

TMS320C5535/34/33/32

Fixed-Point Digital Signal Processor

Silicon Revision 2.2

Silicon Errata



Literature Number: SPRZ373

January 2012

1	Introduction	5
1.1	Device and Development-Support Tool Nomenclature	5
1.2	Revision Identification	6
2	Silicon Revision 2.2 Usage Notes and Known Design Exceptions to Functional Specifications	7
2.1	Usage Notes for Silicon Revision 2.2	7
2.1.1	Master Clock Gating With WAKEUP, INT0 , or INT1 Asserted	7
2.1.2	Serial Boot Modes Only Support 16-bit Address Mode	7
2.1.3	Reserved Bits in the RTC Oscillator Register (RTCOSC) [0x192C]	7
2.1.4	Two 1149.1 JTAG Tap Controllers for JTAG Pins (TRST , TCK, TMS, TDI, TDO)	8
2.1.5	Bootloader Disables Peripheral Clocks	8
2.2	Silicon Revision 2.2 Known Design Exceptions to Functional Specifications	9
	Revision History	17

List of Figures

1	Example and Device Revision Codes	6
2	CPU, USB, LCD, and eMMC/SD Data Paths	10
3	USB DMA Read	11
4	CPU Read From USB.....	11

List of Tables

1	Device Revision Codes	6
2	Silicon Revision 2.2 Advisory List.....	9
3	Transmit Path Internal Data Delays.....	12
4	Receive Path Internal Data Delays	12
5	Feedback Path Internal Data Delays	12
6	DMA Transfer Lengths.....	14

TMS320C5535, TMS320C5534, TMS320C5533, TMS320C5532 Silicon Revision 2.2

1 Introduction

This document describes the known exceptions to the functional specifications for the TMS320C553x devices (i.e., TMS320C5535, TMS320C5534, TMS320C5533 and TMS320C5532). For more detailed information on these devices, see the device-specific data manual (literature number [SPRS737](#)).

Throughout this document, unless otherwise specified, TMS320C553x and C553x, refer to the TMS320C5535, TMS320C5534, TMS320C5533 and TMS320C5532 devices.

The advisory numbers in this document are not sequential. Some advisory numbers have been moved to the next revision and others have been removed and documented in the user's guide. When items are moved or deleted, the remaining numbers remain the same and are not resequenced.

1.1 Device and Development-Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all DSP devices and support tools. Each DSP commercial family member has one of three prefixes: TMX, TMP, or TMS (e.g., TMS320C5535AZHHA10). Texas Instruments recommends two of three possible prefix designators for its support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (TMX/TMDX) through fully qualified production devices/tools (TMS/TMDS).

Device development evolutionary flow:

TMX — Experimental device that is not necessarily representative of the final device's electrical specifications.

TMP — Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification.

TMS — Fully-qualified production device.

Support tool development evolutionary flow:

TMDX — Development-support product that has not yet completed Texas Instruments internal qualification testing.

TMDS — Fully qualified development-support product.

TMX and TMP devices and TMDX development-support tools are shipped against the following disclaimer:

"Developmental product is intended for internal evaluation purposes."

TMS devices and TMDS development-support tools have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

Predictions show that prototype devices (TMX or TMP) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices are to be used.

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, ZHH), the temperature range (for example, "Blank" is the commercial temperature range), and the device speed range in megahertz (for example, "10" is the default 100 MHz device).

1.2 Revision Identification

Figure 1 provides an example(s) of the TMS320C553x device markings. The device revision can be determined by the symbols marked on the top of the package.

