VT-142x Family

DSP PMC Module User's Manual

For All 1420 Series DSP PMCs

Evergreen Group Document Number:

560-000-039

Version 2.2

January , 2005

The Evergreen Group Inc

www.evergreengrp.com

.

IMPORTANT NOTICE

The Evergreen Group Inc. (The Evergreen Group) reserves the right to make changes to their products or to discontinue any product or service without notice. We advise customers to obtain the latest version of relevant information to verify, before placing orders, that information they may be relying on is current and complete. We sell all products subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

The Evergreen Group warrants performance of its products to the specifications applicable at the time of sale in accordance with the The Evergreen Group standard warranty. Testing and other quality control techniques are utilized to the extent The Evergreen Group deems necessary to support this warranty. We do not necessarily perform specific testing of all parameters of each device.

CERTAIN APPLICATIONS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). THE EVERGREEN GROUP PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. THE CUSTOMER UNDERSTANDS THAT INCLUSION OF THE EVERGREEN GROUP'S PRODUCTS IN SUCH APPLICATIONS IS FULLY AT THE CUSTOMER'S RISK.

In order to minimize risks associated with the customer's applications, the customer must provide adequate design and operating safeguards to minimize inherent or procedural hazards. The Evergreen Group assumes no liability for applications assistance or customer product design. The Evergreen Group does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of The Evergreen Group covering or relating to any combination, machine, or process in which such products or services might be or are used. The publication of information regarding any third party products or services by The Evergreen Group does not constitute approval, warranty or endorsement thereof by The Evergreen Group.

CompactPCI® is a registered trademark of the PCI Industrial Computers Manufacturers Group.

TI, Code Composer Studio, DSP/BIOS are trademarks of Texas Instruments Corporation.

Windows and Windows NT are registered trademarks of MicroSoft Corporation

VxWorks is a registered trademark of WindRiver Systems, Inc.

Other product and company names herein may be the trademarks of their respective owners.

This document is derived from an original created by Valley Technologies, and is used under license.

Preface

About this manual

We have written this manual for anyone who wants to design systems using the various versions of the VT-142x DSP card, or to use a VT-142x in a hardware / software lab environment for developmental purposes. We assume that the reader has a basic knowledge of computers and digital logic.

This is the document organization:

- *Chapter 1*, Introduction, introduces the VT-142x, including an overview and a description of the VT-142x, its features and specifications.
- *Chapter 2*, Getting Started, provides information you need to get started using your VT-142x. This includes unpacking instructions, a brief description of the board, instructions for initial configuration of the board, and installation instructions.
- *Chapter 3*, Functional Description, describes the VT-142x on a block diagram level, starting with an overview of the VT-142x, followed by a detailed description of several blocks of circuitry.
- *Chapter 4*, I/O Interfaces, provides a description of the various interfaces on the VT-142x. This includes connector locations and pin assignments.
- *Chapter 5*, Programming Considerations, provides basic information useful in programming the VT-142x. This includes memory maps, register definitions, library functions, diagnostic software, and other programming considerations.

Manual Conventions

Throughout this manual, we use the convention of preceding data and address parameters by a character identifying the numeric format. The following table lists the conventions used throughout this document.

Numeric Format Conventions

0x (zero-x)	Specifies a hexadecimal character
0b (zero-b)	Specifies a binary number
No marking	Specifies a decimal number

For example, “11” is the decimal number eleven, while “0x11” is the hexadecimal equivalent of the decimal number 17, and “0b11” is the binary equivalent of the decimal number three. In order to recognize individual bits within a string of hexadecimal characters, we expand the character in-line using parentheses. For example, 0xXXXX_(XX00)000 indicates that the value of each of the least significant fourteen bits is 0 (zero) while the values of the most significant eighteen bits are not important (X=don't care).

A number sign (#) following the name for signals that are level significant denotes that the signal is true or valid when the signal is low. For example, RST# is the low-true PCI reset signal.

A number sign (#) following the name for signals that are edge significant denotes that the actions initiated by that signal occur on high-to-low transition.

In this manual, we use assertion and negation to specify forcing a signal to a particular state. In particular, assertion and assert refer to a signal that is active or true; negation and negate indicate a signal that is inactive or false. We use these terms independently of the voltage level (high or low) that the signals use to represent the corresponding states.

The following table defines the convention used for data and address sizes.

Data and Address Size Conventions

Byte	8 bits numbered 0 through 7, with bit 0 being the least significant.
Half word	16 bits numbered 0 through 15, with bit 0 being the least significant.
Word	32 bits numbered 0 through 31, with bit 0 being the least significant.
Double word	64 bits numbered 0 through 63, with bit 0 being the least significant.

There are several types of notes used in this manual.

Note - A note contains important information about the VT-142x.

Caution - A caution contains information about situations that could result in damage to the VT-142x or loss of data.

Warning - Warnings indicate situations that may result in physical harm to you or the VT-142x.

Acronyms, Abbreviations and Initializations

Appendix C provides a list of acronyms, abbreviations and initializations.

Contents

Figures

Tables

1 Introduction

This chapter introduces the VT-142x, including an overview and a description of the DSP card, its features, and specifications.

1.1 Overview

We have written this manual for anyone who wants to design systems using any of the VT-1420 variations, supply additional capability to an existing VT-142x system, or work with the VT-142x in a lab environment for experimental purposes. We assume that the reader has a basic knowledge of computers and digital logic.

1.2 Model and Part Number Description

Table 1 - Model Number Descriptions

Model #	Processor	I/O Interface	#of Processors
VT-1420	TMS320C6415 or TMS320C6416	Utopia Level II	2
VT-1423		Utopia Level II	1
VT-1426		EMIFB	2
VT-1425		EMIFB	1

Table 2 - Part Number Definitions

VT-____-PMC-____-__-__-_

Processor Speed
- 4 = 400 MHz
- 6 = 600 MHz
- 7 = 720 MHz

Processor
- 15 = TMS320C6415
- 16 = TMS320C6416

Flash Memory Size (per processor)
- F0 = None
- F1 = 1 MB
- F2 = 2 MB (Paged)

SDRAM Memory Size (per processor)
- D0 = None
- D116 = One 16 MB bank
- D216 = Two 16 MB banks
- D132 = One 32 MB bank
- D232 = Two 32 MB banks

Processor Configuration

UTOPIA II Interface
- 1423 = Single Processor
- 1420 = Dual Processor

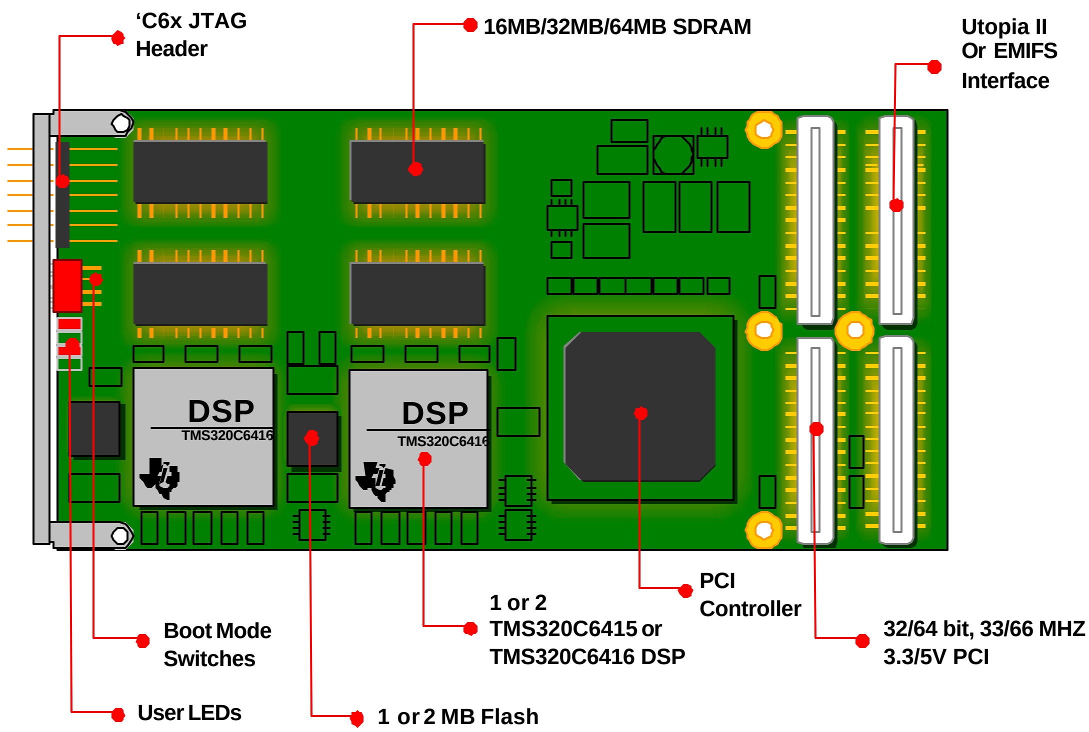
EMIFS Interface
- 1425 = Single Processor
- 1426 = Dual Processor

Note: Some configurations defined by this table may not be available. Please consult The Evergreen Group.

