3GPTM#10	Do not use µDMA with 16-bit PWM mode	X	X	X	X
LM3GPTM#11	Writing the GPTMTnV register does not change the timer value when counting up	X	X	X	X
LM3GPTM#12	The prescalar does not work correctly when counting up in periodic or one-shot mode	X	X	X	X
LM3GPTM#13	Snapshot must be enabled in both Timer A and B when in 32-bit snapshot mode	X	X	X	X
LM3GPTM#14	Writes to some General-Purpose Timer registers cause the counter to increment and decrement in some cases	X	X	X	X
Hibernation Module					
LM3HIB#07	Writes to certain Hibernation module registers sometimes fail	X	X		
LM3HIB#08	Writes to Hibernation module registers may change the value of the RTC	X	X	X	X
LM3HIB#09	VDD3ON mode may not be used	X	X	X	X
LM3HIB#10	The WRC bit in the Hibernation Control register is R/W	X	X	X	X
LM3HIB#11	Hibernation Module 4.194304-MHz oscillator supports a limited range of crystal load capacitance values	X	X	X	X
JTAG and Serial Wire Debug					
LM3JTAG#01	JTAG INTEST instruction does not work	X	X	X	X
LM3JTAG#04	Boundary scan is not functional	X	X	X	X
Memory					
LM3MEM#04	Deep Sleep mode must not be used	X	X	X	X
LM3MEM#05	Mass erase must not be used if flash protection bits are used	X	X	X	X

Table 1. Advisory to Silicon Revision Matrix (continued)

Advisory Number	Advisory Title	Silicon Revision(s) Affected			
		Tempest Rev C3	Tempest Rev C5	Firestorm Rev A1	Firestorm Rev A2
LM3MEM#06	Page erase or program must not be performed on a protected flash page	X	X	X	X
LM3MEM#07	Flash memory endurance cycle specification is 100 cycles	X	X	X	X
LM3MEM#08	Flash memory may be corrupted if programmed at system clock speeds above 50 MHz	X	X	X	X
LM3MEM#09	The ROM_FlashProgram() function may not correctly program the flash memory above 50 MHz			X	X
LM3MEM#10	Some ROM functions are incorrect	X		X	
LM3MEM#11	The BOOTCFG register cannot be reliably written until after a special mass erase	X		X	
LM3MEM#12	Flash corruption or device failure may occur at power on	X		X	
LM3MEM#13	Flash memory corruption may occur when device is unpowered and stored for several months	X	X	X	X
LM3MEM#14	Flash-resident ECC logic can cause Flash memory corruption	X	X	X	X
PWM					
LM3PWM#08	PWM fault latch does not operate correctly	X	X	X	X
LM3PWM#09	Under certain circumstances, the PWM load interrupt is triggered as soon as the PWM is enabled	X	X	X	X
LM3PWM#10	Setting the PWMSYNC bits may not synchronize the PWM counters if PWMDIV is used	X	X	X	X
QEI					
LM3QEI#03	When using the index pulse to reset the counter, a specific initial conditions in the QEI module causes the direction for the first count to be misread	X	X	X	X
SSI					
LM3SSI#01	Freescale SPI mode at low SSIClk frequencies can yield data corruption	X	X	X	X
LM3SSI#02	SSI Receive FIFO Time-out interrupt may assert sooner than expected in slave	X	X	X	X
System Control					
LM3SYSCTL#14	For 64-pin devices, the CLASS field in Device Identification 0 (DID0) register is incorrect	X	X	X	X
LM3SYSCTL#15	The MINOR field in Device Identification 0 (DID0) register is incorrect	X	X		
LM3SYSCTL#16	Debug interface is reset by any type of reset	X	X	X	X
LM3SYSCTL#17	JTAG state machine may advance after certain resets	X	X	X	X
LM3SYSCTL#18	Non-word-aligned write to SRAM can cause incorrect value to be loaded	X	X	X	X
LM3SYSCTL#19	Internal reset supervisors may not prevent incorrect device operation during power transitions	X	X	X	X
LM3SYSCTL#20	The PIOSC cannot be calibrated by the user	X	X	X	X
LM3SYSCTL#21	ROM boot loaders do not function properly when programming blank flash memory			X	X
LM3SYSCTL#22	ROM_UpdateEthernet() is not functional for firmware update			X	X
LM3SYSCTL#23	Brown-out interrupt is never triggered	X		X	
LM3SYSCTL#24	DSDIVORIDE value of 0x1 does not divide Deep Sleep clock by 2	X	X	X	X
UART					

Table 1. Advisory to Silicon Revision Matrix (continued)

Advisory Number	Advisory Title	Silicon Revision(s) Affected			
		Tempest Rev C3	Tempest Rev C5	Firestorm Rev A1	Firestorm Rev A2
LM3UART#01	The RTRIS bit in the UARTRIS register is only set when the interrupt is enabled	X	X	X	X
LM3UART#02	LIN mode Sync Break does not have the correct length	X	X	X	X
LM3UART#03	When UART LIN or SIR mode is enabled, µDMA burst transfer does not occur	X	X	X	X
USB					
LM3USB#05	USB Host controller may not be used to communicate with a low-speed Device when connected through a hub	X	X	X	X
LM3USB#08	USB paired JK jitter compliance test requires automatic waiver	X	X		
LM3USB#09	USB low-speed crossover voltage compliance test requires automatic waiver	X	X	X	X
LM3USB#10	Special considerations for PB1	X	X	X	X
LM3USB#11	USB0DM may be driven after reset	X	X	X	X
LM3USB#12	MCU may fail USB certification if the EPI module is operating	X	X	X	X
LM3USB#13	USB compliance test issue: USB full-speed, far-end signal compliance tests fail with 5 m cable	X		X	
LM3USB#14	USB compliance test issue: USB embedded host low-speed, far-end signal compliance tests fail	X		X	
LM3USB#15	USB controller sends EOP at end of device Remote Wake-Up	X	X	X	X
LM3USB#16	Device sends SE0 in response to a USB bus reset	X	X	X	X
LM3USB#17	USB Resume occasionally does not wake device from Deep Sleep	X	X	X	X
Watchdog Timers					
LM3WDT#01	Writes to Watchdog Timer 1 module WDTLOAD register sometimes fail	X	X		
LM3WDT#02	Watchdog clear mechanism described in the data sheet does not work for the Watchdog Timer 1 module	X	X	X	X
LM3WDT#03	Watchdog Timer 1 module asserts reset signal even if not programmed to reset	X	X	X	X
LM3WDT#04	WDTLOAD yields an incorrect value when read back	X	X	X	X
LM3WDT#05	The Watchdog Load (WDTLOAD) register cannot be changed when using a debugger while the STALL bit is set	X	X	X	X
LM3WDT#06	Reading the WDTVALUE register may return incorrect values when using Watchdog Timer 1	X	X	X	X
LM3WDT#07	Watchdog timer reloads on any write to the Watchdog Interrupt Clear (WDTICR) register	X	X	X	X
LM3WDT#08	Watchdog timer will not issue reset if the Watchdog Test (WDTTEST) register is incorrectly programmed	X	X	X	X
LM3WDT#09	The Watchdog Test (WDTTEST) register can be changed even when the registers are locked	X	X	X	X

4 Tempest-class Rev C5 and Firestorm-class Rev A2 Known Design Exceptions to Functional Specifications

The following issues affect Stellaris Tempest-Class Rev C5 and Firestorm-Class Rev A2 devices.

Table 2. Advisory List

Table 2. Advisory List (continued)

Table 2. Advisory List (continued)

LM3ADC#03 *ADC hardware averaging produces erroneous results in differential mode*

Device(s) Affected: Stellaris Tempest-class Rev C5

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 0.0 V.

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

LM3ADC#04 *Retriggering a sample sequencer before it has completed the current sequence results in continuous sampling*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

LM3ADC#08 *Differential pair encodings are incorrect*

Device(s) Affected: Stellaris Tempest-class Rev C5

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(s): 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
AIN10 and AIN11	5	0x9
AIN12 and AIN13	6	0xA
AIN14 and AIN15	7	0xB

LM3ADC#09 *Digital comparator in last step of sequence does not trigger or interrupt*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): 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.

LM3ADC#10 *Digital comparator interrupts do not trigger or interrupt as expected*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): 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.

LM3ADC#11 *Missing trigger or interrupt when multiple sequences configured for processor trigger and different trigger*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): None.

LM3ADC#12 *ADC sample sequencers priorities are different than expected*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): 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.

LM3ADC#13 *Simultaneous sampling on both ADC modules yields incorrect samples*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): 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.

LM3ADC#14 *Phase offset does not delay as expected if sample sequencers are not triggered at the same time*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): 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.

LM3ADC#15 *The encoding of the VREF field in the ADCCTL register is incorrect*

Device(s) Affected: Stellaris Firestorm-class Rev A2

Description: The encoding of the VREF field in the **ADC Control (ADCCTL)** register is incorrect.

Workaround(s): Use VREF = 0x3 to select the external voltage reference VREFA.

Use VREF = 0x1 to select the external voltage reference VREFA/3.

LM3ADC#16 *The ADC external reference pin does not function*

Device(s) Affected: Stellaris Tempest-class Rev C5 LM3S1N11, LM3S1J11, LM3S1811

Description: The ADC external reference pin, VREFA, does not function on the LM3S1N11, LM3S1J11, and LM3S1811 microcontrollers.

Workaround(s): None. Fixed on devices with date codes of 17 (July 2011) or later. See Section 2, Device Date Code, for more information.

LM3ADC#18 *Data may not be present in the FIFO at the time of the sequence interrupt or trigger*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

Workaround(s): 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 StellarisWare API SysCtlDelay() can be used to add a delay based on your system clock frequency.

