TEXAS INSTRUMENTS

Design Guide
SPRAA18D − October 2005

TMS320C6413 Hardware Designer's Resource Guide

Kevin Jones *DSP Hardware Applications Team*

ABSTRACT

The TMS320C6413 DSP Hardware Designer's Resource Guide is a collection of the most commonly used technical documentation, and models for DSP hardware system designers. Topics covered include Getting Started, Models and Symbols, Bootloading, and Checklists to aid in your initial design and debug efforts. Each section includes pointers to valuable information including technical documentation, models, symbols, and reference designs for use in each phase of design.

Contents

List of Tables

1 Getting Started

1.1 Registering on my.TI

my.TI is a customizable area within the Texas Instruments web site. By registering on my.TI, you can receive the following benefits:

- Quick Reference to information you select as part of your profile.
- Email alerts that inform you of updates to products, technical documentation, and errata.
- The my.TI newsletter providing information on the latest innovations and product releases.

To register on my.TI for updates related to these devices:

1. Go to the device product folders.
2. Select the link called “ADD To myTI” in the upper right hand corner, and follow the on-screen instructions.
3. Select Customize my.TI to specify what you would like to receive notification about.

The following is a link to the product folder.

TMS320C6413 Product Folder

1.2 Training and Support

Texas Instruments offers a variety of training options tailored for your specific needs and requirements. Options include on-line training, webcasts, seminars, single and multi-day workshops, and conferences. For more information about training, visit Texas Instruments Training Home. For assistance with technical questions regarding TI Semiconductor products and services, you can access the Semiconductor Technical Support KnowledgeBase .

1.3 Technical Documentation

1.3.1 Where to Start

The key area for obtaining documentation for these devices are the product folders. When getting started, it is of great importance to have the latest data sheet and silicon errata. Listed below are links to this key information:

- TMS320C6413 Product Folder
- TMS320C6413, TMS320C6410 Fixed-Point Digital Signal Processors Data Manual (SPRS247)
- TMS320C6413, TMS320C6410 Fixed-Point Digital Signal Processors Silicon Errata (SPRZ219)

1.3.2 Using TI Literature Numbers

All TI documentation is assigned a literature number. This number can be used to search for the document on the Web. Technical documentation revisions are indicated by the alpha character at the end of the literature number on the title page, and in the file name.

Use the literature number (without the trailing alpha character) to search the TI website for the document. For example, if a data manual has a literature number of SPRS205B, the ”B” indicates the revision of the document. If the document has no trailing alpha character, it is the original version of the document. When searching for this document on the TI web site, you can simply enter ”SPRS205” as the search keyword.

1.3.3 Peripheral Reference Guides

Each peripheral has a reference guide that provides beneficial information for completing a design. Each peripheral and its respective reference guide is listed here. There are two categories. The first category contains peripherals which connect directly to external devices. The second category lists the internal peripherals.

Peripherals that connect directly to external devices:

- TMS320C6000 DSP Inter-Integrated Circuit (I2C) Module Reference Guide (SPRU175)
- TMS320C6410/C6413 DSP Inter-Integrated Circuit (I2C) Module (SPRZ221)
- TMS320C6000 DSP Multichannel Audio Serial Port (McASP) Reference Guide (SPRU041)
- TMS320C6000 DSP External Memory Interface (EMIF) Reference Guide (SPRU266)
- TMS320C6000 DSP General-Purpose Input/Output (GPIO) Reference Guide (SPRU584)
- TMS320C6000 DSP Host-Port Interface (HPI) Reference Guide (SPRU578)
- TMS320C6000 DSP Multichannel Buffered Serial Port (McBSP) Reference Guide (SPRU580)

Internal peripherals:

- TMS320C6000 DSP Two-Level Internal Memory Reference Guide (SPRU610)
- TMS320C6000 DSP Interrupt Selector Reference Guide (SPRU646)
- TMS320C6000 DSP 32-bit Timer Reference Guide (SPRU582)
- TMS320C6000 DSP Enhanced Direct Memory Access (EDMA) Controller Reference Guide (SPRU234)

1.3.4 Application Reports

Application reports are documents written to describe functionality and usability for a specific application or design. Listed below are application reports that provide useful information about peripherals.

