MSP430FG4260 Device Erratasheet

1 Functional Errata Revision History

Errata impacting device's operation, function or parametrics.

✓ The check mark indicates that the issue is present in the specified revision.

Errata Number	Rev A
DAC2	✓
DAC3	✓
FLL3	✓
LCDA1	✓
LCDA2	✓
LCDA3	✓
LCDA5	✓
LCDA7	✓
PORT6	✓
SDA3	✓
SDA6	✓
SDA7	✓
TA12	✓
TA16	✓
TA21	✓
TAB22	✓
WDG2	✓

2 Preprogrammed Software Errata Revision History

Errata impacting pre-programmed software into the silicon by Texas Instruments.

✓ The check mark indicates that the issue is present in the specified revision.

The device doesn't have Software in ROM errata.

3 Debug only Errata Revision History

Errata only impacting debug operation.

✓ The check mark indicates that the issue is present in the specified revision.

Errata Number	Rev A
EEM20	✓

4 Fixed by Compiler Errata Revision History

Errata completely resolved by compiler workaround. Refer to specific erratum for IDE and compiler versions with workaround.

✓ The check mark indicates that the issue is present in the specified revision.

Errata Number	Rev A
CPU4	✓

Refer to the following MSP430 compiler documentation for more details about the CPU bugs workarounds.

TI MSP430 Compiler Tools (Code Composer Studio IDE)

- MSP430 Optimizing C/C++ Compiler: Check the --silicon_errata option
- MSP430 Assembly Language Tools

MSP430 GNU Compiler (MSP430-GCC)

- MSP430 GCC Options: Check -msilicon-errata= and -msilicon-errata-warn= options
- MSP430 GCC User's Guide

IAR Embedded Workbench

- IAR workarounds for msp430 hardware issues

5 Package Markings

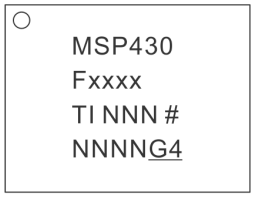
DL48

SOP (DL), 48 Pin

\# = Die revision
○ = Pin 1 location
N = Lot trace code

RGZ48

QFN (RGZ), 48 Pin



\# = Die revision
○ = Pin 1 location
N = Lot trace code

\# = Die revision
○ = Pin 1 location
N = Lot trace code

\# = Die revision
○ = Pin 1 location
N = Lot trace code

NOTE: Package marking with "TM" applies only to devices released after 2011.

6 Detailed Bug Description

CPU4 *CPU Module*

Category Compiler-Fixed

Function PUSH #4, PUSH #8CPU4 - Bug

Description The single operand instruction PUSH cannot use the internal constants (CG) 4 and 8. The other internal constants (0, 1, 2, -1) can be used. The number of clock cycles is different:

PUSH #CG uses address mode 00, requiring 3 cycles, 1 word instruction

PUSH #4/#8 uses address mode 11, requiring 5 cycles, 2 word instruction

Workaround Refer to the table below for compiler-specific fix implementation information.