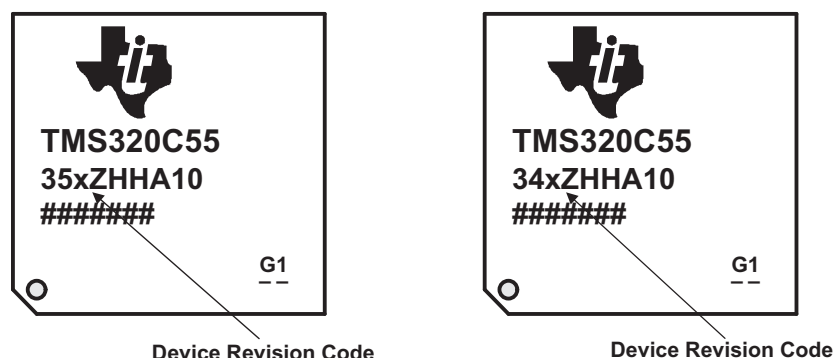


Figure 1. Example and Device Revision Codes

Silicon revision is identified by a device revision code marked on the package and through software. The code on the package is of the format C5535x or C5534x, where "x" denotes the silicon revision. If x is "A" in the device part number, it represents Silicon Revision 2.2 TMS devices. Table 1 lists the information associated with each silicon revision.

Table 1. Device Revision Codes

DEVICE PART NUMBER DEVICE REVISION CODE (x)	SILICON REVISION	PART NUMBERS/COMMENTS
A	2.2	This silicon revision is available as TMS <i>only</i> . TMS320C5535AZHH, TMS320C5534AZHH, TMS320C5533AZHH, TMS320C5532AZHH

Through software, the user can read bits 15-12 of the I/O space Die ID Register 3 (DIEIDR3) [1C43h].

2 Silicon Revision 2.2 Usage Notes and Known Design Exceptions to Functional Specifications

This section describes the usage notes and advisories that apply to silicon Revision 2.2 of the TMS320C553x devices.

2.1 Usage Notes for Silicon Revision 2.2

Usage notes highlight and describe particular situations where the device's behavior may not match presumed or documented behavior. This may include behaviors that affect device performance or functional correctness. These usage notes will be incorporated into future documentation updates for the device (such as the device-specific data sheet), and the behaviors they describe will not be altered in future silicon revisions.

2.1.1 Master Clock Gating With WAKEUP, $\overline{\text{INT0}}$, or $\overline{\text{INT1}}$ Asserted

On silicon Revision 2.2, the DSP can disable the Master Clock by setting bit 15 of the PCGCR register (0x1C02). Once the master clock is disabled, it can only be re-enabled by one of the following events:

- Hardware reset being asserted ($\overline{\text{RESET}}$ = low)
- An enabled RTC alarm or periodic interrupt occurring
- The $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ pins being asserted (low) (level-sensitive)
- The WAKEUP pin being asserted (high) (level-sensitive)

When the master clock is disabled, there are no clocks for edge detection and therefore the $\overline{\text{INT0}}$, $\overline{\text{INT1}}$, and WAKEUP pins are level-sensitive. This means that a low on either the $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ or a high on the WAKEUP pin will force bit 15 of the PCGCR register to "0", enabling the master clock. Attempting to write a "1" to bit 15 of the PCGCR register while $\overline{\text{INT0}}$, $\overline{\text{INT1}}$, or WAKEUP are asserted will be unsuccessful since re-enabling the clocks has a higher priority than disabling them.

When the WAKEUP pin is configured as an output-pin, the WAKEUP pin only functions as a GPO and no longer functions as a WAKEUP pin to re-enable the master clocks. When the WAKEUP pin is configured as an input-pin, the WAKEUP pin's state must be low to disable the master clocks.

2.1.2 Serial Boot Modes Only Support 16-bit Address Mode

On silicon Revision 2.2, the DSP supports the following boot modes: SPI EEPROM, SPI Serial Flash or I2C EEPROM, eMMC/SD/SDHC, UART, and USB. The SPI EEPROM boot supports both 16-bit and 24-bit address modes, while the I2C EEPROM boot mode *only* supports 16-bit address mode.

2.1.3 Reserved Bits in the RTC Oscillator Register (RTCOSC) [0x192C]

For proper device operation on silicon Revision 2.2, the "RESERVED" bits in the RTCOSC register (0x192C) should always be set to "zero".