External Memory Interface (EMIF):

- TMS320C6000 EMIF-to-External SDRAM Interface (SPRA433)
- TMS320C6000 EMIF to External Flash Memory (SPRA568)
- TMS320C6000 EMIF to External Asynchronous SRAM Interface (SPRA542)
- TMS320C6000 EMIF to TMS320C6000 Host Port Interface (SPRA536)

Multichannel Buffered Serial Port (McBSP):

- *TMS320C6000 McBSP Interface to an ST-BUS Device (SPRA511)*
- *Using the TMS320C6000 McBSP as a High Speed Communication Port (SPRA455)*
- *TMS320C6000 McBSP to Voice Band Audio Processor (VBAP) Interface (SPRA489)*
- *TMS320C6000 McBSP: AC’97 Codec Interface (TLV320AIC27) (SPRA528)*
- *TMS320C6000 McBSP Interface to SPI ROM (SPRA487)*
- *TMS320C6000 McBSP: IOM-2 Interface (SPRA569)*
- *TMS320C6000 McBSP: UART (SPRA633)*
- *TMS320C6000 McBSP as a TDM Highway (SPRA491)*
- *TMS320C6000 Multichannel Communications System Interface (SPRA637)*
- *TMS320C6000 McBSP: I^2S Interface (SPRA595)*

Host Port Interface (HPI):

- *TMS320C6000 Host Port to MC68360 Interface (SPRA545)*
- *TMS320C6000 Host Port to the i80960 Microprocessors Interface (SPRA541)*
- *TMS320C6000 Host Port to MPC860 Interface (SPRA546)*
- *TMS320C6000 Host Port to the i80960 Microprocessors Interface (SPRA541)*

2 Board Design and Layout

2.1 High−Speed DSP Systems Design Reference Guide

Today’s digital signal processors (DSPs) are typically run at a 1GHz internal clock rate while transmit and receive signals to and from external devices operate at rates higher than 200MHz. These fast switching signals generate a considerable amount of noise and radiation, which degrades system performance and creates electromagnetic interference (EMI) problems that make it difficult to pass tests required to obtain certification from the Federal Communication Commission (FCC). Good high−speed system design requires robust power sources with low switching noise under dynamic loading conditions, minimum crosstalk between high−speed signal traces, high− and low−frequency decoupling techniques, and good signal integrity with minimum transmission line effects. This document provides recommendations for meeting the many challenges of high−speed DSP system design.

For more information, refer to High−Speed DSP Systems Design Reference Guide (SPRU889).

2.2 Schematics

This section includes ORCAD symbols to assist you in schematic generation. The method chosen to provide the ORCAD symbols is a multi-section or “heterogeneous” package. The logical parts in the package have different graphics, numbers of pins, or properties. This allows the schematic designer to easily connect the various parts on separate schematic pages and yet retain the same reference designator for the part across all schematic pages.

Following are links to download the ORCAD symbols for these devices:

TMS320C6413 ORCAD Symbol (SPRC173)

2.3 Signal Integrity and Timing Considerations

Today’s high-speed interfaces require strict timings and accurate system design. To achieve the necessary timings for a given system, input/output buffer information specification (IBIS) models must be used. These models accurately represent the device drivers under various process conditions. Board characteristics, such as impedance, loading, length, number of nodes, etc., affect how the device driver behaves. The following key information to support these devices.

- TMS320C6413 IBIS Model
- Using IBIS Models for Timing Analysis (SPRA839)

2.4 Board Layout

The significance of electromagnetic compatibility (EMC) of electronic circuits and systems has recently been increasing. This increase has led to more stringent requirements for the electromagnetic properties of equipment. Two property aspects are of interest: the ability of a circuit to generate the lowest (or zero) interference, and the immunity of a circuit to the effects of the electromagnetic energy it is subjected to. The effects on electronic circuits and systems is well documented, but little attention has been paid to circuit behavior and the interference it generates. The following link discusses the important criteria that determine the EMC of a circuit.

Printed-Circuit Board Layout for Improved Electromagnetic Compatibility (SDYA011).

