

TMS320F243 and TMS320F241

DSP Controllers

Silicon Errata

Silicon Revision 2.1

SPRZ150G
September 1999
Revised February 2002



Copyright © 2002, Texas Instruments Incorporated

Contents

1	Introduction	4
1.1	Quality and Reliability Conditions	4
	TMX Definition	4
	TMP Definition	4
	TMS Definition	4
1.2	Revision Identification	5
2	Known Design Marginality/Exceptions to Functional Specifications	6
	Analog-to-Digital Converter (ADC)	6
	PMT Pin	6
	WDFLAG Bit (in WDCR Register)	6
	SCI TX_EMPTY	7
	Write-Write-Read (WWR) of Peripheral Registers (Applicable to F243 Only)	7
	QEP Circuit	8
	ILLADR Bit	8
	PDPINT Pin Operation	9
	NMI Pin	9
	CAPn Input	9
	SPISTE Timing Anomaly	10
	Phantom Interrupts While Using the Real-Time Monitor	10
3	24x CAN Module Exceptions to CAN 2.0B Specifications	11
	Second Stuff Bit is Sampled Dominant	11
	24x CAN Generates an Error Frame up to 1.5 TQ Too Early	11
	No Frame Acceptance After Passive Error Frame Transmission	12
	24x CAN Does Not Stop Sending Active Error Flags (1)	12
	24x CAN Does Not Stop Sending Active Error Flags (2)	12
	24x CAN Does Not Acknowledge a Frame	13
	24x CAN Does Not Send a Passive Error Frame	13
	Back-to-Back Data Transmit/Receive Within the Same CAN Module May Cause Sporadic Loss of Data Integrity .	13
	Method to Service Multiple Interrupts Through the Existing Two-Interrupt Structure	14
	AME and AAM Bits Get Corrupted Upon Reception	15
	CAN Bit Timing	15
4	Programming Considerations for the 24x ADC	16
4.1	Checking for End-of-Sequence	16
5	Errata for TMS320F243, TMS320F241 DSP Controllers Data Sheet (SPRS064C)	17
5.1	10-Bit Dual Analog-to-Digital Converter (ADC) Section, Recommended Operating Conditions Table (p. 103)	17
5.2	Figure 16, TMS320x24x DSP Device Nomenclature, (p.73)	17

6	Errata for TMS320F24x DSP Controllers Reference Guide: Systems and Peripherals (SPRU276C)	17
6.1	Section 3.3, Illegal Address Detect	17
6.2	Section 11.4.4, Master Control Register (MCR)	17
6.3	Section 8.4.1, ADCTRL1 Register	17
7	Documentation Support	18

1 Introduction

This document describes the silicon updates to the functional specifications for the TMS320F243 and the TMS320F241 DSP Controllers, silicon revision 2.1. The updates are applicable to:

- TMS320F243 (144-pin LQFP, PGE suffix)
- TMS320F241 (68-pin PLCC, FN suffix; 64-pin PQFP, PG suffix)

1.1 Quality and Reliability Conditions

TMX Definition

Texas Instruments (TI) does not warranty either (1) electrical performance to specification, or (2) product reliability for products classified as “TMX.” By definition, the product has not completed data sheet verification or reliability performance qualification according to TI Quality Systems Specifications.

The mere fact that a “TMX” device was tested over a particular temperature range and voltage range should not, in any way, be construed as a warranty of performance.

TMP Definition

TI does not warranty product reliability for products classified as “TMP.” By definition, the product has not completed reliability performance qualification according to TI Quality Systems Specifications; however, products are tested to a published electrical and mechanical specification.

TMS Definition

Fully-qualified production device.

1.2 Revision Identification

The device revision can be determined by the lot trace code marked on the top of the package. The location of the lot trace code for the PGE package is shown in Figure 1.

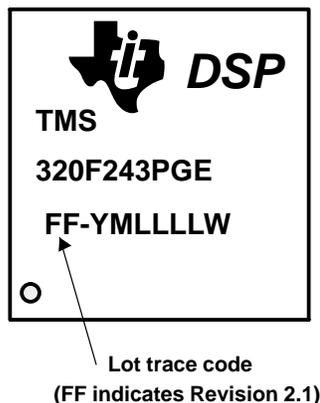


Figure 1. Example, Lot Trace Code for TMS320F243 (PGE)

Prefix of Lot Trace Code	Silicon Revision	Comments
FF	Indicates Revision 2.1	This silicon revision is available as TMS.