1.3 VT-142x Description

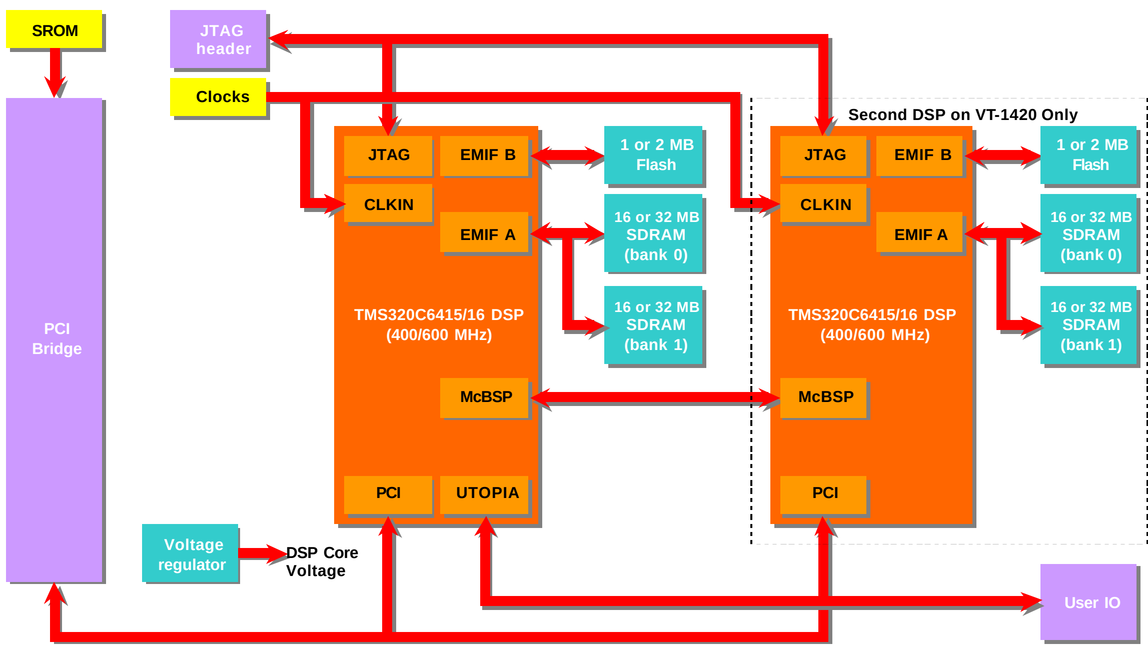
The Evergreen Group's VT-142x is either a single or a dual TMS320C6415/C6416 DSP on an industry standard single width, standard height IEEE 1386.1 PCI Mezzanine Card (PMC). Each DSP performs up to 4800 MIPS. The VT-142x uses an industry standard PCI bus for transfers of data and program.

Figure 1 – VT-142x Physical Layout



The block diagram in Figure 2 illustrates the architecture of the various versions of the VT-1420 family of processor cards. The main blocks are the two Digital Signal Processing (DSP) sub-systems each equipped with a C6400 DSP, external Flash ROM (Read Only Memory) (1 or 2 MB), and Synchronous Dynamic RAM (SDRAM) (16 MB, 32 MB, or 64 MB). In general, the product number VT-142x is used generically throughout this manual to refer to either the VT-1420, VT-1423, VT-1425, and VT-1426 product. The only difference between the different part numbers is the number of DSP devices (1 or 2) and the interface o n connector P14 (Utopia or EMIF interfaces). A part number configuration table is shown in figure xx, below.

Figure 2 – VT-142x and VT-1423 Architecture



Functional areas of the VT-1420 and VT-1423 PMC Module

The VT-142x has a 64-bit, 66 MHz PCI Rev. 2.2 compliant agent device that supports both Initiator and Target transfers . This interface allows the host to perform data and configuration cycles to the VT-142x, and allows the VT-142x to perform data cycles to and from the host or another PCI device.

Finally, the VT-142x provides a JTAG/Emulator port to allow the user to create and debug the target DSP code using the most up-to-date tools provided by Texas Instruments (TI). This port along with the RTDX (Real Time Data eXchange) connection and DSP/BIOS kernel allows the users to debug the code in real time and s ave development time. For a more detailed explanation of the previous terms, refer to the Code Composer Studio User's Guide.

This standard height module fits within a single PCI, CompactPCI, or VME slot. Regardless of your platform, the VT-142x provides you with a modular, low-power solution for your embedded digital signal processing applications.

1.3.1 VT-142x Features

- One or two Texas Instruments C6400 DSP subsystems.
- Each DSP subsystem equipped with:
  - TMS320C6415 or TMS320C6416 DSP at up to 600 MHz
  - Up to 4800 MIPS maximum processing power
  - Multi channel Buffered Serial Port (McBSP) connecting the two DSP chips
  - UTOPIA II Interface (DSPA only) or EMIF Interface, depending on part number
  - Up to 64 MB of SDRAM
  - Up to 2 MB of Flash ROM
  - Red and Green LEDs
  - Easily accessible switch for bootmode selection
- 32/64-bit, 33/66-MHz, 3.3/5V PCI interface
- JTAG Emulator port interface allowing code development and debug with all of the features of TI Code Composer Studio (CCS 2.1 or later).
- "Hot" boot mode change, allows developers to change boot mode settings while the system is powered up and without rebooting
- Single Width, Standard Height IEEE 1386.1 PMC (PCI Mezzanine Card) form factor

1.3.2 VT-142x Specifications

Table 3 – Physical Specifications

Dimensions:	149 mm x 74 mm IEEE P1386.1 Single Width, Standard Height PMC
Storage Temp:	-40 to +85° C
Operation Temp:	0 to +60° C

Table 4 – PMC/CMC Power-Related Specifications

Specification	Calculated Value
+5V Current Peak:	≤ 700 mA
+5V Current Average:	≤ 490 mA
+3.3V Current Peak:	≤ 2000 mA
+3.3V Current Average:	≤ 790 mA
Average Power Dissipated on Side 1	≤ 5.05 W
Ave rage Power Dissipated on Side 2	≤ 0.45 W
Percent of side 1 area, side view, not occupied by components (airflow):	≥ 34%
Percent of side 2 area, side view, not occupied by components (airflow):	≥ 60%

Power values are provided for a VT-142x-PMC-D232-F2-16-6.

2 Getting Started

This chapter provides information you need to get started using your VT-142x. This includes unpacking instructions, a brief description of the board, instructions for initial configuration of the board, and installation instructions.

Table 5 lists the steps you will need to perform before you can use this board, and tells where to find the information you need to perform each step. Be sure to read this entire chapter, including all cautions and warnings, before you begin.

Table 5 – Start-Up Overview

What you need to do...	Refer to...	On page ...
Unpack the hardware.	Section 2.1, Unpacking	7
Configure the VT-142x.	Section 2.2, Configuration	8
Install the VT-142x.	Section 2.4.2, Mounting the VT-142x on a Carrier Board	10
Power up the system.	Section 2.4.3, Applying Power	12
Program the VT-142x as needed for your applications.	Section 5, Programming Considerations	33

2.1 Unpacking

Note - If you notice damage to the shipping carton upon receipt, request that the carrier's agent be present during the unpacking and inspection of the equipment.