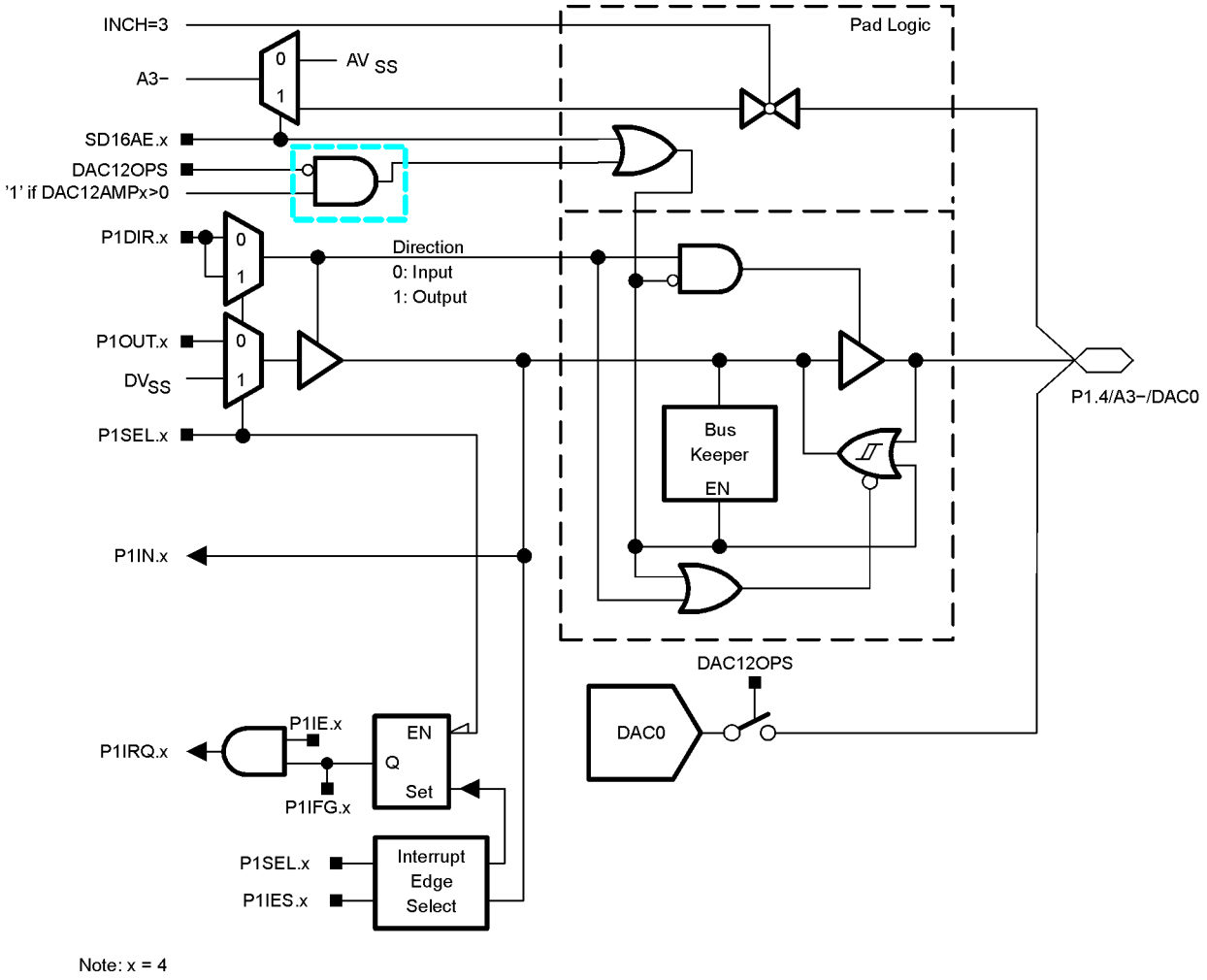
IDE/Compiler	Version Number	Notes
IAR Embedded Workbench	IAR EW430 v2.x until v6.20	User is required to add the compiler flag option below. --hw_workaround=CPU4
IAR Embedded Workbench	IAR EW430 v6.20 or later	Workaround is automatically enabled
TI MSP430 Compiler Tools (Code Composer Studio)	v1.1 or later	
MSP430 GNU Compiler (MSP430-GCC)	MSP430-GCC 4.9 build 167 or later	

DAC2 *DAC12 Module*

Category Functional

Function P1.4 GPIO function is not disabled when P1.4 = DAC0 output

Description The DAC12OPS control bit used to automatically disable the P1.4 I/O logic is inverted as shown in the figure below. When DAC12 is enabled (DAC12AMPx > 0) and DAC12OPS=0, the port I/O for P1.4 will be disabled. Setting DAC12OPS = 1 to connect the DAC12 output to P1.4 will erroneously enable the port GPIO logic.



Workaround The P1.4 I/O logic should be disabled when the DAC12 is enabled using the SD16AE.4 analog enable bit. SD16AE.4=1 will disable the P1.4 I/O logic. The A3 SD16_A analog input cannot be used under this condition.

DAC3 *DAC12 Module*

Category Functional

Function Port P1.4 can not be used if DAC12 is internally enabled.

Description When DAC12 is enabled (DAC12AMPx > 0) and internal use is selected (DAC12OPS=0), the P1.4 digital I/O functionality is disabled.

See also: DAC2 bug description.

Workaround None

EEM20 *EEM Module*

Category Debug

Function	Debugger might clear interrupt flags
Description	During debugging read-sensitive interrupt flags might be cleared as soon as the debugger stops. This is valid in both single-stepping and free run modes.
Workaround	None.