2.5 Power Supply and Sequencing Considerations

Texas Instruments offers several Power Management Products for these devices. In the Power Management selection guide below, refer to the TMS320C6000 section for more information on available solutions. For a complete list of product offerings, visit the power.ti.com website. The following applications reports are helpful in choosing the right power solution:

- *Power Management Selection Guide (SLVT145)*
- *Providing a DSP Power Solution from a 5-V or 3.3-V Only System (SLVA069)*
- *Power Supply Sequencing Solutions for Dual Supply Voltage DSPs (SLVA073)*
- *Dual Output Power Supply Sequencing for High Performance Processors (SLVA117)*

2.6 Power/Thermal Management Considerations

Circuit designers must always consider the effects of thermal stack up – the cascaded effect of the transfer of heat from a device die to the surrounding package. The flow of heat from the device die to ambient must be sufficient to maintain an acceptably low junction temperature, and maximize device reliability. The thermal resistance characteristics for this device are documented in the data sheet. The following application reports discuss thermal analysis, heat sink selection, and power consumption.

TMS320C6x Thermal Design Considerations (SPRA432)

2.7 Boot Mode Configurations

The boot process is determined by the boot configuration stemming from the pull-up/pull-down resistors on the EMIFA address bus, AEA[22:21]. Table 1 shows the possible configurations.

Table 1. Boot Mode

BOOTMODE	Boot Process
00	Boot (Default)
01	HPI Boot (based on $\overline{\text{HPI-EN}}$ pin)
10	Reserved
11	EMIFA 8-bit ROM boot

The three possible boot processes are:

- **No boot process** − The CPU begins direct execution from the memory located at address 0.
- **ROM boot process** − The program located in external ROM is copied to address 0 by the EDMA Controller. This transfer occurs while the CPU is held in reset, so that once it is released, it can start executing code from location 0. For C64x devices, the EDMA copies the first 1K Bytes from the beginning of CE1 (on EMIFA) to address 0.
- **Host boot process** − The CPU is held in reset while the rest of the device is released. During this time, an external host can appropriately change any internal memory and/or registers needed. When it is finished, it needs to set the DSPINT to complete. This action will release the CPU from reset and cause it to start executing from location 0.

The following are links to key information about bootloading:

- TMS320C6413, TMS320C6410 Fixed-Point Digital Signal Processors Data Manual (SPRS247).
- TMS320C6000 Tools: Vector Table and Boot ROM Creation (SPRA544)
- TMS320C6000 HPI Boot Operation (SPRA512)

2.8 Joint Test Action Group (JTAG) Interface

DSP devices have a JTAG interface that allows for emulation hardware and software to communicate with the DSP. The JTAG port also supports boundary scan testability. Listed below are links to this key information.

- TMS320C6000 Board Design: Considerations for Debug (SPRA523)
- TMS320C6000 Board Design for JTAG (SPRA584)
- TMS320C6000 DSP Designing for JTAG Emulation Reference Guide (SPRU641)
- 60-pin Emulation Header Technical Reference (SPRU655)
- Emulation Fundamentals for TI’s DSP Solutions (SPRA439)

2.9 Board Manufacturing

When designing with a high−density BGA package, it is important to be aware of different techniques that aid in the quality of the manufacture. The following is a link to this key information.

Flip Chip Ball Grid Array Package Reference Guide (SPRU811)

3 System Test

3.1 Boundary Scan Description Language (BSDL) Model(s)

BSDL models can be used to facilitate board level testing. Currently we have two different versions of the BSDL, one model for revision 1.1 silicon and another model for revision 2.0 silicon. You can find BSDL models for this device here:

TMS320C6413 GTS BSDL Model

4 Design and Debug Checklists

4.1 Design Checklist

The Design Checklist was put together by Texas Instruments application and field support staff as a guide to considerations made during the design phase of development. Use this check list to keep track of considerations you make during the design phase of development.