Unpack the equipment from the shipping carton. Be careful to remove the VT-142x from the anti-static storage bag using only approved anti-static procedures. Refer to the packing list and verify that all items are present. Save the packing material for storing and reshipping the equipment.

Caution - Avoid touching areas of integrated circuitry. Static dis charge can damage these circuits.

2.2 The VT-142x Board

Figure 3 shows the location of the I/O connectors, headers, and switches on the VT-142x family of PMCs. In addition, it shows the major chips on the module, being the PCI Interface (U3) and the two DSPs (U1, U2).

Since this module is a universal PMC, both the 3.3V and the 5V keying holes are present. This means that it is possible to plug it in any PMC carrier board, as long as the carrier is compliant with the CMC/PMC specifications IEEE 1386 and IEEE 1386.1. In addition, it operates at 66MHz on a 66MHz PCI bus with the on-board PCI bridge providing the rate buffering.

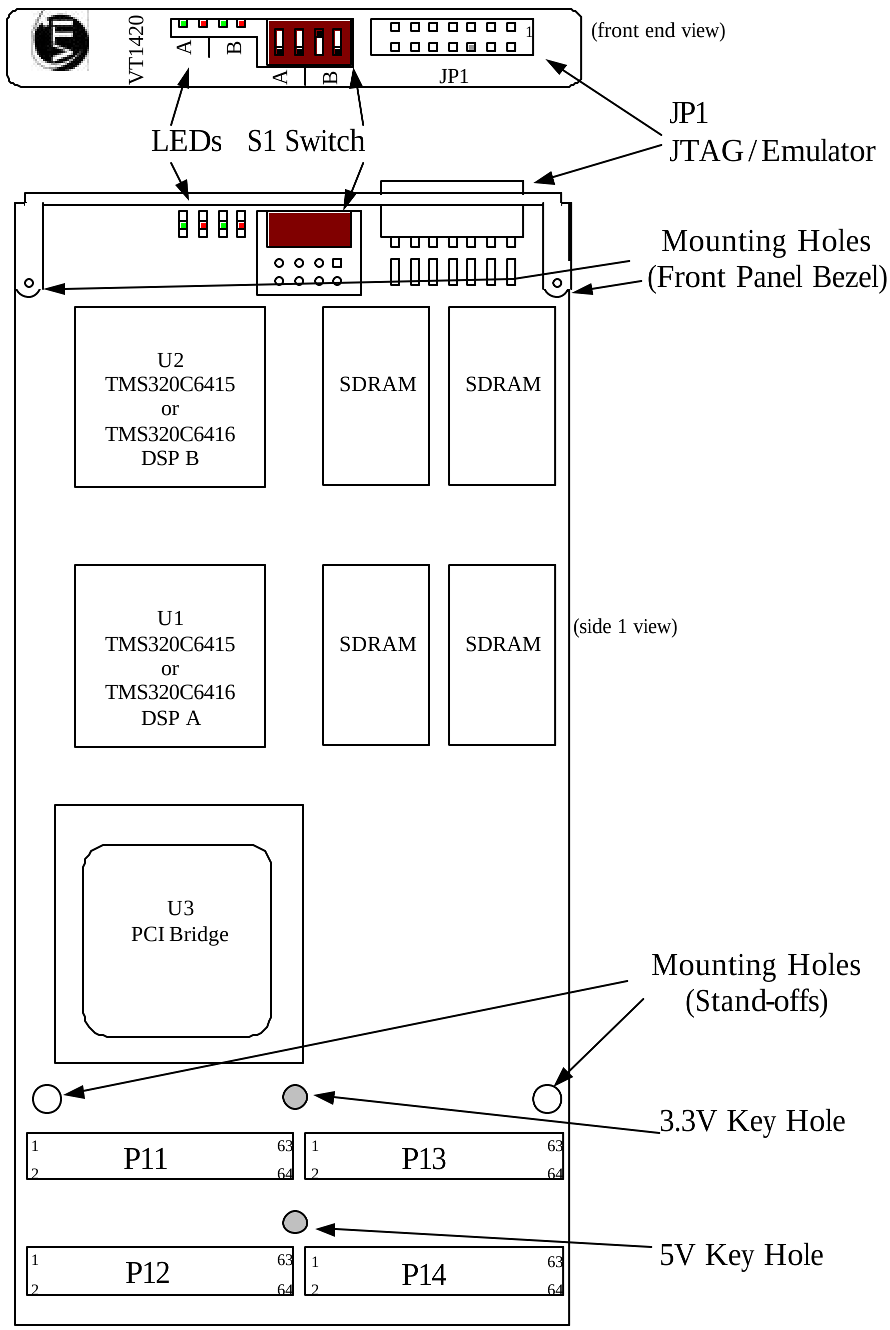
Header JP1 is the JTAG/Emulator header for connection of the C6400 emulator cable. See sections 3.2.1.7 and 4.2.1 for more information about the JTAG emulator interface.

Connectors P11-P13 carry the PCI interface to the PMC carrier board. Connector P14 provides the Utopia or EMIF interface from the first DSP to the carrier board. See sections 3.2.1.9, 3.2.2, and 4.2.2 for more information about the signals on the PMC connectors.

Switch S1 allows you to configure the operating mode of the two DSPs. See sections 3.2.1.5 and 5.8 for more information about boot mode settings.

There are no user-configurable jumpers on the VT-142x.

Figure 3 – Switches, Headers, Connectors, and Mounting Holes



2.3 Configuration

To produce the desired configuration and ensure proper operation of the VT-142x, you only need to verify the BOOT MODE settings on Switch S1. You can leave these settings in the factory default settings (as defined in section 5.8 Setup and Initialization) or set them according to the boot configuration needed. The design of the VT-142x allows the switch settings to be set before or after installing the module onto a carrier card. In fact, the design of the VT-142x allows users to change the boot mode settings after system boot up, as described in sections 3.2.1.5 and 5.8.

2.4 Installation

The following paragraphs discuss installation of the VT-142x onto a carrier board. For Windows driver installation, follow the instructions provided in the distribution ZIP file that is accessible via a web download. The information required to retrieve this ZIP file is provided corresponding with the product shipment. If you did not receive this information, contact The Evergreen Group.

2.4.1 Taking ESD Precautions

We strongly recommend that you use an anti-static wrist strap and a conductive pad when you install the VT-142x. Electronic components can be extremely sensitive to Electrostatic Discharge (ESD). After removing the board from the system or its protective container, place it flat on a grounded, static-free surface with the PMC connectors (P11-P14) and stand-offs facing up. Do not slide the board over any surface.

If an ESD station is not available, you can avoid damage resulting from ESD by wearing an anti-static wrist strap attached to an unpainted metal part of the system chassis.

2.4.2 Mounting the VT-142x on a Carrier Board

You can mount the VT-142x onto any PMC carrier board. We have designed the card to work on PMC carrier boards with 5V or 3.3V signaling. The VT-142x interfaces to the carrier board via the P11-P14 connectors. It also draws +3.3V and +5V power from the carrier board through these connectors.

Refer to Figure 4 and perform the following steps to install the VT-142x on a PMC carrier board. This procedure assumes that you have read the user's manual that came with your PMC carrier board.

1. Attach an ESD strap to your wrist. Attach the other end of the ESD strap to the chassis as a ground. You must have the ESD strap secured to your wrist and to ground throughout the procedure.

2. Perform an operating system shutdown. Turn the AC or DC power off and remove the AC cord or DC power lines from the system. Remove the chassis or system cover as necessary for access to the carrier board.

Caution - Inserting or removing boards with power applied may result in damage to board components.

Warning - Dangerous voltages, capable of causing death, are present in this equipment. Use extreme caution when handling, testing, and adjusting.

3. If you already have the carrier board installed in a card slot, carefully remove it. Lay the carrier board flat, with the backplane connectors facing you and the PMC connectors facing up.