LM3ADC#19 *The first two ADC samples may be incorrect*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description The first two ADC samples taken after the ADC clock is enabled in the **xCGC0** register may be incorrect.

Workaround(s) Reset the ADC peripheral using the **SRCR0** register after the ADC peripheral clock is enabled and before initializing the ADC and enabling the sample sequencer.

LM3DMA#02 *The µDMA controller fails to generate capture mode DMA requests from Timer A in the Timer modules*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): Use Timer B.

LM3DMA#03 *The µDMA does not generate a completion interrupt when transferring to and from GPTM 2A and 2B*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): If a completion interrupt is required, use an alternate GPTM.

LM3ELEC#02 *First two ADC samples from the internal temperature sensor must be ignored*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

Workaround(s): 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.

LM3EPI#01 *At EPI clock speeds over 15 MHz, SDRAM initialization delay is not long enough*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

LM3EPI#02 *In Host-Bus 16 mode, only one byte select is asserted if only 8 bits are read*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

LM3EPI#03 *When non-blocking reads are pending, EPI accesses can cause the NBRFIFO counter to be incorrectly decremented*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): 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.

LM3EPI#04 *In General-Purpose mode, the framing signal is output regardless of the state of the FRMPIN bit*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

LM3EPI#05 *In General-Purpose mode, the read and write strobes are output regardless of the state of the RW bit*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

LM3EPI#06 *In General-Purpose mode, the maximum time to wait for the iRDY signal is derived from the system clock, not the EPI clock*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

LM3EPI#07 *iRDY timing in General-Purpose mode is not as specified in the data sheet*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): 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 where the iRDY signal may be asserted ((25 ns – 12.5 ns) – 10 ns, where 10 ns is the iRDY assertion or deassertion set up time (TRDYSU)). Figure 1 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.

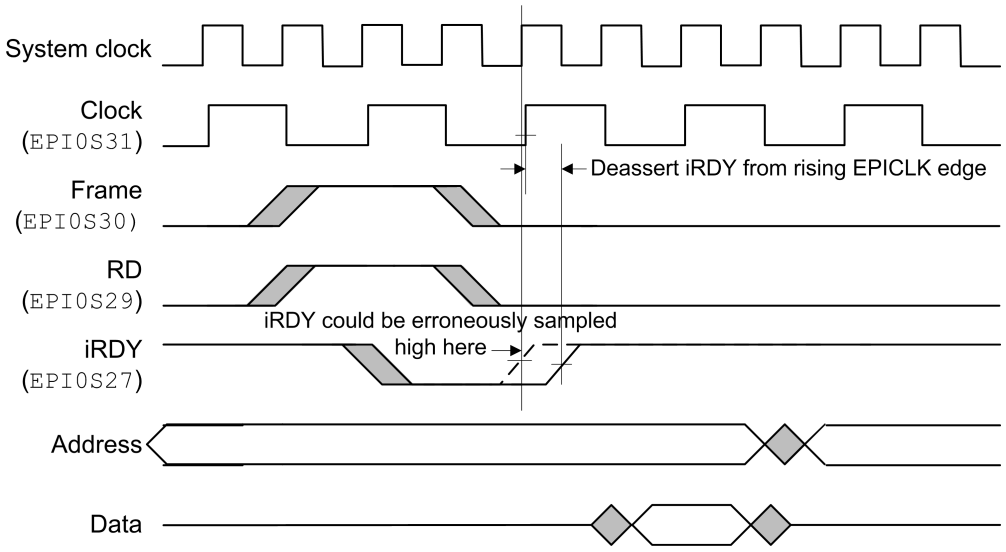


Figure 1. iRDY Timing

LM3EPI#08 *Chip select operation is not correct when using dual chip selects in Host Bus Continuous Read mode*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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 2 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(s): 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 RDWS field to be 0x2 (6 clocks) to account for the 1 clock difference in the strobe width.

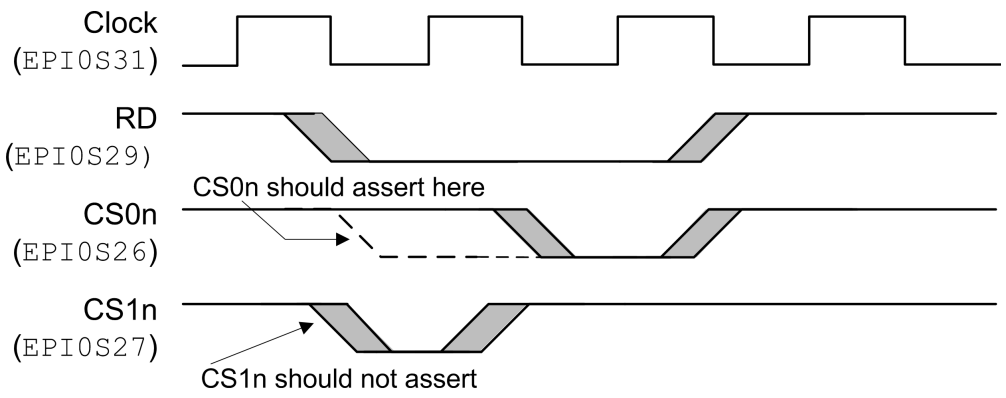


Figure 2. Chip Selects

LM3ETH#01 *Encoding error in the Ethernet MAC LED Encoding (MACLED) register*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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 (that is, exactly the same as encoding 0x1).

Workaround(s): None.

LM3GPIO#05 *PB1 has permanent internal pull-up resistance*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: Regardless of its configuration (GPIO or alternate digital function), PB1 has a maximum internal pull-up resistance of 800 Ω 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(s): 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.

LM3GPTM#03 *The General-purpose timer match register does not function correctly in 32-bit mode*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): None.

LM3GPTM#04 *A spurious DMA request is generated when the timer rolls over in Input-Edge time mode*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

Workaround(s): 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.

LM3GPTM#05 *A spurious DMA request is generated when the timer rolls over the 16-bit boundary*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): Only use DMA with a 16-bit periodic timer.

LM3GPTM#06 *The value of the prescalar register is not readable in Edge-Count mode*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): None.

LM3GPTM#07 *ADC trigger and Wait-on-Trigger may assert when the timer is disabled*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): 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.

LM3GPTM#08 *Wait-on-Trigger does not assert unless the TnOTE bit is set*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

Workaround(s): 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.

LM3GPTM#09 *Do not enable match and timeout interrupts in 16-Bit PWM mode*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

Workaround(s): Ensure that any unwanted interrupts are masked in the **GPTMTnMR** and **GPTMIMR** registers.

LM3GPTM#10 *Do not use µDMA with 16-bit PWM mode*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

Workaround(s): Do not use µDMA to transfer data when the timer is in 16-bit PWM mode.

LM3GPTM#11 *Writing the GPTMTnV register does not change the timer value when counting up*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

Workaround(s): None.

LM3GPTM#12 *The prescalar does not work correctly when counting up in periodic or one-shot mode*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

Workaround(s): Do not use the prescaler when counting up in 16-bit periodic or snap-shot mode.

LM3GPTM#13 *Snapshot must be enabled in both timer A and B when in 32-bit snapshot mode*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

Workaround(s): 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.

LM3GPTM#14 *Writes to some General-Purpose Timer registers cause the counter to increment and decrement in some cases*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: Writes to the following registers when the timer is enabled cause the counter to increment in up count mode and decrement in down count mode when incrementing or decrementing the counter inside the General-Purpose timers:

- **GPTM Timer n Match (GPTMTnMATCHR)**
- **GPTM Timer n Prescale (GPTMTnPR)**

Situations in which the counter is incremented or decremented include:

- RTC Mode
- Input edge count mode

Workaround(s): None.

LM3HIB#07 *Writes to certain Hibernation module registers sometimes fail*

Device(s) Affected: Stellaris Tempest-class Rev C5

Description: Due to a synchronization issue with the independent clock domain of the Hibernation module, writes to certain registers may sometimes fail, even though the WRC bit in the HIBCTL register is set after the write occurs. Registers affected include **HIBRTCC, HIBRTCM0, HIBRTCM1, HIBRTCLD, HIBRTCT**, and **HIBDATA**.

Workaround(s): After performing a write to the listed Hibernation module registers or non-volatile memory, read the contents back and verify that they are correct. If they are incorrect, perform the write operation again.

LM3HIB#08 *Writes to Hibernation module registers may change the value of the RTC*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: If the Hibernation module's RTC counter is active, any write to certain Hibernation module registers that occurs while the RTC counter is changing from the current value to the next can cause corruption of the RTC counter stored in the **HIBRTCC** register. Registers affected are: **HIBRTCC, HIBRTCM0, HIBRTCM1, HIBRTCLD, HIBRTCT**, and **HIBDATA**.

Workaround(s): The user application must guarantee that writes to the affected Hibernation module registers cannot occur on the RTC counter boundary. Any initial configuration of the affected Hibernation module registers must be done before enabling the RTC counter.

There are two ways to update affected Hibernation Module registers after initial configuration:

1. Use the Hibernation RTC match interrupt to perform writes to the affected Hibernation module registers. Assuming the interrupt is guaranteed to be serviced within 1 second, this technique provides a mechanism for the application to know that the RTC update event has occurred and that it is safe to write data to the affected Hibernation module registers. This method is useful for applications that don't require many writes to Hibernation module registers.
2. Set up a secondary time-keeping resource to indicate when it is safe to perform writes to the affected Hibernation module registers. For example, use a general purpose timer in combination with the Hibernation RTC match interrupt. In this scenario, the RTC match interrupt is used to both update the match register value and enable the general purpose timer in one-shot mode. The timer must be configured to have a maximum time-out period of less than 1 second. In this

configuration, a global variable is used to indicate that it is safe to perform writes to the affected Hibernation module registers. When the one-shot timer times out, the timer interrupt updates the global variable to indicate that writes are no longer safe. This procedure is repeated on every RTC match interrupt.