FLL3 *FLL+ Module*

Category	Functional
Function	FLLDx = 11 for /8 may generate an unstable MCLK frequency
Description	When setting the FLL to higher frequencies using FLLDx = 11 (/8) the output frequency of the FLL may have a larger frequency variation (e.g. averaged over 2sec) as well as a lower average output frequency than expected when compared to the other FLLDx bit settings.
Workaround	None

LCDA1 *LCD_A Module*

Category	Functional
Function	High voltage on LCD_A pins.
Description	When static LCD mode is selected, the charge pump does not work correctly. The feedback loop within the charge pump is switched off. This can result in high voltages on LCD_A segment and common pins. WARNING: Using the charge pump when static mode is selected may cause permanent damage to the MSP430 device and/or the LCD.
Workaround	None

LCDA2 *LCD_A Module*

Category	Functional
Function	Floating segment S5
Description	Dedicated LCD segment S5 is floating when LCD_A is disabled: LCDON = 0. In this case S5 should be connected to ground.
Workaround	None

LCDA3 *LCD_A Module*

Category	Functional
Function	Charge pump voltage
Description	The charge pump output voltage has an offset of approximately -200 mV. This reduces the LCD voltage levels specified in the datasheet for LCD_A by the same amount and should be accounted for when selecting a charge pump voltage. See actual values

below:

LCD_A

PARAMETER		TEST CONDITIONS	VCC	MIN	TYP	MAX	UNIT
$V_{CC(LCD)}$	Supply voltage	Charge pump enabled (LCDCPEN = 1; VLCDx > 0000)		2.2		3.6	V
C_{LCD}	Capacitor on LCDCAP (see Note 1)	Charge pump enabled (LCDCPEN = 1; VLCDx > 0000)		4.7			µF
$I_{CC(LCD)}$	Average supply current (see Note 2)	$V_{LCD(typ)}$=3V; LCDCPEN = 1; VLCDx= 1000, all segments on f_{LCD}= f_{ACLK}/32 no LCD connected (see Note 2) T_A = 25°C	2.2 V		3.8		µA
f_{LCD}	LCD frequency					1.1	kHz
V_{LCD}	LCD voltage	VLCDx = 0000			VCC		V
		VLCDx = 0001			2.50		
		VLCDx = 0010			2.56		
		VLCDx = 0011			2.61		
		VLCDx = 0100			2.67		
		VLCDx = 0101			2.72		
		VLCDx = 0110			2.78		
		VLCDx = 0111			2.83		
		VLCDx = 1000			2.89		
		VLCDx = 1001			2.94		
		VLCDx = 1010			3.00		
		VLCDx = 1011			3.05		
		VLCDx = 1100			3.11		
		VLCDx = 1101			3.16		
		VLCDx = 1110			3.22		
		VLCDx = 1111		3.12	3.27	3.42	
R_{LCD}	LCD driver output impedance	V_{LCD} = 3V; LCDCPEN = 1; VLCDx = 1000, I_{LOAD} = ±10µA	2.2 V			10	kΩ

NOTES: 1. Enabling the internal charge pump with an external capacitor smaller than the minimum specified might damage the device.
2. Connecting an actual display will increase the current consumption depending on the size of the LCD.

Workaround None

LCDA5 *LCD_A Module*

Category Functional

Function Wrong cycle time for first cycle of COMx/Sx signals

Description The time of the first cycle of COMx/Sx signals after enabling the LCD_A module is only half of the selected value. All following cycles are correct

Workaround Not required, because it does not influence the LCD function.

LCDA7 *LCD_A Module*

Category Functional

Function Higher current consumption when using shared LCD ports as fast toggling outputs

Description If a shared LCD pin (segment or com line) is used as digital fast toggling output (f>10kHz) and the VLCD is >0V (BG enabled) the device current consumption increases with higher toggling frequencies.

Workaround 1. Do not use shared LCD pins as fast toggling outputs if an LCD is used.