□	Check data sheet and errata for the most up to date information.
□	Reset circuitry? For debugging it is important to be able to reset the DSP when/if it gets into an unstable state. To perform this, one of the easiest things to do is to have a reset button on the board itself. Having a reset supervisor on board enables you to do things like monitor the supply rails for sags in power. The TPS3110 class of devices are the most commonly used reset supervisors from TI. For more information, see the following data sheet.*Reset Circuitry for the TMS320C6000 DSP (SPRA431)*
□	Are boot mode pins configured correctly? The three boot modes for the C6000 devices are: no boot, a boot over the HPI, or a boot from a ROM device located at Chip Enable Space 1. Check the definition for these modes in the Bootloading Guide section of this document and choose the correct configuration you need. It is very useful to include the ability to choose an alternate boot configuration. Use unpopulated resistor pads to allow the choice of different boot modes.
□	Has the DSP input clock source been correctly selected? The input clock source for these devices can be generated by an external clock source or by the on-chip oscillator. The CLKINSEL pin is used to select between the two clock sources. The OSC_DIS pin enables and disables the on-chip oscillator; it should always be pulled to the same logic level as CLKINSEL. Make sure both pins are pulled to the correct level based on your system's needs. Note that the CLKINSEL and OSC_DIS pins have internal pull−up resistors to disable the on-chip oscillator and select an external clock-source input by default.
□	Is the CLKMODEx pin configured correctly? In addition to checking the CLKMODE[3:0] pins to see if they are set up to generate the correct frequency, the CLKOUT4/6 pins should be checked with an oscilloscope. To do this, you must clock EMIFA. If the CLKOUT4/6 signal is correct, this verifies lock of the PLL, in addition to the correct frequency of operation for your DSP. If the CLKOUT4/6 signal is not correct, check that the PLL and the CLKMODE pins are configured correctly. Does the PLL have the correct circuitry around it, following the data sheet recommendations?
□	Is there a provision for changing the clock during debug time? It can be very helpful to set up a jumper on your board to change the clock frequency. This can allow you to detect whether or not problems are related to the high clock rate.

□ Have all other device configuration pins been properly configured?

The TOUT1/LENDIAN, TOUT0/$\overline{\text{HPI\_EN}}$, and HD5 pins are used to set certain configurations in the device. These pins are latched at reset. The TOUT1/LENDIAN pin selects which endian mode for the device. The TOUT0/HPI_EN and HD5 affect HPI, McASP1, and GP0[15:8] pins as follows:

$\overline{\text{HPI\_EN}}$	HD5	HPI	McASP1	GP0[15:8]
Low	Low	16-bit HPI enabled	Available	Not available
Low	High	32-bit HPI enabled	Not available	Not available
High	X	Not available	Available	Available

The TOUT0/HPI_EN pin is pulled low by default through an internal resistor, while the HD5 and TOUT1/LENDIAN pins are pulled through internal resistors. These pins should be configured through external resistors based on your system needs.

□ Is the emulation configured properly?

In a general sense, can you connect to the board via JTAG? Check to make sure EMU[1:0] is connected according to your needs:

EMU[1:0]	Operation
00	Boundary Scan/Normal Mode
01	Reserved
10	Reserved
11	Emulation/Normal Mode

NOTE: Check the data sheet for more information about these pins. To use with Code Composer Studio, Emulation/Normal mode should be selected. /TRST has an internal pull-down, though it may be useful to include an external pull-down.

In the future, TI will be switching to more advanced emulation using a 60-pin header instead of the traditional 14-pin. See the *60-Pin Emulation Header Technical Reference Guide* (SPRU655). For now, leave the extra emulation pins EMU[11:2] unconnected because they have internal pull-ups. For full details on designing with JTAG, see *IEE Std 1149.1 (JTAG) Testability Primer* (SSYA002). These resources will allow you to check your JTAG circuitry for correctness. The following link provides key information.

IEE Std 1149.1 (JTAG) Testability Primer (SSYA002)

□ If it is a multiprocessor, is the TDI/TDO connection tied properly?

In multiprocessor environments, the TDI (JTAG test-port data in) and TDO (JTAG test-port data out) pins need to be tied correctly. The TDI pin on the JTAG header should tie to the TDI pin on the first DSP, and the TDO pin on the first DSP should tie to the TDI pin on the second DSP. This sequence should continue for subsequent DSPs, until the TDO pin of the last DSP connects to the JTAG header’s TDO pin. For more information on designing for JTAG emulation, see Chapter 16 of the peripherals guide (SPRU190).

□ Is there provision for power sequence?

To operate properly, the DSP needs to be powered up in the correct sequence. Check the sequencing for power, according to data sheet specifications.