4. Refer to the carrier board user's manual regarding any hardware configuration that may be required to operate the PMC card properly.

Caution - Avoid touching areas of integrated circuitry. Static discharge can damage these circuits.

5. Remove the filler panel from the selected PMC slot in the front panel faceplate of the host carrier board.

6. Holding the VT-142x over the carrier board, align the PMC connectors and the voltage-keying pin.

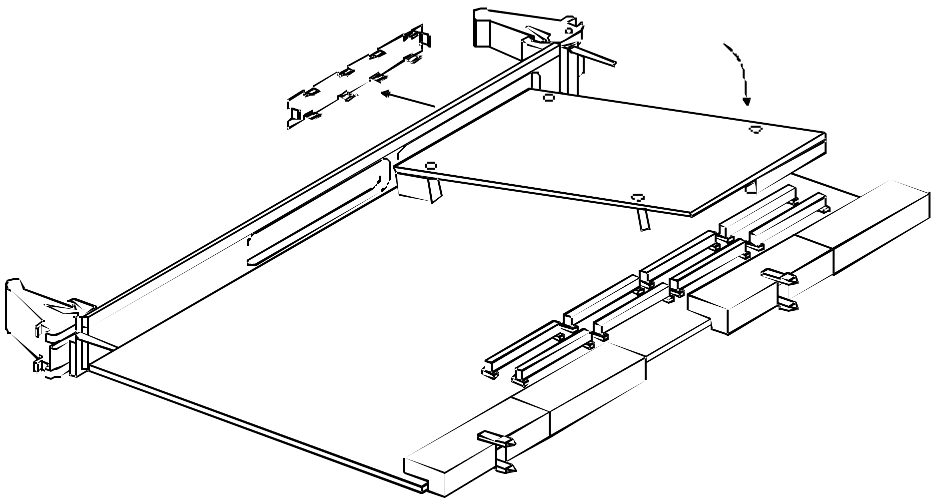
7. Slide the bezel of the VT-142x into the front panel opening from behind and place the VT-142x down on top of the carrier board. When properly aligned, the connectors on the underside of the VT-142x should connect smoothly, using minimal insertion force, with the corresponding connectors on the carrier board.

8. Insert the four short screws, provided with the VT-142x, through the holes on the bottom side of the carrier board and into the VT-142x front bezel and rear standoffs. Tighten the screws.

9. Reinstall the carrier board into its card slot. Be sure you properly seat the board into the backplane connectors, taking care not to damage or bend any connector pins.

10. Replace the chassis or system cover, and reconnect the system to the AC or DC power source.

Figure 4 – VT-142x Placement on PMC Carrier



2.4.3 Applying Power

After you have verified that you have done all of the necessary hardware preparation, that you have made all of the connections correctly, and you have completed the installation, you can power up the system.

3 Functional Description

This chapter describes the VT-142x on a block diagram level, starting with an overview of the VT-142x, followed by a detailed description of several blocks of circuitry.

3.1 General Description

The VT-1420 and VT-1426 contain two TMS320C6415 or two TMS320C6416 DSP subsystems on a single width, standard height, PCI Mezzanine Card (PMC). The VT-1423 and VT-1426 contain a single TMS320C6415 or TMS320C6416 DSP subsystem as indicated in the block diagram in Figure 2 on page 4.

As shown in section 1.3.1, VT-142x Features, the VT-142x offers many standard features that make this module very flexible and suitable for many different signal processing applications, including Voice over IP (VoIP), packet processing, encryption, compression, ATM/SAR and proprietary modems. It is also perfect for desktop development and evaluation of the TMS320C6400 DSPs.

The PCI bus can be used to load code and data to a DSP and its memories. DSP code can be executed from a host controller on the PCI bus. This method of data and program movement is much more efficient than what can be obtained using the JTAG emulator. The VT-142x interfaces to a PCI bus using a non-transparent bridge. This bridge enables the VT-142x to interface to any variant of the PCI bus, including 32-bit/64-bit, 3.3V/5V, and 33MHz/66MHz.

There is an on board Emulator/JTAG connector allowing the user to directly debug DSP code using the TI Code Composer Studio Debug tools. Front panel LEDs and boot mode switches aid in code development. Each DSP has a green and a red LED under General Purpose I/O (GPIO) program control. The VT-142x is designed so that the user can change the boot mode settings of the DSP without having to power down any equipment or reboot the PCI host system. This feature is quite useful to the developer who needs to test his work in more than one boot mode.

3.2 Block Functional Description

The following section refers to the block diagram in Figure 2, showing the VT-142x's general architecture and gives an in depth description to allow the user to efficiently program and use the module.

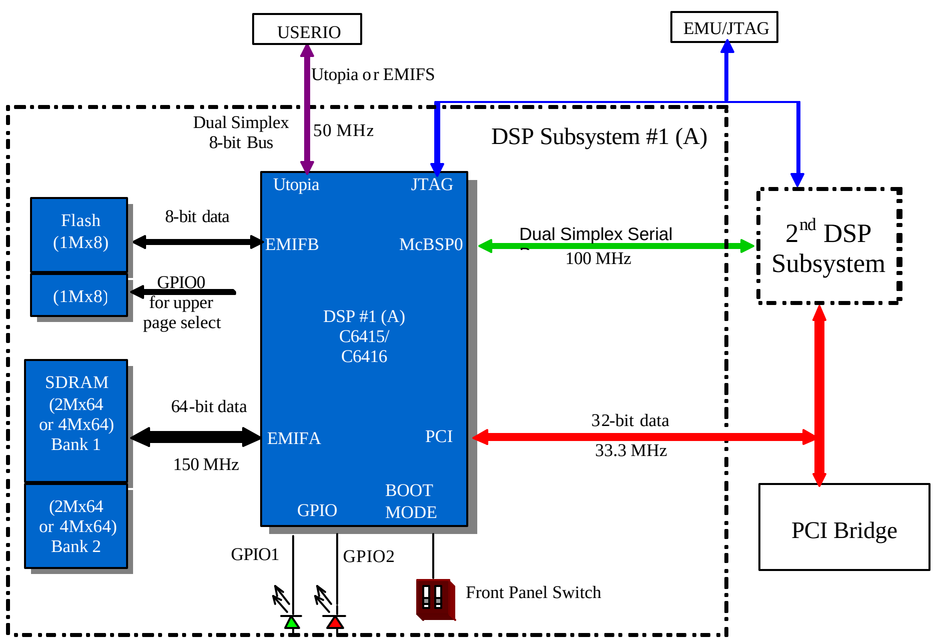
3.2.1 DSP Subsystem

Each DSP subsystem contains a TMS320C6415 or TMS320C6416 processor surrounded by various memory banks as shown in Figure 5. Each DSP subsystem contains up to 2 MB (two 1 MB pages) of Flash ROM and up to 64 MB of SDRAM (two banks each 32 MB). The Flash ROM bank on External Memory Interface B (EMIFB) ships with predefined boot code to place each DSP into a benign state after reset. You can use the Flash ROM for user-defined programs and configuration data for embedded systems. The SDRAM bank on EMIFA can store data and coefficients as well as program sections that cannot fit in the DSP internal 1 MB of RAM.

The VT-142x can be populated with either TMS320C6415 or TMS320C6416 DSPs. In addition, the design of the VT-142x enables the DSP to be configured to run at different core clock rates up to 1GHz. Consult the factory for the latest configuration and in-stock options.

Each DSP subsystem has three interfaces; a PCI Interface, a Multi-Channel Buffered Serial Port (McBSP), and a JTAG/emulator port interface. The PCI host can access each DSP subsystem through the PCI Interface to load programs, access data or service interrupts. DSP Developers use the JTAG/emulator port to debug code. In addition, the UTOPIA level II or EMIF interface from DSP A is routed to the PMC UserIO connector P14.

Figure 5 – DSP Subsystem Details



3.2.1.1 DSP

Each DSP subsystem contains one DSP. The DSP can either be a TMS320C6415 or a TMS320C6416 from Texas Instruments (TI) running up to 720 MHz. The standard VT-142x ships with a TMS320C6416 DSP running at 720 MHz with 1Mbyte of internal Level 2 Cache RAM. This processor is the first from TI's newest and highest performing line of DSPs and is capable of 5780 MIPS while consuming less than 2.5 Watts. In this subsystem, the DSP is configured to operate in little Endian mode only. Refer to the TI documentation in Table 33 for more information about this processor.