LM3HIB#09 *VDD3ON mode may not be used*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: The VDD3ON mode may not be used.

Workaround(s): None. Do not use the VDD3ON mode to enter hibernation.

LM3HIB#10 *The WRC bit in the Hibernation control register is R/W*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: The WRC bit in the **Hibernation Control (HIBCTL)** register can be written. This bit should be a read-only bit.

Workaround(s): Wait until the WRC bit is set before writing to the **HIBCTL** register. Always use a read-modify-write sequence when writing to the register to avoid changing the state of the WRC bit. Changing the value of the WRC bit can cause improper operation.

LM3HIB#11 *Hibernation module 4.194304-MHz oscillator supports a limited range of crystal load capacitance values*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: For some 4.194304-MHz crystals, the manufacturer-recommended crystal value may be outside of the capabilities of the Hibernate module oscillator. If the crystal manufacturer's recommended load capacitance is used, the hibernate oscillator may fail to start.

For a parallel-resonant oscillator circuit, the total load capacitance C_L (as specified by the manufacturer) is calculated as follows:

$$C_L = (C_1 * C_2) / (C_1 + C_2) + C_S \quad (1)$$

Due to the workaround, C_1 and C_2 are limited to 20 pF. Using 3 pF for stray capacitance (C_S), the formula above shows that a crystal with C_L of 13 pF is the highest value supported due to this erratum. Refer to the crystal datasheet to determine which crystals have an acceptable load capacitance range.

Workaround(s): Use load capacitors of 20 pF or less (18 pF is typical). Note that for some crystals, this value may pull the oscillator frequency slightly away from the crystal manufacturer's specified accuracy. Your crystal manufacturer can provide this information.

Alternatively, use an external 32.768-kHz oscillator as the source for the Hibernation module clock.

LM3JTAG#01 *JTAG INTEST instruction does not work*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

Workaround(s): None.

LM3JTAG#04 *Boundary scan is not functional*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: The boundary scan is not functional on this device.

Workaround(s): None.

Fixed on devices with data codes of 1A (October 2011) or later. See Section 2, Device Date Code, for more information.

LM3MEM#04 *Deep Sleep mode must not be used*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: Deep Sleep mode must not be used.

Due to this erratum, the use of this device in USB bus-powered applications is prohibited because sleep mode current consumption exceeds the USB specification.

Workaround(s): Use Sleep or Hibernation mode.

LM3MEM#05 *Mass erase must not be used if flash protection bits are used*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: The mass erase function using the MERASE bit in the **Flash Memory Control (FMC)** register must not be used in systems that clear any of the **Flash Memory Protection Program Enable n (FMPPEn)** bits. Mass erase can be used as long as none of the **FMPPEn** bits are cleared.

Workaround(s): Erase Flash memory with the page erase function using the ERASE bit in the **FMC** register instead of the mass erase function.

LM3MEM#06 *Page erase or program must not be performed on a protected flash page*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: The erase function using the ERASE bit in the **Flash Memory Control (FMC)** register and the program function using the WRITE bit in the **FMC** register or the WRBUF bit in the **FMC2** register must not be used in systems that clear the bit in **FMPPEn** that corresponds to that page of Flash. Erase and program can be used as long as neither of the corresponding **FMPPEn** bits are cleared.

Workaround(s): Only erase and program memory that is not protected by the corresponding **FMPPEn** bits.

LM3MEM#07 *Flash memory endurance cycle specification is 100 cycles*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): 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.

LM3MEM#08 *Flash memory may be corrupted if programmed at system clock speeds above 50 MHz*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

Workaround(s): 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.

LM3MEM#09 *The ROM_FlashProgram() function may not correctly program the flash memory above 50 MHz*

Device(s) Affected: Stellaris Firestorm-class Rev A2

Description: The ROM_FlashProgram() function may not correctly program the Flash memory when the system clock is above 50 MHz. As a result, the ROM boot loader may not function at system clock speeds above 50 MHz.

Workaround(s): When using the ROM_FlashProgram() function, ensure that the system clock frequency is no higher than 50 MHz or load the StellarisWare version of FlashProgram() into Flash memory and use that version of the function. When invoking the ROM boot loader, do not use system clock frequencies above 50 MHz.

LM3MEM#13 *Flash memory corruption may occur when device is unpowered and stored for several months*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: Customer returns have indicated unexpected flash corruption when final product is programmed at the factory and stored, unpowered, on the shelf for several months after programming.

Dielectrics used for silicon Flash bit cells will form “trap” sites when current is tunneled through the dielectric. A trap is an alteration of the SiO_2 structure where it is either missing an atom or has a stretched atom bond. In each Flash cell, these traps can allow electrons to tunnel from the floating gate to the trap and then from the trap to the substrate when the distances are short enough. Physically aligned traps in the dielectric can cause the programmed value of the cell to be leaked. The thinner the dielectric, the fewer traps it takes to form a leakage path. In the 130 nm Flash IP for this device family, the storage dielectric is the same thickness as the logic dielectric (about 60 angstroms).

The Error Correction Circuitry (ECC) has been included as part of the Flash memory to compensate for aligned dielectric traps that cause leaked programmed values. Each 32-bit data word has 6 extra bits of redundant information to allow for single-bit error correction. The 6 ECC bits are generated on each program operation (write or read) to the Flash memory to store the correct encoding for the specific data word. The ECC logic within the Flash IP provides status flags when corrections are made for a read with a single-bit error.

If more than 1 bit of the 38 bits stored (32 data bits + 6 ECC bits) for a 32-bit data word has leaked to a point where an incorrect value is returned, the built-in ECC logic will not be able to correct the incorrect values and the incorrect word will be sent to the CPU at the start of the device.

When power is continuously applied to the device and code is running from Flash, the ECC logic has a real-time monitor function that scans the entire flash every 24 hours of time powered on to detect and repair any single-bit shelf storage errors. The intent of this procedure is to keep errors from accumulating while continuously powered on.

Due to the storage oxide thickness and trap-assisted electron tunneling, there are more leaked cell values than originally expected in the Flash design within several months after programming. As a result, the ECC logic is not able to repair all of the errors in the Flash memory.

Data derived from customer returns predicts that devices that are left unpowered at room temperature storage for 6 months can result in failure rates of 2000-3000 DPPM per year.

Workaround(s): There is no workaround for this issue, but the impact can be minimized by programming the flash memory close to the start of when the system will be used. Please contact your local TI representative for further assistance to help assess this erratum's impact to your application.

LM3MEM#14 *Flash-resident ECC logic can cause Flash memory corruption*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: TI's internal accelerated life testing of the flash Error Correction Circuitry (ECC) has indicated unexpected flash corruption.

The Error Correction Circuitry (ECC) has been included as part of the Flash memory to identify and correct leaked program values in the Flash memory. Each 32-bit data word has 6 extra ECC bits of redundant information to allow for single-bit error correction. The 6 ECC bits are generated on each program operation (write or read) to the Flash memory to store the correct encoding for a specific data word. The ECC logic within the Flash IP provides status flags when corrections are made for a read with a single-bit error.

When power is applied to the device, the ECC monitor function scans the entire Flash in 32kB blocks every 24 hours and will re-program any single-bit in a word that has "leaked". This monitor function is a real-time hardware state machine that runs concurrently with the CPU application without ever stalling the CPU.

When power is applied to the device, the ECC monitor function scans the entire Flash in 32kB blocks every 24 hours and will re-program any single-bit in a word that has "leaked". This monitor function is a real-time hardware state machine that runs concurrently with the CPU application without ever stalling the CPU.

TI has done time-accelerated testing of the ECC monitor function to represent years of scans of the Flash memory. Results indicate that a combination of on-chip noise, noise in the power supply, and system noise can create a condition where the reads performed by the ECC monitor while scanning the Flash memory provide incorrect data. When attempting to make repairs to a 32-bit data word, the ECC monitor may read incorrect data from the Flash memory and re-program the word with this incorrect data, thereby corrupting the 32-bit word.

Bench testing of 10-year accelerated powered operation indicates potential device failure rates of 1000-3000 DPPM per year.

Workaround(s): There is no workaround for this issue, but the impact can be minimized by reducing the amount of noise in the system and in the device power supply. Please contact your local TI representative for further assistance to help assess this erratum's impact to your application.

LM3PWM#08 *PWM fault latch does not operate correctly*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

Workaround(s): 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.

LM3PWM#09 *Under certain circumstances, the PWM load interrupt is triggered as soon as the PWM is enabled*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: A spurious PWM interrupt occurs immediately when the PWM is enabled under the following conditions:

- The PWM Load register contains a nonzero value and
- Either of the PWM Compare registers contains a value less than the value in the PWM Load register and
- PWM interrupts are enabled.

Workaround(s): None.

LM3PWM#10 *Setting the PWMSYNC bits may not synchronize the PWM counters if PWMDIV is used*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: The bits in the **PWM Time Base Sync (PWMSYNC)** register are used to synchronize the counters in the PWM generators. The PWMDIV field in the **PWM Clock Configuration (PWMCC)** register is used to specify a fractional version of the system clock to use for the counters. If the PWMSYNC bits are set when the PWMDIV field is configured to anything other than 0x0, the counters may not be synchronized.

Workaround(s): None.

LM3QEI#03 *When using the index pulse to reset the counter, a specific initial conditions in the QEI module causes the direction for the first count to be misread*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: When using the index pulse to reset the counter with the following configuration in the **QEI Control (QEICTL)** register:

- SIGMODE is 0 indicating quadrature mode
- CAPMODE is 1 indicating both PhA and PhB edges are counted

and the following initial conditions:

- Both PhA and PhB are 0
- The next quadrature state is in the counterclockwise direction

the QEI interprets the state change as an update in the clockwise direction, which results in a position mismatch of 2.