□	Voltage levels changes? The board should be able to accommodate some voltage level changes. It can be useful to accommodate some changes by changing a resistor.
□	Are there any GPIOs pinned out to via or LED for probing? The board should be able to accommodate some voltage level changes. It can be useful to accommodate some changes by changing a resistor. GPIOs can be very useful for debugging. If a GPIO pin is available for use, it is worthwhile to pin it out to a via or an LED to observe the operation.
□	Are the HDS/HAS signals correctly configured to access the DSP? If using the HPI, then you have a choice of configurations. Examples of timing diagrams for when $\overline{\text{HAS}}$ is used or unused (tied high) are in *TMS320C6000 DSP Host−Post Interface (HPI) Reference Guide* (SPRU578). Also, there is a choice on how to assert /HSTROBE using both, one, or none of the $\overline{\text{HDS1/2}}$ in combination with $\overline{\text{HCS}}$. The gate logic for these pins should be checked in the reference guide as well.
□	Are the McBSP signals pinned to via for scope trace? For debugging information it can be very useful to have the McBSP signals pinned out to a via. This allows you to check the signals (clock, frame sync, data, etc.) on a scope for correct operation.
□	Are decoupling caps placed on the board near the DSP? Voltages from traces on a printed circuit board can couple to each other in places where it is not desired, (like power supply planes). To decouple the traces, we add capacitors to absorb some of the voltage and help reduce this effect. For more information on how to correctly place decoupling caps, see the Design FAQ.
□	Do the DSP vias go all the way through the board? It is extremely helpful to have the DSP pins available through all layers. This will increase the layout difficulty and allow visibility into all possible pins on the DSP, which can be a useful for debug.
□	How much general visibility is there on the board? If space allows it, the more signals and pins that are accessible, the easier it is to debug. One common consideration is adding hooks for a logic analyzer on the EMIF bus. This can help with any timing issues that might come up during development.

4.2 Debug Checklist

The Debug Checklist was put together by Texas Instruments application and field support staff as a guide to considerations made during the debug phase of development. Use this check list to keep track of considerations you make during the debug phase of development.

*(a) **Level 1 − Check general operation of DSP, and JTAG connection for development and debugging***

The goal of Level 1 is to determine whether the DSP is functional. This is done by first checking the debug environment, JTAG chain, and by loading and running programs in the DSP memory. If the device is able to pass all of the functions laid out in level 1, then at minimum the device is functional. If there are issues, they likely are with the components outside the DSP.

□	Check general DSP environment. Here, you should check things such as power sequencing and reset circuitry. Is the board being released from reset correctly? This will make sure the DSP is given the chance to start running properly.
□	Start DSP in NO boot mode. Starting the DSP in NO boot mode will allow the device to boot up without having to wait for a host or relying on external memory to be set up correctly. This then allows you to test the voltage levels for both I/O and core against the recommendations in the data sheet.
□	Is the CLKOUT frequency correct? Now that the device is up and running with correct voltages tested from a NO boot power on, check CLKOUT4/6 to make sure the PLL and CLKMODE[3:0] pins are configured properly. This will tell us the exact frequency at which the DSP core is running. It should be noted that in order for CLKOUT4 and CLKOUT6 to be used, EMIFA must be clocked.
□	Does Code Composer Studio run with a simulator (only)? Set up Code Composer Studio to run with a 64x simulator. After starting Code Composer Studio, run a small program in the simulator. If Code Composer Studio is able to run with a simulator, then the installation is functional, and further issues may be emulation-driver related. For example, if Code Composer Studio is able to run a program in the simulator, but is unable to start up with a card or EVM/TEB, then there is a high probability that the emulator driver may be incorrect or incorrectly configured. If that is the case, look at the emulator/JTAG portions of this checklist.
□	Is the scan chain length set correctly? If the scan chain length is not detected properly on your board, Code Composer Studio will not correctly recognize power to the DSP. If it is a multiprocessor board, a scan chain test should return the correct number of devices.
	Check the JTAG scan chain. If you are unable to get Code Composer Studio to run with an emulator, the utility xdsprobe.exe is available to troubleshoot. The xdsprobe.exe utility is available with TI emulators. If there are multiple devices in the chain, the utility should correctly identify all of them. The results of the scan path test should be verified with the setup in the Code Composer Studio Setup Tool. The following is a link for this key information. Using xdsprobe With XDS510 and XDS560 (SPRA758)