3.2.1.2 SDRAM and EMIFA

EMIFA has a 64-bit wide data path that runs the interface clock at one quarter of the core clock rate; 150 MHz for 600MHz boards and 180MHZ for 720 MHz boards. EMIFA connects to one or two banks of 64-bit wide SDRAM. Bank 1 of SDRAM is selected by Chip Enable 0 (CE0) and Bank 2, if present, is selected by Chip Enable 1 (CE1). See the memory map in Table 15. The SDRAM banks can be either 16MB (2Mx64) or 32MB (4Mx64). The factory stocks boards with 1 bank of 32MB SDRAM.

3.2.1.3 Flash and EMIFB

EMIFB has an 8-bit wide data path that also runs the interface clock at one quarter of the core clock rate; 150MHz for 600MHz and 180MHz for 720MHz. EMIFB connects to one bank of 8-bit bottom-boot Flash that is 1MB (1Mx8) or 2MB (2Mx8) in size. The Flash is selected by Chip Enable 1 (CE1) as required by the DSP for booting from Flash. See Table 15 and Table 16 for more information on how the Flash fits into the DSP's memory map. EMIFB only has 20 bits of address which equates to 1Mbyte of addressing. When 2MB of Flash is used, the upper 1MB of Flash is paged by a DSP General Purpose I/O (GPIO). GPIO0 is used to select between the upper and lower pages of the 2MB Flash. After reset, GPIO0 is a logic low ('0") to select the lower page of Flash. This is where boot code should reside. Setting GPIO0 to a logic high ("1") selects the upper page of Flash. The 1MB Flash device is the AM29LV800xB and the 2MB Flash device is the AM29LV160xB, both from AMD.

3.2.1.4 LED Indicators

Each DSP subsystem has a green and a red LED that is visible through the front bezel of the PMC module. These LEDs are under DSP program control. The green LED is illuminated when General Purpose Input/Output 1 (GPIO1) is enabled and set to a logic 0 and the red LED is illuminated when GPIO2 is enabled and set to a logic 0. When GPIO1 and GPIO2 are disabled, for example after reset, these signals default to clock outputs from the DSP. When the GPIO1 and GPIO2 are clock outputs, the LEDs are illuminated but not quite as brightly as when the GPIOs are enabled and set to a logic 0. This immediately enables the user to recognize that power is applied and the board has some level of operation. The software included with this product provides an easy means for users to toggle these LEDs from host functions, embedded C64x library calls, and GEL file functions for Code Composer Studio and compatible emulators.

3.2.1.5 Boot Mode Switches and RESET

Each DSP has two switches associated with boot mode settings. Table 31 illustrates the options available for boot. For each DSP, with both switches in the ON position, the DSP will boot from the PCI host and with both switches in the OFF position, the DSP will boot from its Flash. DSP A and DSP B boot modes are independently selected and do not need to be the same.

The design of the VT-142x allows developers to change the boot mode settings on the 'fly'. The PCI host does not need to be rebooted or power cycled to enable this change. A PCI access to the registers in the PCI bridge will generate a reset to the DSP subsystems. This reset causes the DSP to read the current state of the boot mode settings and boot appropriately. The INTEL 21555 bridge generates a secondary PCI bus reset (S_RST#) to both the PCI (PRST#) and core (RESET#) reset inputs to the DSPs. The host software included with this product provides an easy means to reset and reconfigure both DSPs subsystems via S_RST# under program control from the host.

3.2.1.6 PCI Interface

The PCI interface of the DSP is a 32-bit, 33MHz bus with 3.3V signaling. This interface does not go directly to the host PCI bus but instead is routed as a local PCI bus that is isolated from the host PCI bus via a nontransparent PCI bridge. DSP A, DSP B, and the secondary port of the nontransparent bridge are the three PCI devices on the local PCI bus. There are no connectors or test points on this local PCI bus.

The PCI interfaces of the three devices on the local PCI bus are configured by a controller on the host PCI bus. The software included with this product automatically sets up the Configuration Space Registers of these PCI interfaces and provides an easy means to read and modify all of the settings in the Configuration Space Registers as well as access memory-mapped and I/O space. For configuration space accesses, the VT-142x uses AD30 as IDSEL to select DSP A and AD31 to select DSP B.

3.2.1.7 Emulator/JTAG Header

This header provides a standard 2x7 0.1" connection to the common C6400 emulators. It allows the user to control the DSPs using the Code Composer Studio debug tools to develop and debug code. This header might be either a male or a female header. In the case of a female header, a male-to-male gender changer is provided. For the VT-1423, this header connects to a single DSP and no other JTAG devices. For the VT-142x, this header connects to two DSPs and no other JTAG devices. In either case, the emulator connection is direct to the DSPs with no ESD protection beyond what the DSPs themselves provide. Refer to section 4.2.1 for pin outs and connector information and pay close attention to the warnings.

It is important to follow the manufacturer's instructions for debugging with an emulator. Parallel port emulators require a specific powerup sequence in order to operate properly. Below is the sequence that works with Spectrum Digital's SPI515 emulator and the VT-

142x. In many cases, it is important to have the latest version of Code Composer Studio, CCS V2.1 or later.

1. Start with all components powered down. This includes the PC running Code Composer Studio and emulator, the PCI backplane and VT-142x device under test (DUT), and the emulator itself.

2. Before connecting the emulator to the DUT, be sure to use proper ESD techniques for the reasons explained in section 4.2.1. Connect the emulator to the 2x7 header of the DUT.

3. Power up the emulator.

4. Power up the DUT and PCI backplane controller.

5. Power up and boot the PC running Code Composer Studio. In some lab configurations, this PC might be the PCI backplane controller listed in the previous step.

Proper emulator startup steps are important for reliable operation.

3.2.1.8 McBSP0 Serial Bus

Each DSP has three Multi-Channel Buffered Serial Ports or McBSPs. The first McBSP, McBSP0, is the only one of these three with pins that are dedicated to this McBSP interface. McBSP1 and the Utopia interface share dual-purpose pins. Since the Utopia interface is enabled as the standard default configuration, McBSP1 is then disabled. Similarly McBSP2 is not available because it shares dual purpose pins with an optional serial ROM interface. For more information about McBSPs, refer to Table 33 regarding Texas Instruments documentation for the DSP, especially SPRS164 and SPRU190.

McBSP0 is connected between DSP A and DSP B as show in Figure 2 – VT-142x and VT-1423 Architecture and Figure 5 – DSP Subsystem Details. McBSP0 can be used for interprocessor communication and data transfer. In addition, in a special configuration, McBSP0 can be routed to the UserIO Connector P14. Refer to Table 12, Table 13, and Table 14 for special configuration pin assignments to the UserIO Connector P14 involving McBSPs. In addition pay close attention to the information and warnings in section 4.2.2 regarding the P14 connector.

3.2.1.9 Utopia Level II

The Utopia interface is included on the VT-1420 and the VT-1423 boards. The Utopia interface is a 50 MHz dual simplex bus with 8-bits of data in both transmit and receive directions. Level II Utopia provides addressing in addition to enables to enhance multi-device operation. For more information about Utopia, refer to the Utopia specifications listed in Table 34.

The VT-142x module routes the Utopia interface to the UserIO Connector P14 as illustrated in Table 10. This connection is direct to DSP A and goes only to DSP A. This is connection is a 3.3V only and is not 5V tolerant. Pay attention to the information and warnings in section 4.2.2 regarding the P14 connector.

The VT-142x is compatible with Configuration 7 of the PICMG 2.15 where P14 is defined as user-defined I/O. In addition, the Utopia signals were pinned out on P14 to be compatible with Configuration 3 of this specification. For information on this and other PICMG 2.15 configurations, refer to this PCIMG specification listed in Table 34. Also, refer to section 4.2.2 regarding the P14 connector and play close attention to the information and warnings in this section.

