TL16PC564B Application Notes

Catalog Interface Products

ABSTRACT

This document is a summary of information for using the TL16PC564B device. This information includes crystal and clock considerations, I/O pin information, host CPU/UART interface information, CCR mapping and host/subsystem memory map, host/subsystem arbitration, serial bypass mode information, and other miscellaneous information.

Contents

1 TL16PC564B Introduction

- As described in the data sheet (literature number SLLS225), the TL16PC564B is designed to provide all the functions necessary for a PCMCIA UART subsystem interface. The interface provides a serial-to-parallel conversion for data to and from a modem CODEC/DSP function to a PCMCIA parallel data port format. A host CPU can read the status of UART via the PCMCIA host controller at any point of operation.

  There is an attribute memory consisting of a 256-byte card information structure (CIS) and 8-byte card configuration register (CCR). They are both available to host CPU and subsystem (modem). They both are implemented by using a dual-port RAM. The attribute memory is used in place of the EEPROM, which is normally used for CIS. At power-up, attribute memory is initialized by the subsystem (DSP).

  After power-up reset, the PCMCIA UART is always default to be a memory card according to the PCMCIA spec. All the I/O (UART) operation is disabled. Then the subsystem starts initialization, loads up the CIS into the attribute memory. During this period, IREQZ remains low to indicate to the host CPU that the card is busy. After subsystem finishes the initialization of attribute memory, subsystem sets bit 4 of subsystem control register (SCR) to indicate to the host that the memory card is ready. Bit 5 is then reset to 0, changing the PCMCIA Card configuration from a memory card to I/O card, which also enables I/O (UART) operation. IREQZ now functions as the host CPU interrupt request line.

  There is a unique feature in this part. It is implemented with a subsystem selectable serial bypass mode, which allows the subsystem to bypass the serial portion of the UART and write directly to the receive FIFO and read directly from the transmitter FIFO.

- The subsystem is needed to program the PGMCLK register in order to get the correct baud rate for the COM port. The reset default value of PGMCLK register is 00, which means no UART clock will be provided to the core logic of the UART COM port. Since the PGMCLK register is only accessible from the subsystem, the subsystem must be in place to use the UART COM port.
- Briefly, the subsystem has to be responsible for loading the CIS information right after reset during the initialization. The read/write operation of the subsystem can be used to access the

CIS memory. The normal subsystem would be a TI DSP. The subsystem read can be used to read back the data stored in the CIS for verification.

- The TL16PC564B data sheet does not cover much about the UART operation. Refer to TL16C550C data sheet (literature number SLLS177) for a more detailed description of UART operations, baud rate calculations, and the programmable baud generator.
- Unfortunately, we do not have an evaluation kit or sample CIS code to help you with the development, because this part was a custom part.

2 Crystal and Clock Consideration

Using the oscillator is a better choice, this will eliminate the tuning circuit components (for space considerations on PCMCIA cards) but at an increased cost. For crystal/oscillator selection, please contact a vendor (e.g., Digitkey at **http://www.digikey.com** or call 1−800−344−4539 for a free catalog). A vendor can supply crystals or oscillators that are acceptable (e.g., the Connor-Winfield surface-mount crystal and oscillator).

For clock/crystal frequency considerations, please refer to the description in the TL16PC564B data sheet (*attribute memory arbitration*) and *subsystem PGMCLK register/divide-by-n circuit*). There are two clocks needed:

1. ARBCLKI, which is the arbitration clock input, is used to generate the arbitration clock to arbitrate host CPU/subsystem simultaneous access to the same attribute memory location. The actual arbitration clock used for internal arbitration circuitry can be monitored at the device pin ARBCLKO (arbitration clock output). The arbitration clock requirement:
   a. How to determine the ARBCLKO Min Cycle Period (Max Freq)

DEVICE SUPPLY VOLTAGE	ARBCLKO MIN CYCLE PERIOD(MAX FREQ)
5 V	14 ns (71 MHz)
3.3 V	26 ns (38.5 MHz)

   b. How to determine the ARBCLKO Max Cycle Period (Min Freq) If N is the shortest of the host CPU or subsystem access cycle time to the attribute memory, then the upper cycle period limit of ARBCLKO is N/6. For example, if N = 240 ns, then ARBCLKO = 240/6 = 40 ns. So, ARBCLKO max−cycle = 40 ns <=> ARBCLKO min frequency = 25 MHz