□	Does Code Composer Studio run with emulator? Assuming now that Code Composer Studio is working correctly with the simulator, determine if full emulation via JTAG can be achieved. Having emulation functional gives us a more robust development environment with a larger set of peripherals to use and more debugging tools. First, make sure the appropriate drivers for the emulator are loaded into the Code Composer Studio Setup Tool. Then check if the memory map for the specific device is configured upon Code Composer Studio initialization (usually through a startup GEL file in the Code Composer Studio Setup Tool).
□	Can you load/run hello world? Under the tutorials directory of the Code Composer Studio installation, there are two “Hello World” examples. The first one does not use DSP/BIOS and does a simple write to the standard output window using standard C I/O. The second one uses DSP/BIOS and utilizes the LOG object to write to the message log. Verify that both of these example programs are functioning correctly. If they are not, is it a memory verification error? If so, check the memory map. After this, the DSP is working. Now, check the peripherals and all external components.
□	Are the EMIF Clocks set up properly? To interface correctly with external memory like SDRAM, check the specifications for your memory’s speed, then set the clock to the EMIF. Three options exist for the EMIF clock: CPU/4, CPU/6, or external ECLKIN. Check clocks with a scope for proper frequency.
□	Are the CE spaces configured correctly? Using the EMIF control registers for each CE space, make sure that each space is configured for the appropriate form of external memory. *Important note:* If booting from ROM, the ROM device needs to be on CE1, since the on-chip bootloader automatically looks there to start a ROM boot.
□	Do EMIF timings match the data sheets? The data sheets for both the DSP and the external memory device should have timing diagrams. Check the timings from the point of view of both the DSP and external memory, and make sure the signals match their respective data sheets. An IBIS model can also be used to examine timings.
□	Are holds used? If using asynchronous RAM, make sure /HOLD and /HOLDA are used properly. If NOT using asynchronous RAM, make sure the NOHOLD bit in GBLCTL register is set to 0.
□	Is the AARDY pin used? If using asynchronous RAM, make sure AARDY (EMIFA) is being used appropriately. If NOT using asynchronous RAM, the pins are pulled up internally.
□	Are the HDS/HAS signals correctly configured to access the DSP? If using the HPI, then you have a choice of configurations. Examples of timing diagrams for when $\overline{\text{HAS}}$ is used or unused (tied high) are in *TMS320C6000 DSP Peripherals Overview Reference Guide* (SPRU190). Also, there is a choice on how to assert /HSTROBE using both, one, or none of the $\overline{\text{HDS1/2}}$ in combination with $\overline{\text{HCS}}$. The gate logic for these pins should be checked in the reference guide as well.

(b) *Level 2 − External Memory − Read/Write from/to external memory*

Follow the steps below:

1. Load a small program into internal memory that writes to external memory and then verifies all external memory.
2. Have the host program write known values into memory using HPI. Each value should be unique. Have the DSP read back values via JTAG.
3. Have the DSP program write known values into memory. Each value should be unique. The DSP sends an interrupt to the host. The host uses HPI access to read those values back while the DSP is in reset or in a NOP loop.
4. Have the host program write known values into memory. Each value should be unique. Have host generate an interrupt to the DSP. The DSP then checks values.

NOTE: If no HPI is being used in the system, then simply use the DSP and JTAG to read and write memory values.

This not only verifies operation of any external memory that might be used in the system, it also verifies operation of the HPI.

(c) *Level 3 − Boot and Run code*

Now that we have verified operation of both the DSP and possibly the HPI, you can add the boot mode you desire. Check the Bootloading Guide, section 2.7, for details on different boot configurations. Whichever boot process is being used, write a simple test program to make sure you can get the DSP from power-on, to reset, to running the code you desire it to run. The program should do some/all of the following:

1. Toggle an LED(s).
2. Use EDMA to do some data transfer.
3. Generate an interrupt from EDMA transfer.
4. Send a sine wave out through McBSP.
5. Toggle some GPIO pins.
6. Run POST (power-on self test).
7. Run checks on any peripherals your system will use.

5 Summary

Using the information provided in this document, along with documentation that is pointed out for each step of the design process, a DSP designer will be able to make more knowledgeable decisions concerning their design.

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.

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

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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