3.2.1.10 EMIF B Interface Option (VT-1425 / 26)

The VT-1425/26 models route the EMIF B interface from DSP A to the UserIO Connector on P14 as illustrated in Table 6.

3.2.1.11 Interrupts, Timer and General Purpose IO

Each DSP subsystem has discrete inputs and outputs, some of which have dedicated functions while others have functions that are user-defined. The following table captures these signals and identifies the intended function. Notice that some of this information is redundant with information found in earlier sections such as LED control.

Table 7 – DSP Discrete I/O and Interrupt Definition

DSP Signal Name	Direction	Signal Name	Description
GP0 (General Purpose I/O 0)	Output	Flash Page	When enabled, this GPIO selects the lower (default) page in Flash when set to a logic 0 and selects the upper page when set to a logic 1. See section 3.2.1.3
GP1	Output	GREEN_LED	When enabled, this GPIO illuminates the GREEN LED when set to a logic 0 and extinguishes it when set to a logic 1. See section 3.2.1.4.
GP2	Output	RED_LED	When enabled, this GPIO illuminates the RED LED when set to a logic 0 and extinguishes it when set to a logic 1. See section 3.2.1.4.
GP7/EXT_INT7 (Shared GPIO and Interrupt input)	Input	DSP_ID	This signal is intended to be used as a general purpose input to identify DSP A versus DSP B. This input will be low (logic 0) for DSP A and high for DSP B.
TINP0 (Timer Input 0)	Input	CONFIG0	This input indicates the DSP clock speed. When equal to 0, DSP is running at 600MHz. When equal to 1, DSP is running
TINP1	Input	CONFIG1	This input indicates the DSP type. When equal to a logic zero, DSP is type 320C6416. When this input is high, DSP is type 320C6415.
GP4/EXT_INT4	Input	EXT_INT4	Intended to be used as an interrupt input, the source of this signal is from the other DSP's TOUT0. See section 3.2.3.
TOUT0	Output	EXT_INT4'	This output can be used to interrupt the other DSP via its GP4/EXT_INT4 input. See section 3.2.3.
GP6/EXT_INT6	Input	INTA#	Intended to be used as an interrupt input, the source of this signal originates from the secondary port of the PCI bridge. (R127 must be populated with 0Ω resistor for DSP B). See section 3.2.3.
GP5/EXT_INT5	Input	EXT_INT5 (User-defined)	Primarily intended as interrupt input, this signal could be used as general purpose I/O. This signal can be routed to P14 when specially configured. See Table 13 for user I/O pin assignments options.
TOUT1	Output	TIMER_OUT1 (User-defined)	This signal is entirely intended for the user to define and can be routed to P14. See Table 13 for user I/O pin assignments options. It is also possible to route this INTD# of the PCI I/F.
TIN2/ TOUT2	Input/ Output	TIMER_IN2 / OUT2 (User-defined)	This signal is entirely intended for the user to define. It is connected to both the input and output pins of timer 2 and can be routed to P14. See Table 13 and Table 14 for user I/O pin assignments options. As an input, it can behave as an interrupt.

3.2.2 PCI Bridge

The INTEL 21555 PCI bridge provides the VT-142x with a completely universal PCI interface to the host processor. The host processor sees a PCI interface that is either 64-bits or 32-bits, 66 MHz or 33 MHz, and 3.3V or 5V PCI signaling. The bridge translates the host PCI bus, in whatever form, to the local 32-bit, 33MHz, 3.3V PCI bus to the DSPs. For example, the VT-142x will operate as a 66MHz device in a 66MHz bus with the bridge providing the rate buffering to the local 33MHz bus. In addition to this PCI host-to-local bus translation, the bridge provides other benefits as well, including:

- Rate buffering between host and DSPs with 256 Byte transaction buffers
- Isolated local PCI transactions between DSPs
- Scratchpad registers for doorbell messages, accessible by both the DSPs and the host processor
- 'Downstream' address translations initiated by a PCI device on the primary port to DSPs on the secondary port
- 'Upstream' address translations initiated by the DSPs on the secondary port to a PCI device on the primary port
- Isolated local PCI reset allows the host to reset the DSPs without rebooting the whole system. This can be extremely useful to the developer who needs to change boot modes.
- Higher PCI performance with less latency and burst rates over 500Mbytes/sec

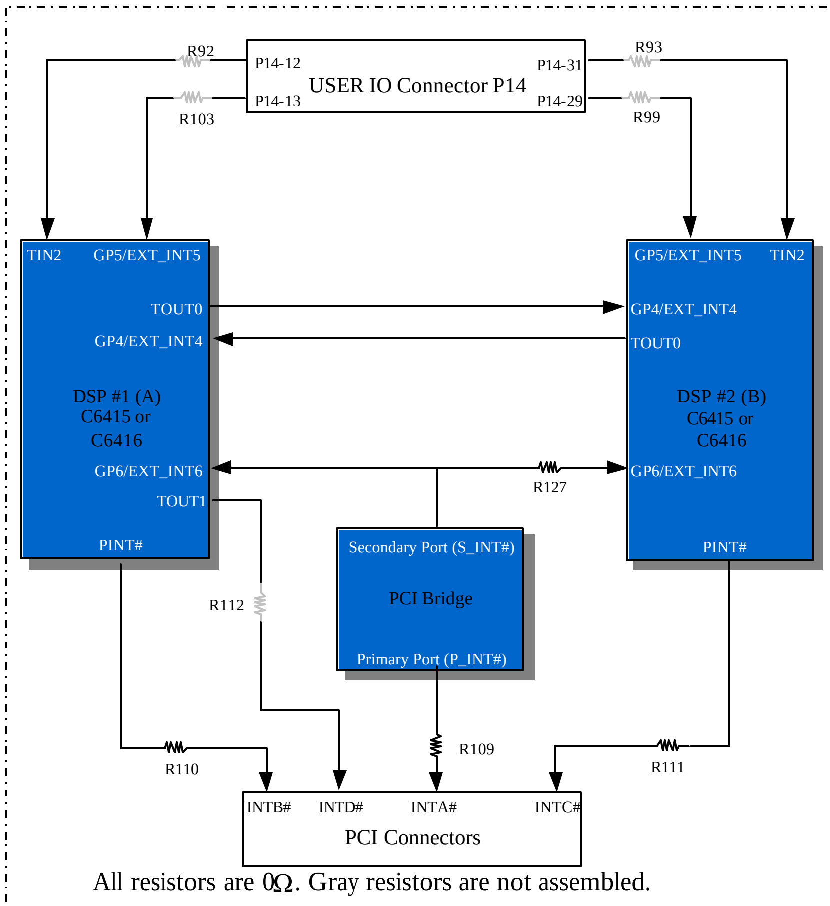
The VT-142x arranges the PCI Bridge so that the Primary Port of the bridge faces the host processor and the Secondary Port of the bridge faces the DSPs on the local PCI bus. The PCI bridge is partially configured via a serial PROM. The software included with this product provides a host utility that can read and program this serial PROM. In addition, the utility also configures the remaining portions of the PCI bridge. For more information about configuring the bridge's two ports, refer to sections 5.3 and 5.4 and the INTEL documentation listed in Table 33.

3.2.3 Interrupt Structure

The VT-142x provides significant interrupt features. As illustrated in Figure 6, each DSP can directly generate an interrupt to the PCI host or indirectly via the PCI bridge, the DSPs can interrupt each other, and each DSP routes discrete I/O and interrupt inputs (see section 3.2.1.11) to the User IO connector P14.

Refer to the DSP and Bridge vendors' documentation (Table 33 – Manufacturers' Documents) for programming information regarding the use of these interrupts.

Figure 6 – Interrupt Structure



3.2.4 Optional Signal and Pin Assignments

The standard configuration for the VT-1420/23 series modules only connects the Utopia bus to P14. In this configuration, the VT-1420/23 is compatible with PICMG 2.15 Configuration 7 and in addition P14 is pin assigned per PICMG 2.15 Configuration 3. However for OEM purchases, the VT-1420/23 family can be configured to connect other signals to P14 via 0Ω jumper resistors. These include the following:

- P14 No Connects (N/C) can be replaced with McBSP0 signals as detailed in Table 13 and Table 14.