   − The N could be the shortest of tc2, tc3, and its similar parameters for the subsystem access to attribute memory. Take a look at Figure 8 in the data sheet.

   c. Assume 5 V is used as supply voltage and N = 240 ns as in the previous example, the required ARBCLKO frequency is (for example): 25 MHz ≤ ARBCLKO ≤ 71 MHz We recommend using a frequency not too close to the lower limit to avoid arbitration problems caused by host/subsystem performance variation. Of course, using a frequency close to higher limit is better.

   d. Once the ARBCLKO frequency range is decided, four programmable divider values of 1, 2, 4, 8 can be selected to make sure the relationship is met between ARBCLKO and ARBCLKI: ARBCLKO freq = ARBCLKI freq / divider The divider can be programmed by driving the ARBPGM1 and ARBPGM0 pins to four possible combinations. Please refer to data sheet for a more detailed description.

2. XIN crystal clock is used for the UART. The subsystem is **needed** to be able to program the PGMCLK register (which is only accessible by subsystem) to the desired divider value

(do not confuse this with the ARBCLKO divider) in order to get the desired input UART clock frequency. Please refer to *Subsystem PGMCLK Register/divide−by−n circuit* and Figure 1. *XIN Clock Timing Waveforms* in the data sheet for a detailed description.

Example

A very common UART input clock frequency is 1.8432 MHz, which is used as the default frequency to set up the COM port baud rate by Microsoft Windows. If the desired end application's required baud rate is less than or equal to 115 Kbit/s, this frequency is acceptable. Assuming 1.8432 MHz is acceptable, if the input XIN crystal frequency is 32 MHz, now the divider value can be determined by following the instructions in the PGMCLK value vs. divider table in the data sheet.

NOTE: A user can use the exact multiple of 1.8432 MHz as the input frequency to the UART such that the same number of multiple of the baud rate is achieved with the same UART internal divisor value which is set by the Windows COM port driver (There is no assurance that this will always work.). This does not imply that a custom host driver is not needed.

3 I/O Pin Information FAQ

Q: Which inputs are hysteresis inputs and which are non-hysteresis inputs?

A: There is only one hysteresis input on TL16PC564B:
the RESET pin (67) features standard CMOS-compatible input buffer with hysteresis.

Q: Which inputs are fail safe?

A: The input portion of these two bi-directional buses, SAD0−SAD7 and HD0−HD7, feature failsafe CMOS-compatible input buffers.

Q: How about all the other inputs, what is their buffer type?

A: All the other inputs (other than RESET, SAD0−SAD7, and HD0−HD7) have standard CMOS-compatible input buffers.

Q: Which outputs are fail safe?

A: The RSTZ pin (11) and the STSCHGZ pin (74) feature failsafe open-drain CMOS output buffers with the following difference: RSTZ has 8-mA buffer and STSCHGZ has 4-mA buffer.

Q: Which outputs are open-drain?

A: RSTZ and STSCHGZ as described in the previous question. OUT1Z and OUT2Z feature standard open-drain output buffers, 4 mA. There are clamping (protection) diodes to the path of both V_{CC} and GND for OUT1Z and OUT2Z pins.

Q: How about all the other outputs, what is their buffer type?

A: All the other outputs other than RSTZ, STSCHGZ, OUT1Z, and OUT2Z feature standard 3-state CMOS output buffers with 4 mA sink/4m A source, except the NANDOUT pin (12), which has the same buffer type above with 2 mA sink/2 mA source driving capability.

Q: The data sheet says that for low voltage operation (3.3-V supply), the non-hysteresis inputs meet TTL levels (0.8 V low, 2.0 V high). Do these non-hysteresis inputs also meet TTL levels when the chip is in standard voltage mode (5-V supply)?

A: No. They are CMOS-compatible I/O when the supply voltage is 5 V.

Q: We are not using a multiplexed subsystem address/data bus — we are separating the 9-bit address and 8-bit data buses. We are also using Intel mode. We believe, therefore, that the ALE signal is not required, so the ALE input pin (26) should just be tied high (V_{CC}). Is this correct?