2.1.4 Two 1149.1 JTAG Tap Controllers for JTAG Pins ($\overline{\text{TRST}}$, TCK, TMS, TDI, TDO)

The silicon Revision 2.2 C553x devices have two internal 1149.1 JTAG Tap controllers but only one set of corresponding JTAG pins ($\overline{\text{TRST}}$, TCK, TMS, TDI, TDO). One TAP controller supports emulation and the other supports JTAG 1149.1 Boundary Scan. Only one of the two TAPs is internally connected to the pins at a time and it is the latched state of the EMU0 pin that determines which TAP is connected. The EMU0 pin is latched on the rising edge of $\overline{\text{TRST}}$ and from that time forward the selected tap is connected to the pins. If the latched state of EMU0 is "0", the boundary scan tap is selected and customers may perform boundary scan testing. If the latched state of EMU0 is "1", the DSP's emulation tap is selected and customers may perform emulation with TI's Code Composer Studio™ IDE Emulation Debugger.

Note: Because of the device's internal (and recommended external) pullup on the EMU0 pin and the fact that the emulation pods (e.g., XDS560) do not drive the EMU0 pin while $\overline{\text{TRST}}$ is driven low-to-high, the emulation tap will normally be the one selected. However, customers who wish to do boundary scan testing will need to have an external pulldown (2 k Ω is recommended), with sufficient strength to overcome the internal pullup, so that the boundary scan tap is connected to the JTAG pins.

2.1.5 Bootloader Disables Peripheral Clocks

After hardware reset, on silicon Revision 2.2 C553x devices, the DSP boots via the bootloader code in ROM. During the boot process, the bootloader queries each peripheral to determine if it can boot from the peripheral. At that time, the required peripheral's clock will be enabled for the query and then disabled when the bootloader is finished with the peripheral. By the time the bootloader releases control to the user code, all peripheral clocks will be off and all domains in the ICR, except the CPU domain, will be idled. After the boot process is complete, the user is responsible for enabling and programming the required clock configuration for the DSP.

For example on the C5535 device, the bootloader disables both the MPORT and FFTHWA. To enable the MPORT and FFT HWA, write 0x000E to the ICR registers and issue an "idle" command.

Assembly Code Example:

```
*port(#0x0001) = #(0x000E)
idle
```

C Code Example:

```
*(ioport volatile unsigned *)0x0001 = 0x000E;
asm("    idle");    // must add at least one blank before idle in " ".
```

For example on the C5534 device, the bootloader disables the MPORT. To enable the MPORT, write 0x0002E to the ICR registers and issue an "idle" command.

Assembly Code Example:

```
*port(#0x0001) = #(0x0002E)
idle
```

C Code Example:

```
*(ioport volatile unsigned *)0x0001 = 0x0002E;
asm("    idle");    // must add at least one blank before idle in " ".
```


2.2 Silicon Revision 2.2 Known Design Exceptions to Functional Specifications

Table 2. Silicon Revision 2.2 Advisory List

Title	Page
Advisory 2.0.1 — USB: Endianness Incompatibility	10
Advisory 2.0.4 — I2S: I2S Internal Data Delay	12
Advisory 2.0.5 — USB: USB Queue Manager Reads Only 16-bit Address of USB Descriptors	13
Advisory 2.0.8 — DMA: DMA Transfer Length Must be a Multiple of $4 \times 2^{(\text{Burst Mode})}$	14
Advisory 2.0.9 — USB Boot Does Not Work if the On-Chip USB_LDO Supplies Power to the USB Core	15
Advisory 2.0.10 — USB CPPI Receive Starvation Interrupt	16

Advisory 2.0.1 **USB: Endianness Incompatibility**

Revision(s) Affected 2.2

Details

The C553x CPU is a word addressable and big endian architecture. The CPU interfaces to the rest of the system through several ports: MPORT, XPORT, DPORT, and IPORT. The DMA transfers data from peripherals to on-chip memory through the MPORT in 32-bit packets. The CPU accesses the peripherals through the XPORT in 8- or 16-bit packets. The CPU accesses external memory in 8-bit (configured through a system register setting), 16-bit (through CPU single word access), or 32-bit (through CPU double-word access). External data accesses occur through the DPORT.