- P14 No Connects (N/C) can be replaced with Interrupt and GPIO signals as detailed in Table 13 and Table 14.

- P14 Utopia signals can be replaced with McBSP1 signals as detailed in Table 12.

Please contact The Evergreen Group if you have a requirement for any of the above configurations.

4 I/O Interfaces

This chapter provides a description of the various interfaces on the VT-142x. This includes connector locations and pin assignments.

4.1 Connector Locations

Figure 3 on page 9 shows the location of the connectors on the VT-142x.

Header JP1 is the JTAG/Emulator header for connection of the C6400 emulator cable. See sections 3.2.1.7 and 4.2.1 for more information about the JTAG emulator interface.

Connectors P11-P13 carry the PCI interface to the PMC carrier board. Connector P14 provides the Utopia or EMIF (depending on the model number) interface from the first DSP to the carrier board. See sections 3.2.1.9, 3.2.2, and 4.2.2 for more information about the signals on the PMC connectors.

Switch S1 allows you to configure the operating mode of the two DSPs. See sections 3.2.1.5 and 5.8 for more information about boot mode settings.

There are no user-configurable jumpers on the VT-142x.

!

Caution – There are ground signals on the P14 connector that may not be compatible with your carrier board. Damage could occur if the VT-142x and carrier board do not have compatible UserIO connector pin assignments This situation would void the product warranty.

4.2 Connector Pin Assignments

This section summarizes the pin assignments for the following groups of interconnect signals for the VT-142x:

- JTAG Emulator Connector
- PMC Connectors
- PMC User IO Connector P14 options

4.2.1 JTAG Emulator Connector (JP1)

The pin assignments for this connector contain all the signals needed to interface to an emulator for C6400 DSP as required by Texas Instruments. Table 8 shows the pin assignments for the JTAG emulator connector.

!

Caution – The signals on the JTAG emulator connector are LVTTL voltage levels (3.3V) and are not 5V tolerant. Damage to the DSPs could occur if a 5V emulator is connected. This situation would void the product warranty.

!

Caution – The JTAG signals go directly to the DSPs without any additional ESD protection. Use proper ESD techniques when handling or making a connection to JP1.

!

Caution – The JTAG emulator connector is for debug purposes and is not as rugged as the PMC connectors. Rough handling of this connector could result in breakage. This situation would void the product warranty.

Table 8 – JTAG Emulator Connector (JP1) Pin Assignments

Pin #	Signal Name	Description
1	TMS	JTAG Test Mode Select
2	TRST#	JTAG Test Reset pin
3	TDI	JTAG Test Data Input
4	GND	Ground signal
5	VCC (3.3V)	Power for the Emulator pod
6	N/C	No connection (key)
7	TDO	JTAG Test Data Out
8	GND	Ground signal
9	TCK	JTAG Test Clock
10	GND	Ground signal
11	TCK_RET	Return signal for the clock path
12	GND	Ground signal
13	EMU0	Emulator connection 0
14	EMU1	Emulator connection 1

4.2.2 PMC Connectors (P11-P14)

Table 9 – PMC Connectors (P11 and P12) Pin Assignments (all models)

P11				P12			
Pin	**Signal**	**Signal**	**Pin**	**Pin**	**Signal**	**Signal**	**Pin**
1	N/C	-12V	2	1	12V	N/C	2
3	GND	INTA#	4	3	N/C	N/C	4
5	INTB#	INTC#	6	5	N/C	GND	6
7	BUSMODE1#	5V	8	7	GND	N/C	8
9	INTD#	N/C	10	9	N/C	N/C	10
11	GND	N/C	12	11	N/C	3.3V	12
13	CLK	GND	14	13	RST#	N/C	14
15	GND	GNT#	16	15	3.3V	BUSMODE4#	16
17	REQ#	5V	18	17	PME#	GND	18
19	VIO	AD[31]	20	19	AD[30]	AD[29]	20
21	AD[28]	AD[27]	22	21	GND	AD[26]	22
23	AD[25]	GND	24	23	AD[24]	3.3V	24
25	GND	C/BE[3]#	26	25	IDSEL	AD[23]	26
27	AD[22]	AD[21]	28	27	3.3V	AD[20]	28
29	AD[19]	5V	30	29	AD[18]	GND	30
31	VIO	AD[17]	32	31	AD[16]	C/BE[2]#	32
33	FRAME#	GND	34	33	GND	N/C	34
35	GND	IRDY#	36	35	TRDY#	3.3V	36
37	DEVSEL#	5V	38	37	GND	STOP#	38
39	GND	N/C	40	39	PERR#	GND	40
41	N/C	N/C	42	41	3.3V	SERR#	42
43	PAR	GND	44	43	C/BE[1]#	GND	44
45	VIO	AD[15]	46	45	AD[14]	AD[13]	46
47	AD[12]	AD[11]	48	47	M66EN	AD[10]	48
49	AD[09]	5V	50	49	AD[08]	3.3V	50
51	GND	C/BE[0]#	52	51	AD[07]	N/C	52
53	AD[06]	AD[05]	54	53	3.3V	N/C	54
55	AD[04]	GND	56	55	N/C	GND	56
57	VIO	AD[03]	58	57	N/C	N/C	58
59	AD[02]	AD[01]	60	59	GND	N/C	60
61	AD[00]	5V	62	61	ACK64#	3.3V	62
63	GND	REQ64#	64	63	GND	N/C	64

Table 10 – PMC Connectors (P13 and P14) Pin Assignments (standard VT-1420/23)

P13				P14			
Pin	**Signal**	**Signal**	**Pin**	**Pin**	**Signal**	**Signal**	**Pin**
1	N/C	GND	2	1	TXSOC	GND	2
3	GND	C/BE[7]#	4	3	GND	RXADR4	4
5	C/BE[6]#	C/BE[5]#	6	5	TXCLAV	TXADR4	6
7	C/BE[4]#	GND	8	7	RXADR3	GND	8
9	VIO	PAR64	10	9	N/C	GND	10
11	AD[63]	AD[62]	12	11	GND	N/C	12
13	AD[61]	GND	14	13	N/C	GND	14
15	GND	AD[60]	16	15	GND	RXENB#	16
17	AD[59]	AD[58]	18	17	TXADR3	RXCLAV	18
19	AD[57]	GND	20	19	TXADR2	GND	20
21	VIO	AD[56]	22	21	N/C	TXENB#	22
23	AD[55]	AD[54]	24	23	GND	RXADR2	24
25	AD[53]	GND	26	25	TXCLK	GND	26
27	GND	AD[52]	28	27	GND	TXADR1	28
29	AD[51]	AD[50]	30	29	TXADR0	RXADR1	30
31	AD[49]	GND	32	31	N/C	GND	32
33	GND	AD[48]	34	33	GND	RXADR0	34
35	AD[47]	AD[46]	36	35	TXD7	N/C	36
37	AD[45]	GND	38	37	TXD6	GND	38
39	VIO	AD[44]	40	39	N/C	RXD7	40
41	AD[43]	AD[42]	42	41	GND	RXD6	42
43	AD[41]	GND	44	43	RXCLK	GND	44
45	GND	AD[40]	46	45	GND	RXD5	46
47	AD[39]	AD[38]	48	47	TXD5	RXD4	48
49	AD[37]	GND	50	49	TXD4	GND	50
51	GND	AD[36]	52	51	GND	RXD3	52
53	AD[35]	AD[34]	54	53	TXD3	RXD2	54
55	AD[33]	GND	56	55	TXD2	GND	56
57	VIO	AD[32]	58	57	N/C	RXD1	58
59	N/C	N/C	60	59	TXD1	RXD0	60
61	N/C	GND	62	61	TXD0	GND	62
63	GND	N/C	64	63	GND	RXSOC	64

Table 11 – PMC Connectors (P13 and P14) Pin Assignments (standard VT-1425/26)