A: Yes, tie ALE to V_{CC}. However, this is extremely important, please tie the SSAB pin (3) to V_{CC}. This pin is used to select between a multiplexed address/data bus subsystem interface (SSAB=0) and a subsystem interface with separate address and data buses (SSAB=1). It also has an internal pull-down resistor. This should really be the pin you need to worry about.

Q: When the chip is reset, could you please advise what state its output and I/O pins go to?

A: Below is the summary:

- For bus, SAD0−7 and HD0−7, as usual, avoid bus contentions, they are power-up default to input mode.
- Since these four pins have open-drain buffer, they power-up default to Hi-Z:
  RSTZ, STSCHGZ, OUT1Z and OUT2Z
- UART section outputs:
  SOUT (H), RTSZ (H), DTRZ (H), BAUDOUTZ (H), DIVNCLKO (H)
- Other outputs:
  INPACKZ (H), IRQ (L), IREQZ (L) doubles as READY/BUSY during power-up initialization. During reset, it is low to indicate the memory card is busy.

Q: Can the XIN pin (42) be the same as the arbitration clock the ARBCLKI pin (5), ie the pins get connected together?

A: Please refer to the *Crystal and Clock Consideration* section of this application note for a detailed description.

4 Host CPU/UART Interface FAQ

Q: What is the purpose of defining the COM port in the subsystem control register (address 110h, bits 0−1)? From my experiments, this setting has no affect on the COM address in any way. Further more, this register is not changed by the host PC, regardless of the address configured by the host PC.

A: There are two ways to access/address the UART from the host side: Addressing Mode and Non-addressing Mode. They are described in the *host CPU/UART interface* section of the data sheet. Basically, these two modes are selected by setting or disabling bit 3 of the subsystem control register. The power-up default is non-addressing mode, which is also the mode after a valid reset.

- Addressing Mode Host CPU address bits HA9, HA7, HA6, HA5, and HA3 are combined with conditional derivatives of HA4 and HA8 to select the UART (HA4 and HA8 are used to select COM ports 1−4 based on settings in the subsystem control register). In this mode, setting the subsystem control register bit s 1 and 0 is the only way to control/select which COM port address you want to use. Host or CIS do not have control over the COM port setting. Furthermore, the subsystem is the only one that is allowed to access/update the subsystem control register (at address 0x110h in subsystem memory map). With the setting in SCR[1,0], the subsystem only allows a host to access the UART

in one of the four conventional COM port address: COM1 (0x3F8), COM2 (0x2F8), COM3 (0x3E8), COM4 (0x2E8). There is some address comparison logic inside the chip. Whenever the host tries to access the UART, besides other qualifying condition, the related logic compares the address presented on HA0 − HA9. If that address matches the COM port setting in SCR[1,0], the access is treated valid and would go through. Otherwise, the access is denied.

- Non-addressing mode To use this mode, subsystem control register bit 3 must remain clear. In this mode, UART access is done via the logic level of CE1Z/CE2Z and REGZ. CE1Z and CE2Z are combined such that either of these two signals in combination with REGZ enable the UART in the event that these signals are present. The COM port setting in SCR[1,0] has no effect in this mode.

5 CCR Mapping and Host/Subsystem Memory Map

Q: Do the card configuration registers (CCRs) require setting up by the subsystem processor or are they automatically set up/ written to by the host PC PCMCIA driver?

A: Subsystem is responsible for setting up all the CIS and CCR after power-up reset. This needs to be done before you could turn the memory card to the I/O card.

Hardware-wise, CCR0−CCR7 are not part of the dual port RAM, however they are implemented with D-FlipFlop without reset, which means they also power-up to a random value after reset. The subsystem can then upload the setting/configuration data to them. After that, the host can modify it. Both host and subsystem have access to CCR0−CCR7. There is an arbitration logic inside to avoid race condition.

Q: The PCMCIA Specification Release 2.0 defines the following:

Configuration Option Register	Address Offset 0
Card Status Register	Address Offset 2
Pin Replacement Register	Address Offset 4
Socket and Copy Register	Address Offset 6

What is the exact mapping of the TL16PC564B's card configuration registers CCR0−CCR7 onto the PCMCIA PC card standard configuration registers?