Workaround(s): None.

LM3SSI#01 *Freescale SPI mode at low SSIClk frequencies can yield data corruption*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): 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.

LM3SSI#02 *SSI Receive FIFO Time-out interrupt may assert sooner than expected in slave*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: The SSI receive FIFO time-out interrupt may assert sooner than 32 system clock periods in slave mode if the CPSDVSR field in the **SSI Clock Prescale (SSICPSR)** register is set to a value greater than 0x2. Master mode is not affected by this behavior.

Workaround(s): In some cases, software can use the SCR field in the **SSI Control 0 (SSICR0)** register in combination with a CPSDVSR field value of 0x2 to attain the same SSI clock frequency. For example, if the desired serial clock rate is SysClk/48, then CPSDVSR = 0x2 and SCR = 0x17 can be used instead of CPSDVSR = 0x18 and SCR = 0x1 to achieve the same clock rate, using the equation SSInCLK = SysClk / (CPSDVSR * (1 + SCR)). If there is not a value of SCR that can be used with CPSDVSR = 0x2 to attain the required serial clock rate, then the receive FIFO time-out feature cannot be used.

LM3SYSCTL#14 *For 64-pin devices, the CLASS field in Device Identification 0 (DID0) register is incorrect*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: The CLASS field, bits[23:16], in the **Device Identification 0 (DID0)** register is incorrect.

For Tempest-class microcontrollers, the CLASS field should be 0x04; for Firestorm-class microcontrollers, the CLASS field should be 0x06. Instead, the field reads as 0x03 for both classes.

Workaround(s): None.

LM3SYSCTL#15 *The MINOR field in Device Identification 0 (DID0) register is incorrect*

Device(s) Affected: Stellaris Tempest-class Rev C5

Description: The MINOR field, bits[7:0], in the **Device Identification 0 (DID0)** register is incorrect. The MINOR field should be 0x05 indicating the fifth metal layer change. Instead, the field reads as 0x04.

Workaround(s): None.

LM3SYSCTL#16 *Debug interface is reset by any type of reset*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: The Serial Wire JTAG Debug Port (SWJ-DP) is reset by any reset condition. Therefore, any access to a debugger is lost, including breakpoints, watchpoints, vector catch, and trace. These reset types include:

- Watchdog reset
- Brown-out reset
- Software reset
- Reset pin assertion
- Main oscillator fail

Normal operation of the device is not affected by the reset of the SWJ-DP, however, users should bear this functionality in mind during development and debugging of applications. If a debugger does a SYSRESREQ, or if the debugger is being used in a session and a system reset occurs due to one of the reset sources above, then the debugger loses its state, including breakpoints, watchpoints, vector catch, and trace. Most debuggers attempt a recovery, usually after reporting the error to the user. If the debugger is able to recover control, the state of the application at that time reflects that the code has been running from reset and has not stopped on any breakpoints. If the application has breakpoint instructions physically in the code, such as for system calls that run through the debugger, then the code will have entered the fault handlers.

Workaround(s): Because some ARM debuggers expect to maintain connectivity when a system reset is requested, the SYSRESREQ bit in the **Application Interrupt and Reset Control (APINT)** register should not be used when using these debuggers; instead the VECTRESET bit, which only resets the core, should be used so that debug connectivity is uninterrupted. VECTRESET does not reset on-chip peripherals, which must be reset with specific reset operations.

When debugging code that requires a software reset, the SYSRESREQ software reset mechanism in the NVIC (which is used by the Stellaris Peripheral Driver Library SysctlReset() and ROM_SysCtlReset() APIs) should not be used; instead, use the sequence of register writes with a VECTRESET in the NVIC as shown in the code below.

In addition, the ROM is mapped into address 0x0 during reset. The ROM code determines if boot loading is needed, and if not, transfers control to the normal application in Flash memory. As a result, the ROM is visible to the debugger on the reset entry. Debugging can be affected during Flash memory verification because the debugger compares the expected image with the ROM contents and not the Flash memory as intended. The disassembly shown to the user is also affected. To avoid these issues, debuggers must switch off the ROM mapping. However, if the debugger in use does not switch off the ROM, the user can either step through the first assembly instructions until the ROM gets remapped or write a 1 to the BA bit in the **ROM Control (ROMCTL)** register at location 0x400F.E0F0 using the debugger GUI, debugger command line, or debugger startup script.

Use of any reset source listed above other than software reset causes the debugger to lose connectivity.

A consequence to using VECTRESET and a debugger simultaneously, as described in the workaround, is that the USB may fail to enumerate when using a debugger. When debugging a board with USB and a crystal with a frequency greater than 8 MHz, the debugger writes the **RCC** register with the XTAL value for 8 MHz. As a result, the USB PLL cannot lock onto the required frequency and requires a hard reset.

When debugging with USB and a crystal greater than 8 MHz, always disable the USB PLL before writing the **RCC** XTAL value to ensure that the USB PLL starts up correctly.

```
// Disable processor interrupts.
//
IntMasterDisable();
//
// Disable the PLL and the system clock divider (this is a NOP if they are
// already disabled).
//
HWREG(SYSCTL_RCC) = ((HWREG(SYSCTL_RCC) & ~(SYSCTL_RCC_USESYSDIV)) |
                     SYSCTL_RCC_BYPASS);
HWREG(SYSCTL_RCC2) |= SYSCTL_RCC2_BYPASS2;
//
// Now, write RCC and RCC2 to their reset values.
//
HWREG(SYSCTL_RCC) = 0x078e3ad0 | (HWREG(SYSCTL_RCC) & SYSCTL_RCC_MOSCDIS);
HWREG(SYSCTL_RCC2) = 0x07806810;
HWREG(SYSCTL_RCC) = 0x078e3ad1;
//
// Reset the deep sleep clock configuration register.
//
HWREG(SYSCTL_DSLPCLKCFG) = 0x07800000;
//
// Reset the clock gating registers.
//
HWREG(SYSCTL_RCGC0) = 0x00000040;
HWREG(SYSCTL_RCGC1) = 0;
HWREG(SYSCTL_RCGC2) = 0;
HWREG(SYSCTL_SCGC0) = 0x00000040;
HWREG(SYSCTL_SCGC1) = 0;
HWREG(SYSCTL_SCGC2) = 0;
HWREG(SYSCTL_DCGC0) = 0x00000040;
HWREG(SYSCTL_DCGC1) = 0;
HWREG(SYSCTL_DCGC2) = 0;
//
// Reset the remaining SysCtl registers.
//
HWREG(SYSCTL_PBORCTL) = 0;
HWREG(SYSCTL_IMC) = 0;
HWREG(SYSCTL_GPIOHBCTL) = 0;
HWREG(SYSCTL_MOSCCTL) = 0;
HWREG(SYSCTL_PIOSCCAL) = 0;
HWREG(SYSCTL_I2SMCLKCFG) = 0;
//
// Reset the peripherals.
//
HWREG(SYSCTL_SRCR0) = 0xffffffff;
HWREG(SYSCTL_SRCR1) = 0xffffffff;
HWREG(SYSCTL_SRCR2) = 0xffffffff;
HWREG(SYSCTL_SRCR0) = 0;
HWREG(SYSCTL_SRCR1) = 0;
HWREG(SYSCTL_SRCR2) = 0;
//
// Clear any pending SysCtl interrupts.
//
HWREG(SYSCTL_MISC) = 0xffffffff;
//
// Wait for any pending flash operations to complete.
//
while((HWREG(FLASH_FMC) & 0xffff) != 0)
{
}
while((HWREG(FLASH_FMC2) & 0xffff) != 0)
{
}
// Reset the flash controller registers.
//
```

```
HWREG(FLASH_FMA) = 0;
HWREG(FLASH_FCIM) = 0;
HWREG(FLASH_FCMISC) = 0xffffffff;
HWREG(FLASH_FWBVAL) = 0;
//
// Issue the core reset.
//
HWREG(NVIC_APINT) = NVIC_APINT_VECTKEY | NVIC_APINT_VECT_RESET;
```

LM3SYSCTL#17 *JTAG state machine may advance after certain resets*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: Due to an issue with the reset logic, the JTAG state machine may advance to a random state if one of the following resets occurs:

- Hardware Reset via the $\overline{\text{RST}}$ pin
- Brown-Out Reset
- Software System Request Reset (using SYSRESREQ)
- Watchdog Reset
- MOSC Failure Reset

On most devices, these state transitions do not cause any noticeable problems. Some devices, however, eventually execute random JTAG instructions after multiple resets of the type listed above. Since some JTAG instructions can interfere with device operation, steps must be taken to avoid this behavior in a production environment.

In a development environment where a JTAG debugger is being used, this issue is likely to go unnoticed. Many JTAG debuggers pull TCK Low after establishing a connection with the device, which prevents the random state transitions from happening. Also, many JTAG debuggers reset the microcontroller using VECTRESET instead of SYSRESREQ, which also avoids the problem.