2 Known Design Marginality/Exceptions to Functional Specifications

Advisory

Analog-to-Digital Converter (ADC)

Revision(s) Affected: 2.1

Details: When the IM bit (bit 14 in the ADCTRL2 register) = 1, the second conversion must be initiated before the first conversion is complete; otherwise, the ADC module may hang up and fail to do further conversions.

Workaround: Initiate the second ADC conversion before the first conversion is complete, or set this bit to zero.

CAUTION:

When the IM bit = 1, the ADCEOC bit is cleared **only** at the end of two conversions, i.e., ADCEOC is not cleared after a single conversion.

Revision(s) Affected: 2.1

Details: When the pseudo-ADC is enabled for the first time after a “power-on reset” (POR), a conversion is triggered (i.e., a conversion takes place even when no conversion has been initiated). Warm resets do not have this problem; only PORs have this problem.

Workaround: Perform a dummy conversion as part of the ADC initialization and flush the ADCFIFOs (after ensuring an EOC). After this, the ADC may be used as desired.

Advisory

PMT Pin

Revision(s) Affected: 2.1

Details: When the PMT pin is left floating (as described in the *TMS320F243, TMS320F241 DSP Controllers* data sheet, literature number *SPRS064*), glitches are observed in both the GPIO pins and the memory-select pins in noisy environments.

Workaround: For proper operation of the 24x DSP, the PMT pin should be connected to ground.

Advisory

WDFLAG Bit (in WDCR Register)

Revision(s) Affected: 2.1

Details: The power-on reset (POR) state of the WDFLAG bit is undefined. This could cause confusion if the user code attempts to differentiate a watchdog-initiated reset from a power-on reset.

Workaround: If an application implements a mechanism to differentiate a POR from other types of reset, the WDFLAG bit must be cleared after a POR. Once this is done, the WDFLAG bit can be set only by a watchdog reset.

Advisory

SCI TX_EMPTY

Revision(s) Affected: 2.1

Details: The TX_EMPTY bit (bit 6 in the SCICTL2 register) is supposed to get set when the transmit buffer and the shift register are **both** empty. However, this bit may occasionally be read as being set even when there is data in the transmit buffer.

Assuming two characters are being written back-to-back to the SCI, the TX_EMPTY bit is expected to go high after the second character is shifted out of the SCITXD pin. Occasionally, the TX_EMPTY bit goes high right after the first character is shifted out (i.e., when the second character is still in the transmit buffer). If the TX_EMPTY bit is polled to check the end-of-transmission of all characters, it may provide incorrect information about the status of the transmission.

Workaround: Follow these steps:

1. Read the TX_EMPTY bit.
2. If set, execute 2 NOPs and proceed to Step 3. Else, repeat Step 1.
3. Read the TX_EMPTY bit again.
4. If the bit is still set, then there are no more characters to be transmitted. If the bit is not set, it indicates transmission is not complete. Proceed to Step 1.

Advisory*Write-Write-Read (WWR) of Peripheral Registers (Applicable to F243 Only)***Revision(s) Affected:** 2.1

Details: This problem occurs only when code is run from external program memory at non-zero wait-states (for program memory). When two writes are performed to successive peripheral registers and immediately followed by a read of either one of the registers, a read of the first write returns an incorrect value. Writes and reads from/to external, DARAM memory are not affected by this problem.

Workaround: Insert a NOP between the second write instruction and read or use zero wait-state for the program memory.

Advisory

QEP Circuit

Revision(s) Affected: 2.1**Details:**

After a DSP reset, the QEP module fails to detect the first transition that occurs on QEP input pins. Therefore, if the first transition occurs after a GP Timer has been initialized and enabled as the QEP counter (i.e., to use QEP as source of clock), the first transition will not be counted by the GP Timer. The result is an error of one count in the GP Timer out of a total of 1024 counts for a 256-line encoder, or 4096 counts for a 1024-line encoder. However, the issue is not a concern under any of the following conditions:

1. *The first transition happens **before** the GP Timer is initialized and enabled as QEP counter.* This ensures that all transitions are counted after initialization.
2. *After the first index pulse is received and if the index pulse is used to recalibrate the GP Timer (through capture interrupt).* The recalibration corrects the error in the GP Timer; therefore, from the time the first index pulse is received, the QEP counter becomes accurate.