A: To be able to answer this question, we need to look at several aspects of the issue:

1. PCMCIA specification issue:
   First of all, PC564B is PCMCIA Specification 2.01 compatible, therefore, we need to look at 2.01, not the 2.0 specification. There is a specification change from 2.0 to 2.01 to correct a lot of typographical errors in 2.0.
2. Some important CCRs are defined in the PCMCIA Specification 2.01. From the book *The PCMCIA Developer's Guide* written by Michael T. Mori (ISBN#0-9640342-0-4), there are only four CCRs defined here:
   a. COR (Configuration Option Register) — needed for all I/O cards.
   b. CCSR (Card Configuration and Status Register) — Optional, but must be implemented if the PC card supports audio (on BVD2). You might not need this, since TL16PC564B does not support audio and is intended for I/O operation only. There is one bit, PwrDwn, which is only valid if the PC card supports power-down mode, however, TL16PC564B does not support power-down mode.

c. PRR (Pin Replacement Register) — Optional, this needs to be implemented only if any of the four memory status bits need to be read during I/O mode. During I/O operation, TL16PC564B is solely an I/O card and all these bits are meaningless.

d. SCR (Socket and Copy Register) — Optional, provides support in the case where multiple PC cards with the same I/O address co-exist in the same system. You can implement it if you want.

3. CIS and CCRs host/subsystem memory map All the addresses described in the data sheet are in decimal format. Table 1 lists the decimal-to-hex mapping that might be helpful. With Table 1, you could map the registers with address offsets 0, 2, 4, 6 to CCR0, 1, 2, 3 respectively. Of course, you can only treat them as flip flops, since they do not reflect any status of the hardware automatically (they are just storage space). You would have to let your subsystem/host load/update the information at these addresses.

Table 1. Host CPU Attribute Memory Space

REGISTER NAME	DECIMAL ADDRESS	HEX ADDRESS
CIS	0 − 255	0x0 − 0x0FF
CCR0	256	0x100
CCR1	257	0x101
CCR2	258	0x102
CCR3	259	0x103
CCR4	260	0x104
CCR5	261	0x105
CCR6	262	0x106
CCR7	263	0x107

NOTE 1: Since the addressing is based on HA9 − HA1 (HA0=0), this meets the PCMCIA specification definition: the attribute memory and CCRs are 8-bit registers at even byte addresses.

Table 2. Subsystem Memory Space

CONTROL SPACE	DECIMAL ADDRESS	HEX ADDRESS
Subsystem Control Register	272	0x110
PGMCLK Register	288	0x120
UART SPACE		
MCR bit5 (WR Only)	304	0x130
DLL (RD Only)	304	0x130
IER (RD Only)	305	0x131
FCR (RD Only)	306	0x132
LCR (RD Only)	307	0x133
MCR (RD Only)	308	0x134
LSR (RD Only)	309	0x135
MSR (RD Only)	310	0x136
DLM (RD Only)	311	0x137
TFIFO (RD Only)†	320	0x140
RFIFO (WR Only)†	320	0x140

† They are accessible only when serial bypass mode is on.

6 Brief Summary of Host/Subsystem Arbitration

- Principle of Arbitration:

  The arbitration control circuitry synchronizes the asynchronous accesses of the host CPU and subsystem to the DPRAM and CCR and controls the access based on the pending CPU and subsystem attribute memory operation.

- Arbitration Summary:
  - Host RD/Subsystem WR or Host WR/Subsystem RD to same CIS location (AMATCH=1). In either case, RD always wins the arbitration. RD is processed first to prevent the data already in that location before a new WR to the same location.
  - Host RD/Subsystem RD to same CIS location (AMATCH=1). Host RD wins the arbitration.
  - Host WR/Subsystem WR to different CIS location (AMATCH=0). Arbitration does not affect either. Both WRs are processed concurrently.
  - Host WR/Subsystem WR to the same CIS location (AMATCH=1). When both the host CPU and subsystem are performing simultaneous write operations to the same address, the host CPU is allowed to write and the subsystem write is ignored.