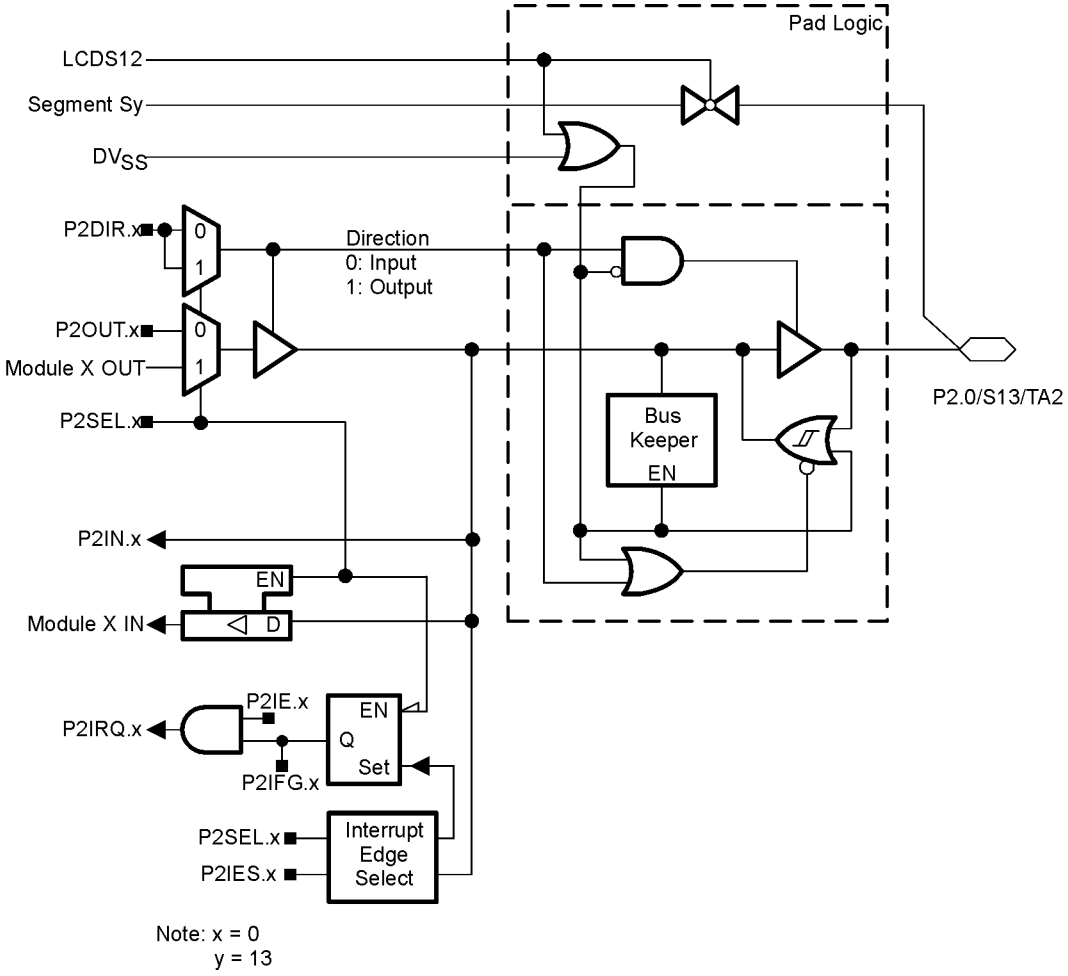
2. Reduce the toggle frequency of the shared pin to <10kHz.

PORT6 *PORT Module*

Category Functional

Function P2.0 module function

Description In addition to GPIO and LCD functionality, P2.0 also has Timer_A3 module output capability. When P2SEL.0 = 1 and P2DIR.0 = 1, P2.0 becomes a TA2 output.



PIN NAME (P2.X)	X	FUNCTION	CONTROL BITS / SIGNALS		
			P2DIR.x	P2SEL.x	LCDS12
P2.0/S13/TA2	0	P2.0 Input/Output	0/1	0	0
		N/A	0	1	0
		Timer_A3.TA2	1	1	0
		S13	X	X	1

Workaround N/A

SDA3 *SD16_A Module*

Category Functional

Function The interrupt delay function can result in incorrect conversion data

Description The interrupt delay operation can result in incorrect conversion data when SD16INTDLYx = 01, 10 or 11.

Workaround Use SD16INTDLYx = 00 setting (interrupt generated after fourth conversion). This applies to the first conversion in Continuous mode and to each conversion in Single mode.

SDA6 *SD16_A Module*

Category Functional

Function SD16CCTL0 write leads to unexpected results from SD16_A.

Description 1) No write attempt should be made to the register address 0x00B8 since it might lead to malfunctioning of SD16A.

2) Bit-0 and bit-15 of register SD16CCTL0 should not be set as it might lead to unexpected results from the SD16A.

Workaround None

SDA7 *SD16_A Module*

Category Functional

Function Reduced performance when input buffer is enabled.

Description The SD16_A performance is degraded if the high-impedance input buffer is enabled (SD16CCTLx.SD16BUFx > 0) and the analog input voltage is in the range of AVSS to AVSS+0.2V. Avoid analog input voltages in this range when the high-impedance input buffer is enabled.