Workaround(s): There are two workarounds, each with trade-offs:

- Connect TCK to GND through a 10-K resistor in the final board design. This avoids the problem completely but it causes the chip to draw a small amount of additional current, since the default pin configuration of TCK includes a weak internal pull-up resistor.
- Implement a software routine to explicitly reset the JTAG state machine after every system reset. This works for most cases, but it does not protect the ROM boot loader from erroneous JTAG instruction execution.

```
void
ResetJTAGState(void)
{
    volatile unsigned char ucToggleCount, ucDelayCount;
    //
    // Enable GPIO port C
    //
    HWREG(SYSCTL_RCGC2) = SYSCTL_RCGC2_GPIOC;
    //
    // Dummy read to make sure GPIO port C has time to enable before we
    // proceed.
    //
    ucToggleCount = HWREG(SYSCTL_RCGC2);
    //
    // Unlock the GPIOs on port C
    //
    HWREG(GPIO_PORTC_BASE + GPIO_O_LOCK) = GPIO_LOCK_KEY_DD;
    HWREG(GPIO_PORTC_BASE + GPIO_O_CR) = 0x01;
    //
    // "Toggle" TCK at least 5 times while TMS remains high. We're relying on a
    // digitally disabled pin to feed a "zero" into the JTAG module, and we're
```

```
    // also relying on an external pull-up to feed us our "one". To be extra
    // conservative, let's try 10 toggles.
    //
    for(ucToggleCount = 0; ucToggleCount < 10; ucToggleCount++)
    {
        //
        // Turn off the digital enable for PC0
        //
        HWREG(GPIO_PORTC_BASE + GPIO_O_DEN &= ~ (0x01);

        //
        // Delay a little to make sure the signal propagates through the JTAG
        // state machine (make sure these delays do not get optimized out by
        // the compiler).
        //
        for(ucDelayCount = 0; ucDelayCount < 100; ucDelayCount++)
        {
        }

        //
        // Turn on the digital enable for PC0
        //
        HWREG(GPIO_PORTC_BASE + GPIO_O_DEN) |= 0x01;

        //
        // Delay a little
        //
        for(ucDelayCount = 0; ucDelayCount < 100; ucDelayCount++)
        {
        }
    }
}
```

LM3SYSCTL#18 *Non-word-aligned write to SRAM can cause incorrect value to be loaded*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): 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).

LM3SYSCTL#19 *Internal reset supervisors may not prevent incorrect device operation during power transitions*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s):

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 $\overline{RST}$ 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 3.
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, $\overline{RST}$ 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 4 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 RST Low prior to VDD falling below 3.0 V. Figure 5 shows the resulting waveform of the circuit shown in Figure 4.

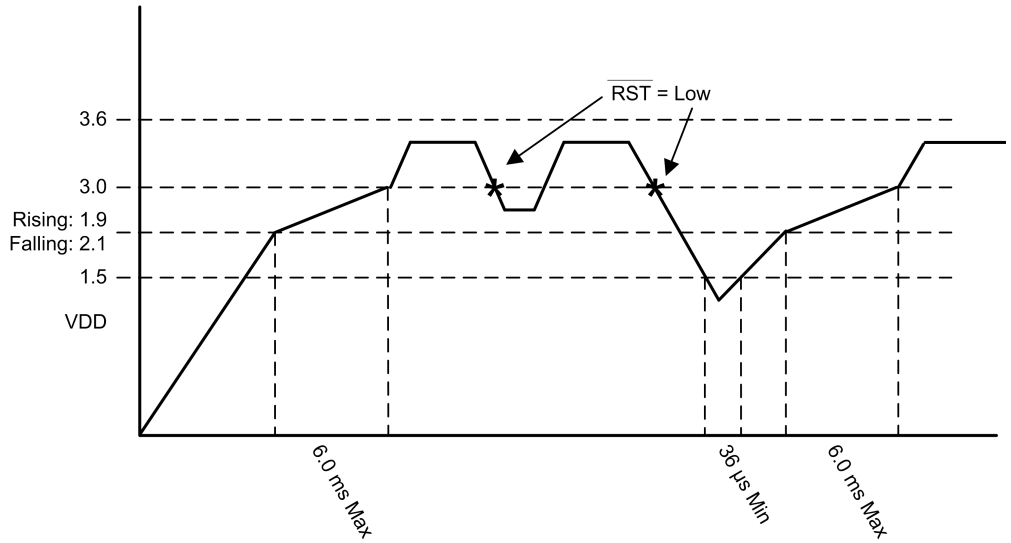


Figure 3. VDD Waveform Signature Limits

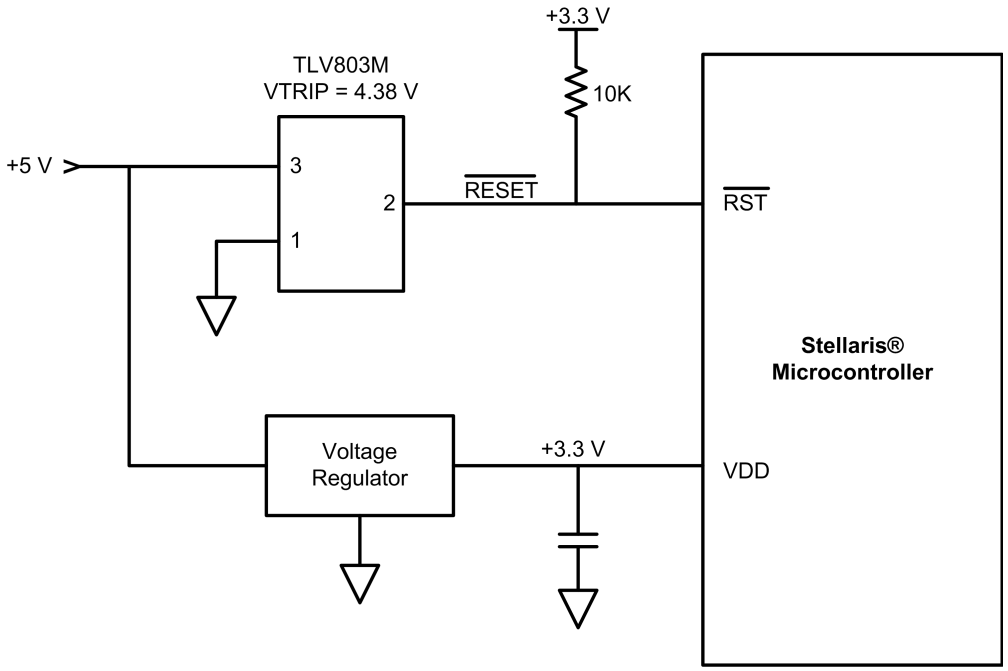


Figure 4. Using a Voltage Supervisor to Monitor the Voltage Rail

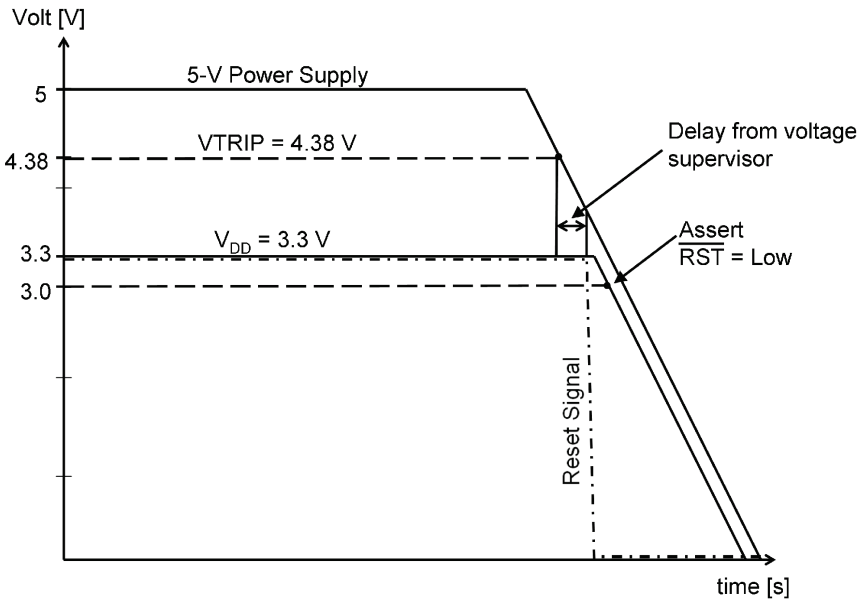


Figure 5. Resulting Waveform Using the Voltage Supervisor Circuit

LM3SYSCTL#20 *The PIOSC cannot be calibrated by the user*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): None.

Fixed on devices with date codes of 17 (July 2011) or later. See Section 2, Device Date Code, for more information.

LM3SYSCTL#21 *ROM boot loaders do not function properly when programming blank flash memory*

Device(s) Affected: Stellaris Firestorm-class Rev A2

Description: All ROM boot loaders are not functional for programming blank Flash memory. After reset, the ROM boot loader checks the reset program counter at address 0x0000.0004. If the data at this address is 0xFFFF.FFFF, the boot loader assumes that the Flash memory is blank and goes through an initialization process. The initialization process enables the UART, SSI, I^2C, USB, and Ethernet modules and then checks for a signal present on each interface. During the initialization process, the boot loader writes to an on-chip Ethernet PHY. For devices without an on-chip Ethernet PHY, the ROM boot loader remains in an infinite loop.

Workaround(s): Use a boot loader resident in Flash memory.

LM3SYSCTL#22 *ROM_UpdateEthernet() is not functional for firmware update*

Device(s) Affected: Stellaris Firestorm-class Rev A2

Description: The ROM_UpdateEthernet() function initializes the Ethernet controller and then looks for a signal present on the interface. During the initialization process, the boot loader writes to an on-chip Ethernet PHY. Because this device does not have an on-chip Ethernet PHY, the ROM boot loader remains in an infinite loop.

Workaround(s): Use a boot loader resident in Flash memory or a serial boot loader when performing firmware updates.

LM3SYSCTL#24 *DSDIVORIDE value of 0x1 does not divide Deep Sleep clock by 2*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: A value of 0x1 for the DSDIVORIDE bit field in the **Deep Sleep Clock Configuration (DSLPCLKCFG)** register does not provide divide by two capability for the Deep Sleep clock. The Run-mode clock divider will be used instead. All other DSDIVORIDE values work as expected when entering Deep Sleep.