Workaround(s):

1. Make the first transition happen before the GP timer is initialized and enabled as QEP counter. This is usually the case because typically the rotor shaft is locked to a known position before the GP Timer is initialized. Locking the rotor shaft will generate transitions on QEP input pins, unless the rotor shaft is exactly aligned to the known position (which is a rare case). Disturbing the rotor shaft on purpose takes care of the rare case.
2. Use the index pulse of the encoder to recalibrate the GP Timer used as QEP counter.

Advisory

ILLADR Bit

Revision(s) Affected: 2.1**Details:**

The ILLADR bit in the SCSR1 register may be set even though no illegal address access took place.

Workaround:

This is a problem with the debugger-hardware interaction. This bit may be set on an emulation suspend such as a breakpoint or while single stepping. However, this bit works correctly in run time (i.e., this bit works correctly when the program is free-run without the JTAG connector being connected). This can be easily verified by writing a test code (running out of Flash) that copies the status of this bit to the XF bit and then monitoring the status of the XF pin using an oscilloscope.

Advisory*PDPINT* Pin Operation

Revision(s) Affected: 2.1

Details: If the $\overline{\text{PDPINT}}$ pin is low during a device reset, it prevents the recognition of valid PDPINT interrupts in the future.

Workaround: To overcome this problem, bit PIRQR0.0 (for PDPINTA) and bit PIRQR2.0 (for PDPINTB) need to be cleared to zero. This can be achieved by writing a zero to these bits or by writing a one to the corresponding PIACKRn.0 bits. In addition to this, the EVAIFRA.0 bit (or EVBIFRA.0 bit) should be cleared as appropriate.

The PIRQRn and PIACKRn registers are *not* intended to be used in user applications. The workaround mentioned herein is a special case during the initialization of the device. After initialization, these registers should not be used by the user code.

Advisory*NMI* Pin

Revision(s) Affected: 2.1

Details: The $\overline{\text{NMI}}$ pin is level-sensitive in the F243/F241 devices. If this pin is held low for longer than necessary, multiple interrupts will be asserted. If the $\overline{\text{NMI}}$ pin is sensed low after the code has branched to the NMI vector, the return address pushed onto the stack will be the NMI vector itself. If this phenomenon occurs repetitively, it can fill the stack with the address of the NMI vector (0024h), pushing out the "original" return address of the program. After executing the NMI ISR, the program branches back to the NMI vector and executes the NMI ISR again. This goes on forever and the control never returns to the main program. This fact has to be taken into account while using the $\overline{\text{NMI}}$ pin in target designs.

Workaround: None

Advisory*CAPn* Input

Revision(s) Affected: 2.1

Details: If a CAPn input is high while the capture units are being enabled, it triggers spurious captures. The CAPFIFO status bits would indicate that there are two words in the FIFO, when in reality, no capture was triggered. If the capture interrupt is enabled, it will trigger the capture interrupt as well.

Workaround: Initialize the CAPFIFO status bits (and enable the capture interrupt) after configuring the capture units.

Advisory *$\overline{\text{SPISTE}}$ Timing Anomaly***Revision(s) Affected:** 2.1

Details: The $\overline{\text{SPISTE}}$ signal has a timing anomaly due to which, it goes high sooner than it should (i.e., before the last bit has ended). This results in unreliable read of the last bit of the word by both the master and the slave. Following is a summary of the bug description depending on which SPI clocking scheme is used:

- **CLOCK POLARITY = 0, PHASE = 0**
The $\overline{\text{SPISTE}}$ signal goes high 3 CPU clock cycles after the falling edge of the SPICLK for the last bit of transmission.
- **CLOCK POLARITY = 0, PHASE = 1**
The $\overline{\text{SPISTE}}$ signal goes high 3 CPU clock cycles after the rising edge of the SPICLK for the last bit of transmission.
- **CLOCK POLARITY = 1, PHASE = 0**
The $\overline{\text{SPISTE}}$ signal goes high 3 CPU clock cycles after the rising edge of the SPICLK for the last bit of transmission.
- **CLOCK POLARITY = 1, PHASE = 1**
The $\overline{\text{SPISTE}}$ signal goes high 3 CPU clock cycles after the falling edge of the SPICLK for the last bit of transmission.