Workaround None

TA12 *TIMER_A Module*

Category Functional

Function Interrupt is lost (slow ACLK)

Description Timer_A counter is running with slow clock (external TACLK or ACLK)compared to MCLK. The compare mode is selected for the capture/compare channel and the CCRx register is incremented by one with the occurring compare interrupt (if TAR = CCRx). Due to the fast MCLK the CCRx register increment (CCRx = CCRx+1) happens before the Timer_A counter has incremented again. Therefore the next compare interrupt should happen at once with the next Timer_A counter increment (if TAR = CCRx + 1). This interrupt gets lost.

Workaround Switch capture/compare mode to capture mode before the CCRx register increment. Switch back to compare mode afterwards.

TA16 *TIMER_A Module*

Category Functional

Function First increment of TAR erroneous when IDx > 00

Description The first increment of TAR after any timer clear event (POR/TACLR) happens immediately following the first positive edge of the selected clock source (INCLK, SMCLK, ACLK or TACLK). This is independent of the clock input divider settings (ID0, ID1). All following TAR increments are performed correctly with the selected IDx settings.

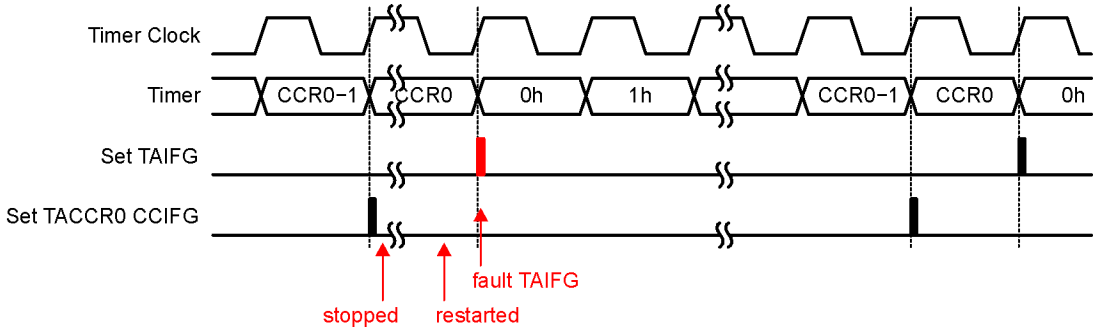
Workaround None

TA21 *TIMER_A Module*

Category Functional

Function TAIFG Flag is erroneously set after Timer A restarts in Up Mode

Description In Up Mode, the TAIFG flag should only be set when the timer counts from TACCR0 to zero. However, if the Timer A is stopped at TAR = TACCR0, then cleared (TAR=0) by setting the TACLR bit, and finally restarted in Up Mode, the next rising edge of the TACLK will erroneously set the TAIFG flag.



Workaround None.

TAB22 *TIMER_A/TIMER_B Module*

Category Functional

Function	Timer_A/Timer_B register modification after Watchdog Timer PUC
Description	Unwanted modification of the Timer_A/Timer_B registers TACTL/TBCTL and TAIV/TBIV can occur when a PUC is generated by the Watchdog Timer(WDT) in Watchdog mode and any Timer_A/Timer_B counter register TACCRx/TBCCRx is incremented/decremented (Timer_A/Timer_B does not need to be running).
Workaround	Initialize TACTL/TBCTL register after the reset occurs using a MOV instruction (BIS/BIC may not fully initialize the register). TAIV/TBIV is automatically cleared following this initialization. Example code: MOV.W #VAL, &TACTL or MOV.W #VAL, &TBCTL Where, VAL=0, if Timer is not used in application otherwise, user defined per desired function.

WDG2 *WDT Module*

Category	Functional
Function	Incorrectly accessing a flash control register
Description	If a key violation is caused by incorrectly accessing a flash control register, the watchdog interrupt flag is set in addition to the expected PUC.
Workaround	None

7 Document Revision History

Changes from family erratasheet to device specific erratasheet.

1. Errata SDA1 was removed
2. Errata SDA6 was added
3. Errata TAB22 was added
4. Errata SDA7 was added
5. Errata LCDA7 was added
6. Errata LCDA5 was added
7. RGZ48 package markings have been updated

Changes from device specific erratasheet to document Revision A.

1. Errata EEM20 was added to the errata documentation.

Changes from document Revision A to Revision B.

1. Errata TA21 was added to the errata documentation.

Changes from document Revision B to Revision C.

1. Package Markings section was updated.

Changes from document Revision C to Revision D.

1. TA21 Description was updated.

Changes from document Revision D to Revision E.

1. Function for CPU4 was updated.
2. Workaround for CPU4 was updated.

Changes from document Revision E to Revision F.

1. Erratasheet format update.
2. Added errata category field to "Detailed bug description" section

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

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