Workaround(s): Software must program the SYSDIV bit field of the **Run-Mode Clock Configuration (RCC)** register to the desired divider before entering Deep Sleep if Deep Sleep clock divide by 2 was intended for use. Note that when configuring the SYSDIV bit field, this will affect the Run-mode clock divider. Do not configure the clock divider such that the system clock speed is faster than the maximum clock frequency of 80 MHz before entering Deep Sleep.

LM3UART#01 *The RTRIS bit in the UARTRIS register is only set when the interrupt is enabled*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): 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.

LM3UART#02 *LIN mode Sync Break does not have the correct length*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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.

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(s): Adjust the BLEN encoding to correspond to the actual Sync Break required.

LM3UART#03 *When UART LIN or SIR mode is enabled, µDMA burst transfer does not occur*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

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

LM3USB#05 *USB Host controller may not be used to communicate with a low-speed Device when connected through a hub*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): None.

Fixed on devices with date codes of 0x1A or later. In addition, the system clock on the MCU must be at least 30 MHz. See Section 2, Device Date Code, for more information.

LM3USB#08 *USB paired JK jitter compliance test requires automatic waiver*

Device(s) Affected: Stellaris Tempest-class Rev C5

Description: The USB compliance test results in a Paired JK jitter failure.

Workaround(s): This failure comes with an automatic waiver from usb.org so certification can still be granted.

LM3USB#09 *USB low-speed crossover voltage compliance test requires automatic waiver*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: The USB compliance test results in a crossover voltage range failure when talking to low-speed devices.

Workaround(s): This failure comes with an automatic waiver from usb.org so certification can still be granted.

LM3USB#10 *Special considerations for PB1*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: When using PB1 as a GPIO or digital alternate function, special considerations are required due to issue LM3GPIO#05.

Workaround(s): **USB Device Only Workaround:** The USB VBUS signal must be monitored using a GPIO as an input to detect connect and disconnect. Because of this issue, a GPIO other than PB1 should be used, because PB1 is not 5-V tolerant.

USB OTG 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 LM3GPIO#05 still apply.

LM3USB#11 *USB0DM may be driven after reset*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): 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.

LM3USB#12 *MCU may fail USB certification if the EPI module is operating*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

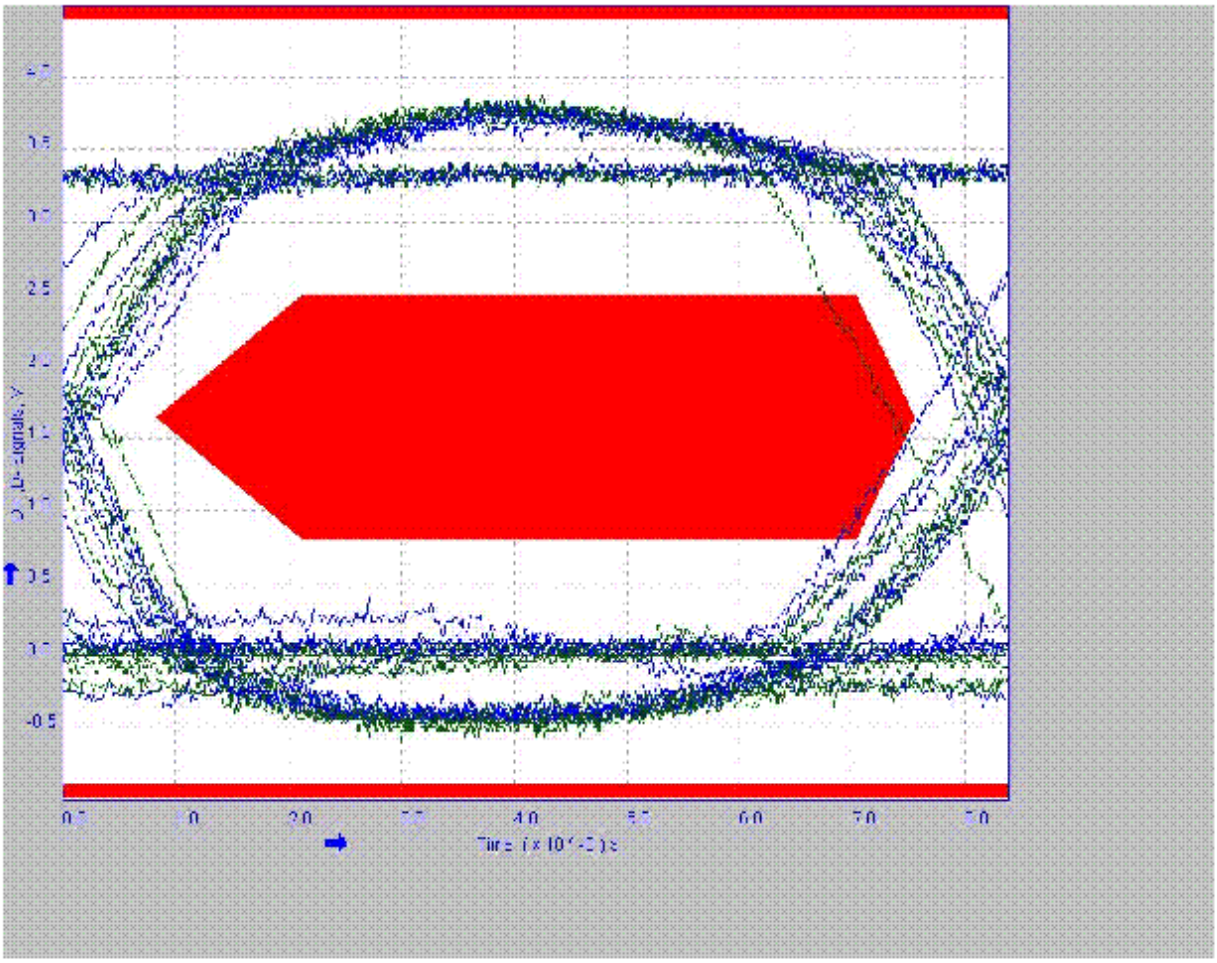
Description: If the EPI module is operating, the USB interface may fail USB certification. Failures have been seen for the eye diagram and jitter (Figure 6). Because of these issues, there is a potential for data corruption on the USB interface when using the USB and EPI at the same time. The risk of 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.

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(s): None.

- 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

Figure 6. Example of USB Certification Failure

LM3USB#15 *USB controller sends EOP at end of device Remote Wake-Up*

Devices(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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 observed. This does not affect USB certification.

Workaround(s): None.

LM3USB#16 *Device sends SE0 in response to a USB bus reset*

Devices(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: The USB Device (Tiva C MCU) will send an Single Ended Zero (SE0) bus state (USB0DP and USB0DM driven low) in response to a USB bus reset from the Host. Per USB specification, the Device should not drive these pins in the event of a USB bus reset. This does not affect USB certification.

Workaround(s): None.

LM3USB#17 *USB Resume occasionally does not wake device from Deep Sleep*

Devices(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: If configured to wake from Deep Sleep mode using a USB Resume signal, the device may remain in Deep Sleep mode if the Host tries to resume the Device from Suspend with a USB bus reset before it can enter Deep Sleep. There is a finite window of time where the RESUME interrupt is not realized. This window is from when the RESUME bit in the **USB Device RESUME Interrupt Status and Clear (USBDRISC)** register is cleared (write a 1) to before the Device enters Deep Sleep. During this time, if a bus reset or wake-up signal is issued by the Host, then the **USBDRISC** status bit clearing causes the valid USB bus operation to be lost.

Workaround(s): To prevent this from occurring, perform one of the two options:

- Ensure that the USB Suspend handler is exited only after a WFI instruction is processed by the core.
- Use Sleep mode instead of Deep Sleep mode and keep the USB module enabled in Sleep mode.

To minimize the window of time when the RESUME interrupt can be lost and reduce the risk of this issue occurring, clear the RESUME bit as close as possible to entering Deep Sleep.

NOTE: If using Sleep mode with the USB module enabled (second workaround), MOSC must be the clock source, using the PLL, and the system clock must be at least 30 MHz. As a result of the higher system clock and using Sleep mode instead of Deep Sleep mode, the current consumption will be higher with this workaround.

LM3WDT#01 *Writes to Watchdog Timer 1 module WDTLOAD register sometimes fail*

Device(s) Affected: Stellaris Tempest-class Rev C5

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(s): 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.

LM3WDT#02 *Watchdog clear mechanism described in the data sheet does not work for the Watchdog Timer 1 module*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): 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.

LM3WDT#03 *Watchdog Timer 1 module asserts reset signal even if not programmed to reset*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): 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.

LM3WDT#04 *WDTLOAD yields an incorrect value when read back*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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(s): None.

LM3WDT#05 *The Watchdog Load (WDTLOAD) register cannot be changed when using a debugger while the STALL bit is set*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

LM3WDT#06 *Reading the WDTVALUE register may return incorrect values when using Watchdog Timer 1*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description Incorrect values may be read from the **Watchdog Value (WDTVALUE)** register at the Watchdog Timer 1 base address when using Watchdog Timer 1.

Workaround(s) None.

LM3WDT#07 *Watchdog timer reloads on any write to the Watchdog Interrupt Clear (WDTICR) register*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

LM3WDT#08 *Watchdog timer will not issue reset if the Watchdog Test (WDTTEST) register is incorrectly programmed*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

Description: Bit 0 of the **Watchdog Test (WDTTEST)** register should never be written to 1, but if rogue software sets this bit, the second watchdog timeout does not cause a reset.

Workaround(s): Ensure that software does not set bit 0 of the **Watchdog Test (WDTTEST)** register.