Note that the timing issue is aggravated when PHASE = 1, since the time window for which the $\overline{\text{SPISTE}}$ signal is valid for the last bit gets reduced further.

Workaround: The $\overline{\text{SPISTE}}$ pin should be used in its GPIO configuration. This pin can then be manipulated manually in software. Note that this workaround does not require any hardware change.

Advisory*Phantom Interrupts While Using the Real-Time Monitor***Revision(s) Affected:** 2.1

Details: When the real-time monitor (RTM) is used, phantom interrupts may be occasionally seen—i.e., the PIVR register is loaded with 0000h when an interrupt is asserted. This problem is seen **only while debugging with the RTM** and is not applicable in actual applications.

Workaround: Check the PIVR value in the ISR. If a phantom interrupt is detected, return to the main code.

3 24x CAN Module Exceptions to CAN 2.0B Specifications

Advisory*Second Stuff Bit is Sampled Dominant***Revision(s) Affected:** 2.1**Details:** The sample point of the 24x CAN is up to 1 TQ (time quantum) too early. In a worst-case system, the 24x CAN misinterprets received bits and disturbs the bus with error frames.**Workaround:** An indirect solution exists to solve the problem. The system integrator who defines the position of the sample point has to consider the mismatching sample point position. If this is done, a worst-case system is fully functional with following restriction:

Restriction: The maximum allowed oscillator/resonator tolerance and the maximum bus length are restricted compared to a fully functional CAN system.

Note: This problem impacts the CAN bit timing configuration. Using crystal/oscillators instead of resonators will mitigate the bit timing issues. A spreadsheet to calculate the correct sampling point in light of this problem is available from Texas Instruments.

Advisory*24x CAN Generates an Error Frame up to 1.5 TQ Too Early***Revision(s) Affected:** 2.1**Details:** When the 24x CAN detects a stuff error, it transmits the error frame up to 1.5 TQ too early. Similar to the *Second Stuff Bit is Sampled Dominant* advisory above, this fault only has an impact in a worst-case system.**Workaround:** As described in the *Second Stuff Bit is Sampled Dominant* advisory, the system integrator has to consider the fault when specifying the CAN system.

Restriction: As described in the *Second Stuff Bit is Sampled Dominant* advisory, the “too early” error frame reduces the maximum allowed oscillator/resonator tolerance and the maximum bus length.

Note: This problem impacts the CAN bit timing configuration. Using crystal/oscillators instead of resonators will mitigate the bit timing issues.

Advisory*No Frame Acceptance After Passive Error Frame Transmission*

Revision(s) Affected: 2.1

Details: The 24x CAN needs to be in error-passive state. When it transmits a passive error flag and another CAN starts transmitting a data frame during the third bit of intermission, then the 24x CAN does not receive this frame.

Workaround: Due to relaxed oscillator tolerances, it is possible/allowed that a transmission starts in the third bit of intermission. In order to avoid start of transmission in the third bit of intermission, the system integrator has to consider this fault when calculating the maximum allowed oscillator tolerance for the overall CAN system.

Restriction: As already mentioned, the maximum allowed oscillator tolerance and the maximum bus length will be reduced.

Note: This problem impacts the CAN bit timing configuration. Using crystal/oscillators instead of resonators will mitigate the bit timing issues.

Advisory*24x CAN Does Not Stop Sending Active Error Flags (1)*

Revision(s) Affected: 2.1

Details: This problem can lock up the CAN system. The 24x CAN is simulated to send an active error frame while being a receiver, and in parallel, its RX pin is permanently forced to recessive level. The 24x CAN does not stop sending active error frames until the stuck-to-recessive condition is removed. This finding is similar to that described in the *24x CAN Does Not Stop Sending Active Error Flags (2)* advisory, but in this case, the 24x CAN acts as receiver and therefore does not go bus off.

Workaround: Problem can occur when there is a fault on the CAN network. The DSP has to be reset.