Some peripherals (e.g., USB, LCD, and eMMC/SD) have sensitivity to the data endianness. LCD: Endianness is controlled through software. Default is little endian. MMC/SD: Endianness is controlled through software. Default is big endian. USB: DMA transfer to/from USB buffer is big endian. CPU XPORT access to USB buffer is little endian. USB endianness is not software controllable. The two endiannesses of the USB could result in inter byte swap.

Figure 2 shows data paths that could create byte/word swaps.

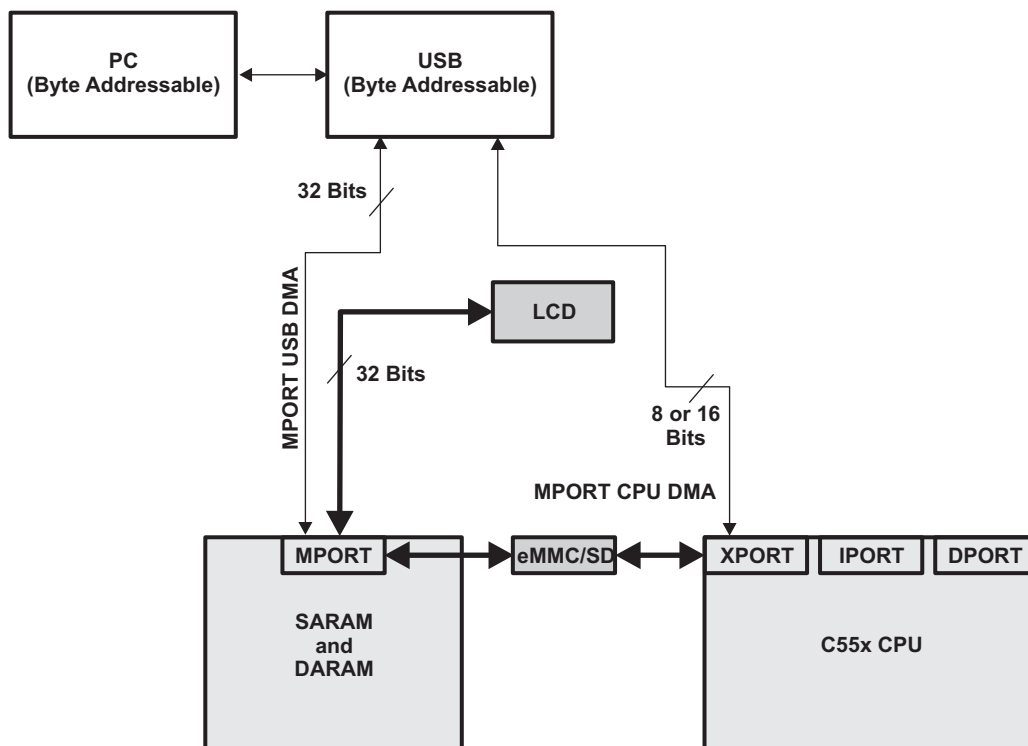


Figure 2. CPU, USB, LCD, and eMMC/SD Data Paths

Example Case: Data transfer between the USB and CPU (see [Figure 3](#) and [Figure 4](#)).

Data can be read from the USB by the USB DMA or CPU. The USB DMA accesses on-chip memory through the MPORT and only performs a 32-bit read while the CPU can perform a 16-bit read via the XPORT. When USB DMA transfers data from the USB buffers to on-chip memory, the data transfer is 32-bits and is handled in big endian fashion, so no data swap occurs (see [Figure 3](#)). However, when the CPU, through the XPORT, accesses the USB buffers, the data is accessed 16-bits at a time in little endian fashion resulting in an inter byte swap (see the [Figure 4](#)).