LM3WDT#09 *The Watchdog Test (WDTTEST) register can be changed even when the registers are locked*

Device(s) Affected: Stellaris Tempest-class Rev C5 and Firestorm-class Rev A2

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

5 Tempest-class Rev C3 and Firestorm-class Rev A1 Known Design Exceptions to Functional Specifications

The following issues affect Stellaris Tempest-Class Rev C3 and Firestorm-Class Rev A1 devices but were fixed on Stellaris Tempest-Class Rev C5 and Firestorm-Class Rev A2 devices.

In addition, all issues in Section 4, "Tempest-class rev C5 and Firestorm-class Rev A2 Known Design Exceptions to Functional Specifications", also affect Rev C3 and Rev A1 devices.

Table 4. Advisory List

LM3ADC#17 *ADC errors exceed specifications*

Device(s) Affected: Stellaris Firestorm-class Rev A1

Description: The error for INL, DNL and Gain for the ADC units exceeds specifications. The error values are shown in Table 5.

Table 5. ADC Error Value

Parameter	12-Bit mode	10-Bit mode
INL	16	4
DNL	8	4
Gain	120	30

Workaround(s): See the application note "Stellaris LM3S ADC Calibration" (SPMA034) for information on how to adjust for error.

LM3MEM#10 *Some ROM functions are incorrect*

Device(s) Affected: Stellaris Tempest-class Rev C3 and Firestorm-class Rev A1

Description: The following ROM functions do not work and should not be used.

- ROM_USBHostMode
- ROM_CANBitRateSet
- ROM_uDMAChannelTransferSet

Workaround(s): Use StellarisWare functions in Flash memory.

LM3MEM#11 *The BOOTCFG register cannot be reliably written until after a special mass erase*

Device(s) Affected: Stellaris Tempest-class Rev C3 and Firestorm-class Rev A1

Description: The **Boot Configuration (BOOTCFG)** register cannot be reliably written until a special type of mass erase is executed.

Workaround(s): Execute the special mass erase sequence described in the “Recovering a 'Locked' Microcontroller” section of the JTAG chapter of your Stellaris microcontroller data sheet prior to a write access to the **BOOTCFG** register. The debug port unlock sequence in LMFlash Programmer can be used to perform this special type of mass erase.

Fixed on devices with date codes of 0B (November 2010) or later. See Section 2, Device Date Code, for more information.

LM3MEM#12 *Flash corruption or device failure may occur at power on*

Device(s) Affected: Stellaris Tempest-class Rev C3 and Firestorm-class Rev A1

Description: There is a small risk of flash corruption or device failure on power up. The issue can occur with certain V_{DD} and V_{DDC} power sequences. The failure is not in the flash memory itself but in the control logic to the flash.

Workaround(s): To eliminate the risk of flash corruption, two power-on requirements must be met:

- The ramp of both V_{DD} and V_{DDC} must begin below 0.2 V.
- V_{DDC} must reach at least 1.0 V before V_{DD} rises above 1.5 V.

Figure 7 details these requirements. Three workaround circuits have been identified that meet these requirements and are described below.

Normally V_{DDC} is supplied by the device's internal voltage regulator from the LDO output pin, however in some circuits the internal regulator may not meet this V_{DDC} timing requirement. A circuit combining an external 1.2 V regulator, a voltage supervisor and a power switch can be used to ensure that this timing requirement is met. The 1.2 V regulator has an integrated Power-OK (POK) circuit that is used to enable V_{DD} when V_{DDC} reaches 1.08 V. During power-down or transient conditions, the POK circuit disables the load switch if V_{DDC} drops below 1.02 V or V_{DD} drops below 1.5 V. The load switch has an internal clamp to accelerate V_{DD} decay.

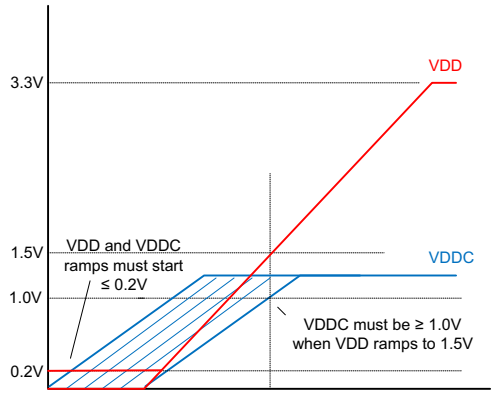


Figure 7. V_{DDC} and V_{DD} Rise Time Relationships

When implementing this workaround, it is important to consider all possible power conditions for the system, including:

- Brown-out (momentary sags in the power source)
- Switch and contact bounce
- Other EMI susceptibility tests
- Various battery and power source disturbances

Three recommended circuits that eliminate the occurrence of this issue are shown below. Although the LDO regulator output is unused in the workaround circuit, a capacitor (1-3 µF) must remain connected for regulator stability. In addition, the LDO pin of the Stellaris device must be disconnected from the external 1.2 V LDO to prevent electronic over-stress of the pin. All of these circuits include two jumpers which must be implemented to bypass the workaround circuit for future silicon revisions.

Figure 8 shows a small chip-scale load switch to control V_{DD}. This circuit is suitable for V_{DD} current up to 2 A peak.

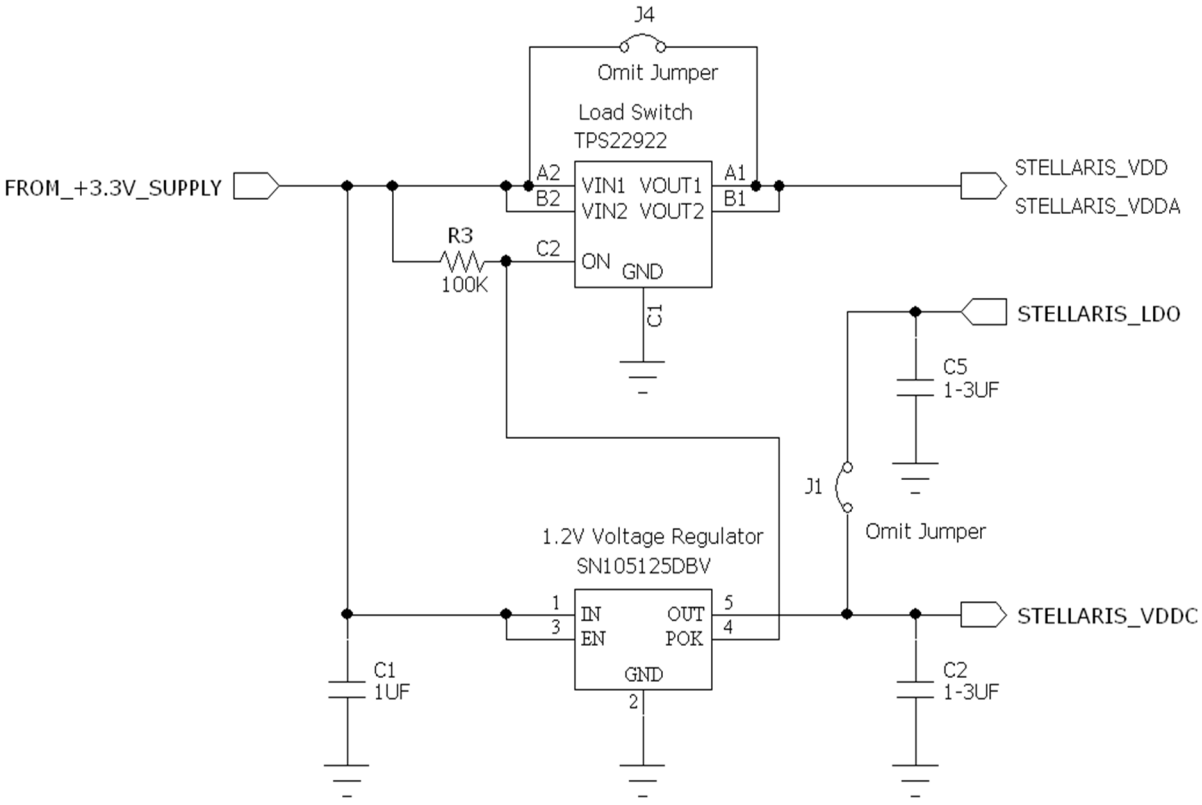


Figure 8. Recommended Voltage Supply Circuit 1

Figure 9 shows a larger SOT-packaged load switch with V_{DD} current capabilities up to 400 mA peak (both channels in parallel).