7 More About Serial Bypass Mode

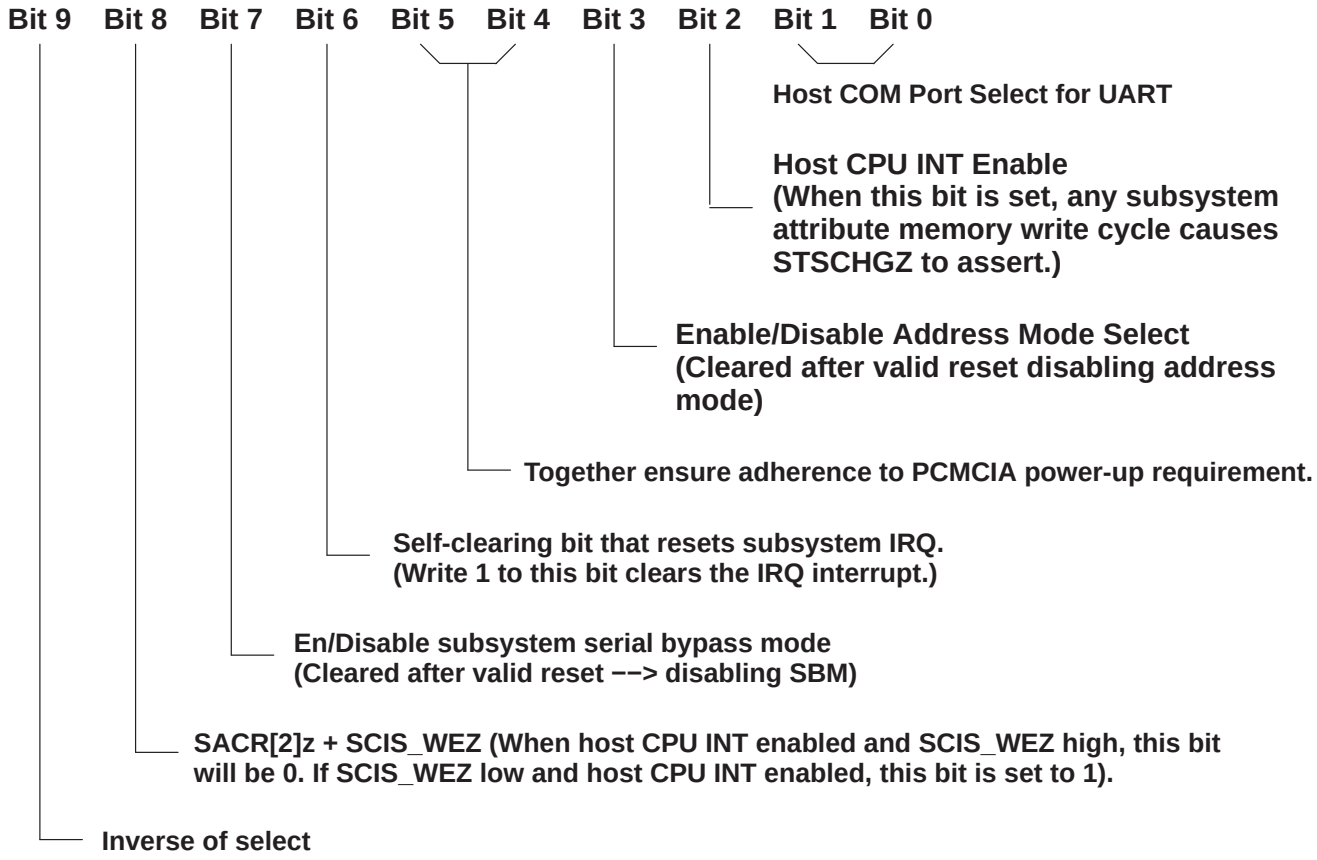

Figure 1. Subsystem Control Register (SACR) Power-Up Value: 0x322 or 0x22 (SCR Only 8-Bit)

NOTE: The SACR[2], according to the definition, is called Host CPU INT Enable. Actually this bit has no control over the S2HIREQZ interrupt line, which is the interrupt from the UART. Regardless of whether SACR[2] is set or not, S2HIREQZ is always enabled as long as certain interrupts in the UART's IER register are enabled. More precisely, SACR[2] should be claimed as host CPU interrupt enable for subsystem write to attribute memory. THE STSCHGZ acts like the subsystem to host interrupts for subsystem writes to attribute memory.