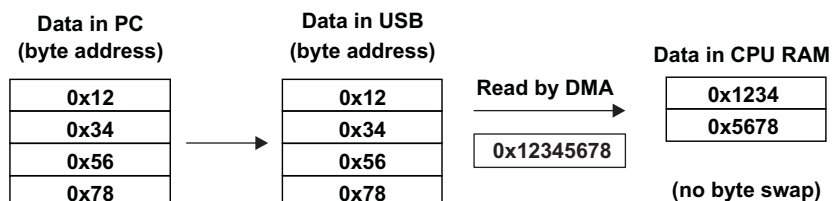


Figure 3. USB DMA Read

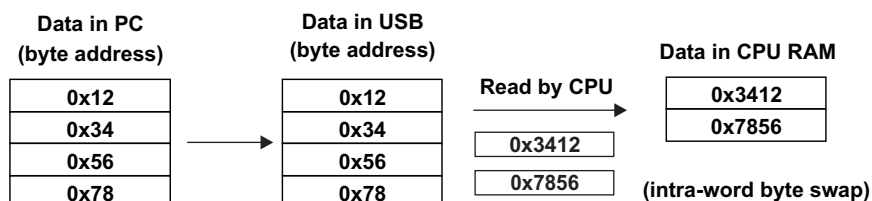


Figure 4. CPU Read From USB

Workaround(s)

To correct this issue, software is required to fix the intra-word byte swap on the data prior to processing on the device CPU.

Advisory 2.0.4 I2S: I2S Internal Data Delay
Revision(s) Affected 2.2

Details

The I2S module has an internal delay for the data transmit/receive path that varies depending on the settings of the Pack bit and Word Length in the I2S Control Register and the FSDIV in the Sample Rate Generator Register.

[Table 3](#) shows the transmit path internal data delays. Feedback path refers to an I2S transmit pin that is externally connected to the I2S receive pin.

Table 3. Transmit Path Internal Data Delays

PACK	DATA DELAY
1	The first five transmit frames will be zero data. On the sixth transmit frame, the data written to the transmit register will be shifted out on the DX pin.
0	The first three transmit frames will be zero data. On the fourth transmit frame, the data written to the transmit register will be shifted out on the DX pin.

[Table 4](#) shows the receive path internal data delays:

Table 4. Receive Path Internal Data Delays

WORD LENGTH	DATA DELAY
8-bit	The RX data registers will contain zero data for the first two frames.
10, 12, 14, and 16-bit	The RX data registers will contain zero data for the first three frames.
18, 20, 24, and 32-bit	The RX data registers will contain zero data for the first two frames.

[Table 5](#) shows the feedback path internal data delays:

Table 5. Feedback Path Internal Data Delays

PACK	WORD LENGTH	FSDIV BITS I2SSRATE.[5:3]	DATA DELAY
1	8-bit	000	Data received on the 7th sample
	8-bit	001 – 101	Data received on the 6th sample
	10-, 12-, 14-, and 16-bit	001	Data received on the 5th sample
	10-, 12-, 14-, and 16-bit	010 – 101	Data received on the 4th sample
0	8-, 10-, 12-, 14-, 16-, 18-, 20-, 24-, and 32-bit	000 – 101	Data received on the 3rd interrupt

Workaround(s) Zero data should be Ignored.

Advisory 2.0.5 *USB: USB Queue Manager Reads Only 16-bit Address of USB Descriptors*

Revision(s) Affected 2.2

Details

The C553x has 23-bit address space but the USB Queue Manager (QMGR) is a 32-bit register that holds the address of the USB descriptors. A descriptor itself is a structure with information about the addresses of the source/destination data buffers and their sizes. The address of a particular descriptor is written to the QMGR register for a particular DMA endpoint. The CPU writes the address of a descriptor to the QMGR register for a for a DMA endpoint. The QMGR fires the USB DMA to read the descriptor at the address pointed to in the QMGR register and sets up the DMA endpoints for future transfers. When a USB host connects and performs a transfer, the QMGR copies the address of the descriptor to a completed queue. Upon receiving the USB interrupt, the USB driver should read the 32-bit descriptor address in the QMGR completion queue to determine which DMA endpoint has completed transferring data. Even though the CPU can write a 32-bit value into the QMGR register, it can only read the lower 16-bits of this register. Thus, the descriptor can only be allocated in the CPU memory map to the same lower 16-bit address and all descriptors must be placed in one contiguous block of 64K words in SARAM.

For example:

- USB descriptor A is located at 0x008000
- USB descriptor B is located at 0x018000

The descriptor A and B will be considered the same descriptor.