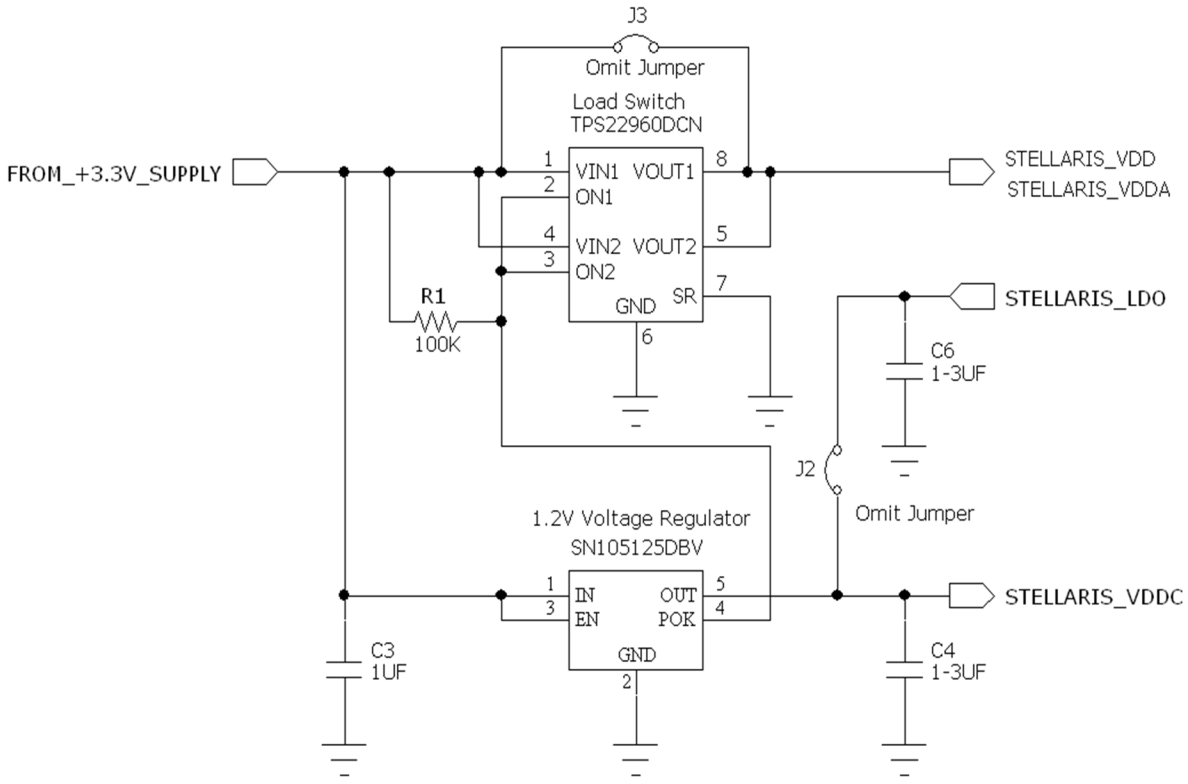


Figure 9. Recommended Voltage Supply Circuit 2

Figure 10 is recommended for designs that require an industrial-temperature operating range. The TPS3808G12 is a dedicated 1.2 V voltage supervisor that ensures V_{DD} is only applied once V_{DDC} is valid. An important consideration is the power dissipation in the 1.2 V LDO voltage regulator. The regulator should remain within its thermal limits while accommodating a worst-case V_{DDC} current of 125 mA. The TPS79912DRV regulator has a θja of 74.2 ºC/W. With a 2.4 V (maximum) voltage drop, the power dissipation is 300 mW. The junction temperature will be approximately 108ºC (23ºC + 85ºC) at 85ºC ambient temperature which is well within the rating of the part. For more information about thermal metrics, see the IC Package Thermal Metrics application report, SPRA953.

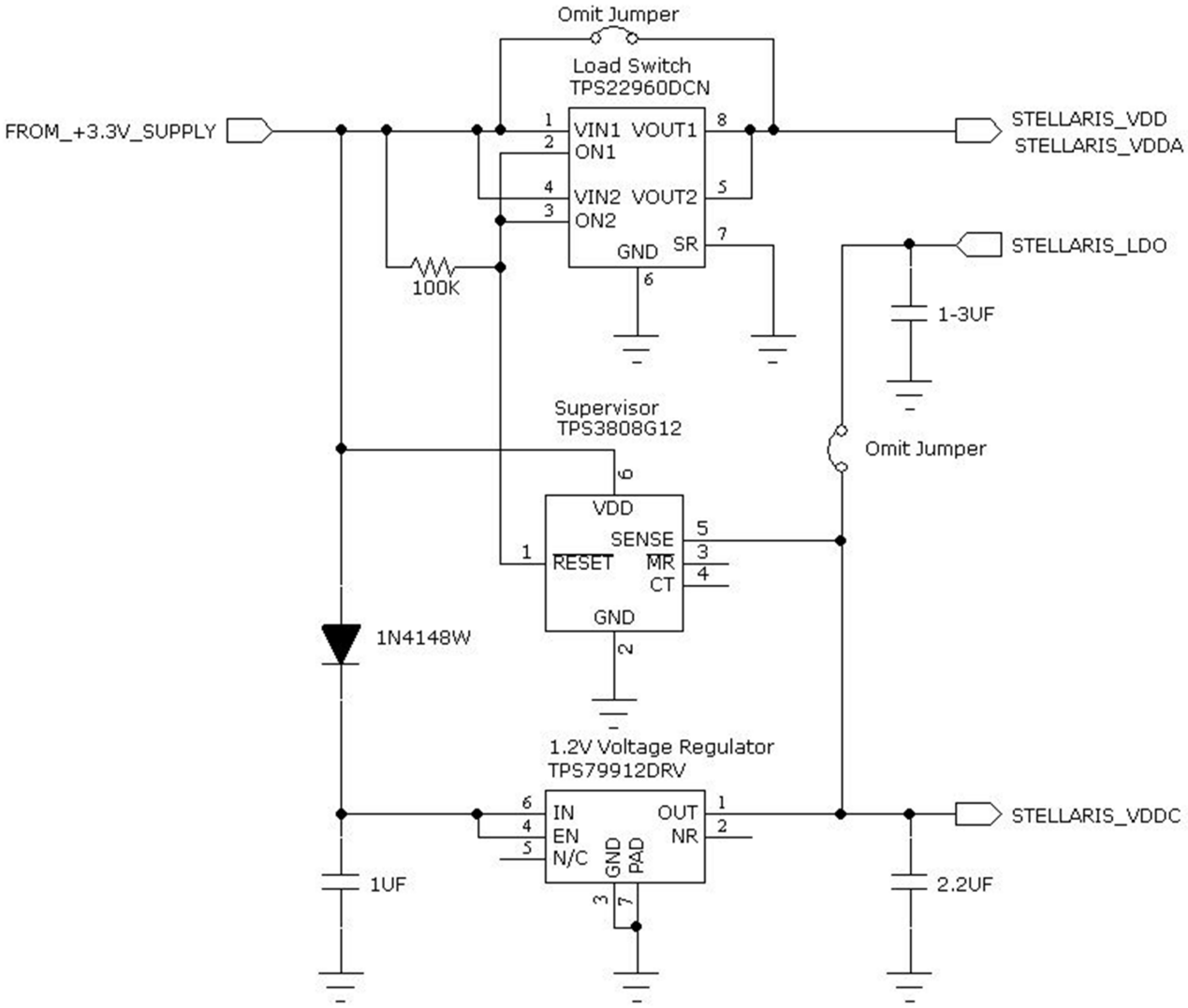


Figure 10. Recommended Voltage Supply Circuit 3

LM3SYSCTL#23 *Brown-out interrupt is never triggered*

Device(s) Affected: Stellaris Tempest-class Rev C3 and Firestorm-class Rev A1

Description: The brown-out circuitry always resets the microcontroller when V_{DD} drops to the brown-out threshold voltage (V_{BTH}), regardless of the state of the BORIOR bit in the **PBORCTL** register.

Workaround(s): None. Fixed on devices with date codes of 17 (July 2011) or later. See Section 2, Device Date Code, for more information.

LM3USB#13 *USB compliance test issue: USB full-speed, far-end signal compliance tests fail with 5 m cable*

Device(s) Affected: Stellaris Tempest-class Rev C3 and Firestorm-class Rev A1

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(s): 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.

LM3USB#14 *USB compliance test issue: USB embedded host low-speed, far-end signal compliance tests fail*

Device(s) Affected: Stellaris Tempest-class Rev C3 and Firestorm-class Rev A1

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 the 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 only full-speed devices are not affected by this erratum. USB device systems are full-speed only and thus are not affected by this erratum.

Workaround(s): None.

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

This silicon errata revision history highlight the technical changes made to this document.

SEE	ADDITIONS/MODIFICATIONS/DELETIONS
Revision * (June 2014) Changes Below:	
Global	• Combined individual LM3Sxxx device errata into one errata document and formatted to TI style. • Created new issue numbers for easier tracking; number will stay the same in each revision going forward.
Section 4 Known Design Exceptions to Functional Specifications	Added the following advisories: • LM3ADC#18: Data may not be present in the FIFO at the time of the sequence interrupt or trigger • LM3ADC#19: The first two ADC samples may be incorrect • LM3GPTM#14: Writes to some General-Purpose Timer registers cause the counter to increment and decrement in some cases • LM3JTAG#01: JTAG INTEST instruction does not work • LM3MEM#13: Flash memory corruption may occur when device is unpowered and stored for several months • LM3MEM#14: Flash-resident ECC logic can cause Flash memory corruption • LM3PWM#09: Under certain circumstances, the PWM load interrupt is triggered as soon as the PWM is enabled • LM3PWM#10: Setting the PWMSYNC bits may not synchronize the PWM counters if PWMDIV is used • LM3QEI#03: When using the index pulse to reset the counter, a specific initial conditions in the QEI module causes the direction for the first count to be misread • LM3SSI#02: SSI Receive FIFO Time-out interrupt may assert sooner than expected in slave • LM3SYSCTL#24: DSDIVORIDE value of 0x1 does not divide Deep Sleep clock by 2 • LM3USB#15: USB controller sends EOP at end of device Remote Wake-Up • LM3USB#16: Device sends SE0 in response to a USB bus reset • LM3USB#17: USB Resume occasionally does not wake device from Deep Sleep • LM3WDT#05: The Watchdog Load (WDTLOAD) register cannot be changed when using a debugger while the STALL bit is set • LM3WDT#06: Reading the WDTVALUE register may return incorrect values when using Watchdog Timer 1 • LM3WDT#07: Watchdog timer reloads on any write to the Watchdog Interrupt Clear (WDTICR) register • LM3WDT#08: Watchdog timer will not issue reset if the Watchdog Test (WDTTEST) register is incorrectly programmed • LM3WDT#09: The Watchdog Test (WDTTEST) register can be changed even when the registers are locked Corrected the following advisories: • LM3EPI#08: RDWS bit field in workaround used to select 6 EPI clocks, not MODE bit field. Removed the advisory LM3UART#04: UART transfers fail at certain system clock frequency and baud rate combinations: • Does not apply to device families, clarification added to the data sheet.

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