PCMCIA Power-Up Requirement

1. Always memory card after power-up
2. I/O (UART) operation disabled
3. IREQZ is low, which doubles as the host CPU READY/BUSYZ line, indicating card is busy (power-up and reset condition).
4. Wait until the subsystem finishes the initialization of attribute memory. Once done, the subsystem sets bit 4 to indicate to the host that the memory card is ready. Now bit 5, bit 4 = 11 and IREQZ is high −> READY.
5. Reset bit 5 to 0, changing the configuration from a memory card to I/O card, enabling I/O (UART) operation. IREQZ now functions as the host CPU interrupt request line.

Power-Up Meaning for SACR(0:9) − Value 0x322:

1. Default select COM2 −> bit 1, bit 0 = 10.

To meet PCMCIA power-up requirements, the following is needed:

2. Disable Host INTRPT request (memory card not ready, IREQZ is READY/BUSYZ line) −> bit 2 = 0.
3. Disable I/O (UART) operation and disable address mode −> bit 3 = 0.
4. PCMCIA memory card power-up requirement −> bit 5, bit 4 = 10.
5. Power-up/reset, also memory card, and clear the IRQ to subsystem. −> bit 6 = 0.
6. Reset, disabling the serial bypass mode, −> bit 7 = 0.

After Power-Up, Subsystem Write SACR(0:7) − Value 0x1A

1. Use default select COM2 −> bit 1, bit 0 = 10.
2. Disable host INTRPT request −> bit 2 = 0.
3. Enable I/O (UART) operation (clear the BUSY signal) and enable address mode −> bit 3 = 0 −> 1.
4. PCMCIA card configuration changed from memory to I/O card −> bit 5, bit 4 = 10 −> 01.
5. No change to power-up/reset −> bit 6 = 0.
6. The serial bypass mode remains disabled, −> bit 7 = 0.
7. SACR[9] = SACR[8] = 1.

If the serial bypass mode is desired, SCR[7] can be set to 1 to enable the feature. All the other bit settings in the SACR should remain the same.

8 Miscellaneous FAQs

Q: Would TL16PC564B run with Win95 Plug-and-Play without writing a special driver? We are trying to implement a basic serial interface fairly quickly. We would like to see that Win95 could cope with it properly if we set up the CIS and CCR registers correctly. The core PnP functionality concerns us.

A: First of all, please do not confuse this with the Windows' normal Plug-N-Play utility, that is done by the Plug-N-Play add-on card controller and Plug-N-Play BIOS, which has nothing to do with the PCMCIA. Our product group also has a Plug-N-Play controller product for I/O and memory add-on cards. You do not need that for PCMCIA. PCMCIA has its own card insertion and detection protocol.

For Windows auto configuration with this device, We do not have the detailed information to provide, since this device was originally designed to be a custom part for customers who did write their own custom driver. However, based on some customer feedback, they did say that they have been successful in using the Windows 95 PCMCIA host driver. Basically, to configure the device, you need to power-up and load the CIS and CCR information from the subsystem, then follow the description in the *subsystem control register* section of the data sheet and some of our application notes to turn the memory card to an I/O card when you are done with the CIS and CCR upload.

Q: How do interrupts work in the serial bypass mode (for both the host and subsystem)?

A: As we described in the data sheet, serial bypass mode is implemented to allow a high-throughput path to/from the host CPU. When this mode is enabled and the SCR[7] is set to 1, the serial portion of the UART is bypassed and the subsystem has direct parallel access to the receiver FIFO and transmitter FIFO. In short, the subsystem can: 1) Write only to the UART's receiver FIFO (at address 140h) and 2) Read only from the UART's transmitter FIFO (at address 140h).

At this time, all host CPU interrupts operate normally except for receiver parity, framing, and breaking interrupts, since all of these events are only meaningful when the serial communication is in place. For example, if you want to send more than 16 bytes of data from subsystem to host, you can set the receiver trigger level at 16, also enable the receiver data available interrupt. Right after you dumped 16 bytes of data in the UART's receiver FIFO, the host CPU would be interrupted via the IREQZ pin (88). You can find out which type of interrupt is pending by reading the UART's IIR register as normal. The previous scenario assumes the host is not pulling the UART receiver FIFO at the same time of the subsystem write.

As far as the subsystem, there will not be any interrupt indication to it except the host CPU write to attribute memory event.

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