Workaround(s) The USB descriptors should be placed in ONE CONTIGUOUS BLOCK of 64K words (2^{16}) in memory.

Advisory 2.0.8 DMA: DMA Transfer Length Must be a Multiple of $4 \times 2^{(\text{Burst Mode})}$
Revision(s) Affected 2.2

Details If the transfer length register has a value that is zero or *not* a multiple of $4 \times 2^{(\text{Burst Mode})}$ when the DMA transfer begins, it will cause an unexpected operation of DMA.

While the DMA transfers 32-bit words from a source address to a destination address, the value in the DMA transfer length register is the length in bytes. For example, if the total DMA transfer length is 4 32-bit words and the burst size = 1, $4 \times 4 = 16$ should be written to the DMA transfer length register. The burst size should also be considered. Burst size is the minimum data transfer size; therefore, the total DMA Transfer Length should be $4 \times 2^{(\text{Burst Mode})}$. For more details, see [Table 6](#)

Table 6. DMA Transfer Lengths

BURST MODE	BURST SIZE (32-BIT WORD)	DMA TRANSFER LENGTH (BYTES)
0	1	Multiple of 4 (minimum 4)
1	2	Multiple of 8 (minimum 8)
2	4	Multiple of 16 (minimum 16)
3	8	Multiple of 32 (minimum 32)
4	16	Multiple of 64 (minimum 64)

Workaround(s) Write only a multiple of $4 \times 2^{(\text{Burst Mode})}$ to the DMA transfer register before the DMA starts.

Advisory 2.0.9	<i>USB Boot Does Not Work if the On-Chip USB_LDO Supplies Power to the USB Core</i>
-----------------------	--

Revision(s) Affected	2.2
-----------------------------	-----

Details	<p>The on-chip USB_LDO is intended to power the USB_ V_{DD1P3} and USB_ V_{DDA1P3} pins. At reset, the USB_LDO is turned off and can be turned on via the USBLDOEN bit (bit 0) in the LDOCNTL register (0x7004). However, the on-chip Bootloader does not enable the USB_LDO before attempting USB boot, so USB boot will not work if the USB_LDO supplies power to the USB_ V_{DD1P3} or USB_ V_{DDA1P3} pins.</p>
----------------	---

Workaround(s)	<p>For the applications that require USB boot, the USB Core (USB_ V_{DD1P3}, USB_ V_{DDA1P3}) must be powered externally.</p>
----------------------	---

Advisory 2.0.10 ***USB CPPI Receive Starvation Interrupt***

Revision(s) Affected 2.2

Details

When an endpoint is enabled for receive transfer(s) that will be serviced via CPPI DMA and data has been received prior to allocating the DMA resource, the DMA will generate a starvation interrupt to notify the application a lack of resource (starvation) in anticipation that the application will furnish the required resource. Once the starvation occurs, the CPPI DMA continues generating interrupts periodically whenever the host tries to send data. It does not stop until the application furnishes a resource.

The USB starvation interrupt is always enabled and cannot be masked off at the USB controller level. Since the DMA continues to generate the starvation interrupt periodically and there exists no capability to mask the starvation interrupt at the USB controller level, the CPU is forced either to fully service the DMA interrupt as it is received or disable all USB interrupts at the CPU level. Disabling the entire USB interrupt might not be the desired option since the CPU needs to be aware of other USB interrupts that are more critical.

Workaround(s)

Dedicated data receiving buffers are recommended. The data buffers should be allocated and available to the CPPI DMA during USB initialization. The required data buffer size is highly dependent on USB host applications. For Windows XP USB Mass Storage device driver, it is recommended to allocate at least 64 KB (128 descriptors; 128*512 = 64 KB) of data buffer space.

Revision History

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

Scope: Applicable updates relating to the devices have been incorporated. Added device-specific information supporting the TMS320C553x Silicon Revision 2.2 devices.

Revision History

SEE	ADDITIONS/MODIFICATIONS/DELETIONS
-----	-----------------------------------

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

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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Mobile Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Automotive and Transportation	www.ti.com/automotive
Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Video and Imaging	www.ti.com/video

TI E2E Community Home Page

e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2012, Texas Instruments Incorporated