Advisory*24x CAN Does Not Stop Sending Active Error Flags (2)*

Revision(s) Affected: 2.1

Details: This problem can lock up the CAN system for a “long” time (until it becomes bus off). When the 24x CAN is forced into error-passive state because of a permanent disturbance on the bus, the 24x CAN keeps on sending an active error frame until it becomes bus off.

Workaround: Problem can occur when there is a fault on the CAN network. The DSP has to be reset.

Advisory*24x CAN Does Not Acknowledge a Frame*

Revision(s) Affected: 2.1

Details: This problem seems to relate to the finding described in the *No Frame Acceptance After Passive Error Frame Transmission* advisory. When the 24x CAN is forced into error-passive state and when another CAN starts transmitting a data frame during the third bit of intermission of its own data frame, then the receive frame is neither acknowledged nor received.

Workaround: Due to relaxed oscillator tolerances, it is possible/allowed that a transmission starts in the third bit of intermission. In order to avoid start of transmission in the third bit of intermission, the system integrator has to consider this fault when calculating the maximum allowed oscillator tolerance for the overall CAN system.

Restriction: As already mentioned, the maximum allowed oscillator tolerance and the maximum bus length will be reduced.

Note: This problem impacts the CAN bit timing configuration. Using crystal/oscillators instead of resonators will mitigate the bit timing issues.

Advisory*24x CAN Does Not Send a Passive Error Frame*

Revision(s) Affected: 2.1

Details: This finding is closely linked to the *24x CAN Does Not Stop Sending Active Error Flags (2)* advisory. The 24x CAN is not able to switch to error-passive state as long as it cannot detect its active error flag on the bus.

Workaround: Problem can occur when there is a fault on the CAN network. The DSP has to be reset.

Advisory*Back-to-Back Data Transmit/Receive Within the Same CAN Module May Cause Sporadic Loss of Data Integrity*

Revision(s) Affected: 2.1

Details: Configure mailboxes 0–2 as receive and mailboxes 3–5 as transmit. Verify transmit and receive mailbox values during back-to-back data transfer for over 100/1000 times (i.e., CAN bus never becomes idle). This process will fail sporadically. Some of the receive or transmit data is not successful. This problem does not occur if only transmission or reception is enabled.

Workaround: If possible, avoid back-to-back high-density data transfer. Add some delay in the data transmit or receive side to avoid high-traffic condition on the CAN bus.

Advisory*Method to Service Multiple Interrupts Through the Existing Two-Interrupt Structure*

Revision(s) Affected: 2.1

Details: A CAN module interrupt will not assert a core-level interrupt if a CAN_IFR bit (set by any previous CAN interrupt) still remains set at the time of occurrence of the new interrupt. For example, when a new CAN interrupt is asserted before the processor has had a chance to clear the CAN_IFR bit that caused the interrupt to happen, the CAN module does not assert a core-level CPU interrupt (INTn) for the new interrupt and may “lose” interrupts. However, the occurrence of any interrupt-causing event will always set the appropriate flag bit in the CAN_IFR register.

Workaround: The CAN_IFR bit that causes an interrupt must be cleared as early as possible within the ISR. This can be done by copying the CAN_IFR register to a memory variable and then clearing the set bit. The memory variable can then be examined to determine which interrupt was asserted. If all CAN_IFR bits are cleared (i.e., zero) when a “new” interrupt is asserted, the “new” interrupt will be recognized and serviced by the core.

Advisory*AME and AAM Bits Get Corrupted Upon Reception*

Revision(s) Affected: 2.1

Details:

This problem surfaces only when “Message filtering” or “Automatic Answer Mode” is used. It applies to any mailbox configured for reception or for mailboxes configured for “automatic reply” to remote requests. It affects the AAM and AME bits of a mailbox after a data-frame (or a remote-frame) is received in that mailbox and leads to a possible corruption of these bits.

This corruption is experienced **only** when a CPU read/write access of any mailbox RAM is executed in the system clock cycle prior to the first write access (to the receive mailbox RAM) of the received data. The AAM and AME bits flip to the state of the CPU accessed data, thereby changing from their original values. A typical scenario is when bits 13 and 14 of any mailbox RAM were read/written as zero exactly one clock cycle prior to the reception of data in a receive mailbox. This will change bits 13 and 14 of the MSGID register of the mailbox that just received the data to zero. Therefore, the AAM and AME bits have to be considered corrupt after any reception.

Problem Effect:

AME bit corruption: When the AME bit of a receive mailbox is unintentionally changed to zero, it effectively shuts down that particular mailbox, since the acceptance mask filtering criterion will not be satisfied anymore. With time, this phenomenon could take place in other receive mailboxes and shut them down as well, effectively preventing the CAN module from receiving any further messages.

AAM bit corruption: When the AAM bit of a mailbox is unintentionally changed to zero, it prevents the CAN node from automatically responding to a “Remote request”. For a mailbox configured for “Automatic Answer Mode,” (i.e., for a mailbox configured as a transmit mailbox with the AAM bit set), this may prevent the CAN module from responding to a remote-request.

Workaround: Reinitialize these bits to the appropriate value after every reception.

Advisory*CAN Bit Timing*

Revision(s) Affected: 2.1

Details:

When the SBG bit (bit 10 of the BCR1 register) is set to 1, the “high” bit may be one TQ shorter/longer than it should be.

Workaround: Set SBG = 0.

4 Programming Considerations for the 24x ADC

4.1 Checking for End-of-Sequence

After a start-of-conversion (SOC) is initiated, two NOPs need to be executed before polling the ADCEOC bit. The EOC bit does not get set until 2 cycles after the write to the ADCTRL1 register (i.e., 2 cycles are needed before the correct EOC value is read). This timing is independent of the ADC clock prescale value. This (inserting 2 NOPs) does not apply if the ADCINTFLAG is polled instead. To reiterate, the NOPs are required only when polling the ADCEOC bit; interrupt-driven conversions do not have this requirement.

5 Errata for *TMS320F243, TMS320F241 DSP Controllers Data Sheet (SPRS064C)*

5.1 *10-Bit Dual Analog-to-Digital Converter (ADC) Section, Recommended Operating Conditions Table (p. 103)*

The MIN and MAX specifications for V_{AI} should be V_{REFLO} and V_{REFHI} , respectively, not V_{SSA} and V_{CCA} .

5.2 *Figure 16, TMS320x24x DSP Device Nomenclature, (p.73)*

Figure 16 shows the device nomenclature for **all** TMS320x24x devices—240, 241, 242, and 243. The Q temperature range (Q = -40°C to 125°C , Q 100 Fault Grading) applies to the F240 device only, and does **not** apply to the F243 and F241 devices.

6 Errata for *TMS320F24x DSP Controllers Reference Guide: Systems and Peripherals (SPRU276C)*

6.1 *Section 3.3, Illegal Address Detect*

Check for illegal address space is made for Program, Data and I/O spaces and *not* for data memory space *only*. Refer to the latest data sheets for the correct memory layout.

6.2 *Section 11.4.4, Master Control Register (MCR)*

The description for the DBO bit (bit 10) is reversed.

6.3 *Section 8.4.1, ADCTRL1 Register*

The correct description for bits 15 and 14 when they are both zero is “Reserved” and not “Stop conversion immediately” as mentioned.

7 Documentation Support

For device-specific data sheets and related documentation, visit the TI web site at: <http://www.ti.com>

To access documentation on the web site:

1. Go to <http://www.ti.com>
2. Click on **TMS320™ DSP Products**
3. Scroll to “**C24X™ DSP Generation**” and click on **DEVICE INFORMATION**
4. Click on a device name and then click on the documentation type you prefer.

For further information regarding the TMS320F243 or the TMS320F241, please refer to the following publications:

- *TMS320F243/F241/C242 DSP Controllers Reference Guide: System and Peripherals*, literature number SPRU276
- *TMS320F/C24x DSP Controllers Reference Guide: CPU and Instruction Set*, literature number SPRU160
- *TMS320F20x/F24x DSP Embedded Flash Memory Technical Reference*, literature number SPRU282
- *TMS320F243, TMS320F241 DSP Controllers data sheet*, literature number SPRS064

TMS320 and C24x are trademarks of Texas Instruments.
Other trademarks are the property of their respective owners.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI 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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

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

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265