P13				P14			
Pin	**Signal**	**Signal**	**Pin**	**Pin**	**Signal**	**Signal**	**Pin**
1	N/C	GND	2	1	BECLKOUT1	GND	2
3	GND	C/BE[7]#	4	3	GND	DX1	4
5	C/BE[6]#	C/BE[5]#	6	5	BCE0#	CLKX1	6
7	C/BE[4]#	GND	8	7	FSX1	GND	8
9	VIO	PAR64	10	9	BAWE#	GND	10
11	AD[63]	AD[62]	12	11	GND	BARE#	12
13	AD[61]	GND	14	13	DSPA_EXT_INT5	GND	14
15	GND	AD[60]	16	15	GND	BAOE#	16
17	AD[59]	AD[58]	18	17	CLKS1	BARDY	18
19	AD[57]	GND	20	19	CLKR1	GND	20
21	VIO	AD[56]	22	21	GP6/EXT_INT6	BEA[20]	22
23	AD[55]	AD[54]	24	23	GND	FSR1	24
25	AD[53]	GND	26	25	BEA[19]	GND	26
27	GND	AD[52]	28	27	BEA[18]	BEA[17]	28
29	AD[51]	AD[50]	30	29	BEA[16]	DR1	30
31	AD[49]	GND	32	31	BEA[15]	GND	32
33	GND	AD[48]	34	33	BEA[14]	BEA[13]	34
35	AD[47]	AD[46]	36	35	BEA[12]	BEA[11]	36
37	AD[45]	GND	38	37	BEA[10]	BEA[9]	38
39	VIO	AD[44]	40	39	BEA[8]	BEA[7]	40
41	AD[43]	AD[42]	42	41	BEA[6]	BEA[5]	42
43	AD[41]	GND	44	43	BEA[4]	BEA[3]	44
45	GND	AD[40]	46	45	GND	BEA[2]	46
47	AD[39]	AD[38]	48	47	BEA[1]	BED[15]	48
49	AD[37]	GND	50	49	BED[14]	BED[13]	50
51	GND	AD[36]	52	51	BED[12]	BED[11]	52
53	AD[35]	AD[34]	54	53	BED[10]	BED[9]	54
55	AD[33]	GND	56	55	BED[8]	BED[7]	56
57	VIO	AD[32]	58	57	BED[6]	BED[5]	58
59	N/C	N/C	60	59	BED[4]	BED[3]	60
61	N/C	GND	62	61	BED[2]	BED[1]	62
63	GND	N/C	64	63	GND	BED[0]	64

! **Caution –** The signals on the P14 connector are LVTTL voltage levels (3.3V) and are not 5V tolerant. Damage to the DSPs could occur if a device with TTL voltage levels (5V) is connected. This situation would void the product warranty.

! **Caution –** There are ground signals on the P14 connector that may not be compatible with your carrier board. Damage could occur if the VT-142x and carrier board do not have compatible UserIO connector pin assignments. This situation would void the product warranty.

4.2.3 Special Configuration UserIO Signals on P14

The standard configuration for the VT-1420/23 series modules only connects the Utopia bus to P14. In this configuration, the VT-1420/23 is compatible with PICMG 2.15 Configuration 7 and in addition P14 is pin assigned per PICMG 2.15 Configuration 3. However for OEM purchases, the VT-1420/23 family can be configured to connect other signals to P14 via 0Ω jumper resistors. These include the following:

- P14 Utopia signals can be replaced with McBSP1 signals as detailed in Table 12.
- P14 No Connects (N/C) can be replaced with McBSP0 signals as detailed in Table 13 and Table 14.
- P14 No Connects (N/C) can be replaced with Interrupt and GPIO signals as detailed in Table 13 and Table 14.

Please contact The Evergreen Group if you have a requirement for any of the above configurations. The following information is supplied for completeness, but The Evergreen Group can not take responsibility for customer modifications.

! **Caution –** Any customer modifications would void the product warranty.

Table 11 illustrates the pin assignments for the standard (VT-1420/23) configuration of P14 with the Utopia interface versus a special configuration that pin assigns McBSP1 to P14 instead of the Utopia. The C6400 DSP has many pins that are dual purpose. Some of the Utopia interface pins are shared with the McBSP1 interface. When R8 is installed, the Utopia signals of DSP A are selected to P14 and when R8 is not installed, the McBSP1 signals are selected as illustrated in Table 11. Since there are more signals on the Utopia interface than the McBSP1 interface, when the McBSP1 interface is selected the remaining Utopia signals are in a Tri-State (high impedance) state but still connected to DSP A. An example of this is pin 1 where the TXSOC Utopia signal becomes Tri-State in the McBSP1 configuration.

Similarly, Table 12 and Table 13 highlight other signal assignment options to P14 (VT-1420/23). The standard configuration is *Not Connected* (N/C) for pins 9, 12, 13, 21, 31, 36, 39, and 57 but these tables indicate how other signals could be connected to these pins via 0Ω resistors. In particular, the McBSP0 shows how this serial port could be routed from DSPA and DSPB to P14. For example when R105 is assembled, DSPA assigns FSR0 and DSPB assigns FXR0 to pin 13. Since McBSP0 connects DSPA to DSPB, either DSP has the capability of operating McBSP0 on P14.

Table 12 – Special Configuration for Utopia and McBSP1 Signals on UserIO PMC Connector P14

USERIO Utopia and McBSP1 Signal Options for P14					
Pin	**Utopia Option R8=1KW**	**McBSP1 Option R8 not assembled**	**Utopia Option R8=1KW**	**McBSP1 Option R8 not assembled**	**Pin**
1	**TXSOC**	**Tri-State**	*GND*	*GND*	2
3	*GND*	*GND*	**RXADR4**	**DX1**	4
5	**TXCLAV**	**Tri-State**	**TXADR4**	**CLKX1**	6
7	**RXADR3**	**FSX1**	*GND*	*GND*	8
9	*N/C*	*N/C*	*GND*	*GND*	10
11	*GND*	*GND*	*N/C*	*N/C*	12
13	*N/C*	*N/C*	*GND*	*GND*	14
15	*GND*	*GND*	**RXENB#**	**Tri-State**	16
17	**TXADR3**	**CLKS1**	**RXCLAV**	**Tri-State**	18
19	**TXADR2**	**CLKR1**	*GND*	*GND*	20
21	*N/C*	*N/C*	**TXENB#**	**Tri-State**	22
23	*GND*	*GND*	**RXADR2**	**FSR1**	24
25	**TXCLK**	**Tri-State**	*GND*	*GND*	26
27	*GND*	*GND*	**TXADR1**	**Tri-State**	28
29	**TXADR0**	**Tri-State**	**RXADR1**	**DR1**	30
31	*N/C*	*N/C*	*GND*	*GND*	32
33	*GND*	*GND*	**RXADR0**	**Tri-State**	34
35	**TXD7**	**Tri-State**	*N/C*	*N/C*	36
37	**TXD6**	**Tri-State**	*GND*	*GND*	38
39	*N/C*	*N/C*	**RXD7**	**Tri-State**	40
41	*GND*	*GND*	**RXD6**	**Tri-State**	42
43	**RXCLK**	**Tri-State**	*GND*	*GND*	44
45	*GND*	*GND*	**RXD5**	**Tri-State**	46
47	**TXD5**	**Tri-State**	**RXD4**	**Tri-State**	48
49	**TXD4**	**Tri-State**	*GND*	*GND*	50
51	*GND*	*GND*	**RXD3**	**Tri-State**	52
53	**TXD3**	**Tri-State**	**RXD2**	**Tri-State**	54
55	**TXD2**	**Tri-State**	*GND*	*GND*	56
57	*N/C*	*N/C*	**RXD1**	**Tri-State**	58
59	**TXD1**	**Tri-State**	**RXD0**	**Tri-State**	60
61	**TXD0**	**Tri-State**	*GND*	*GND*	62
63	*GND*	*GND*	**RXSOC**	**Tri-State**	64

Table 13 – Special Configuration for and McBSP0, Interrupt and other Signals on UserIO PMC Connector P14 (Odd Row)

McBSP0, Interrupt and other USERIO Signal Options for P14			
Pin	**Standard**	**Signal Option 1 (Rx=0W)**	**Signal Option McBSP0 (Ry=0W) DSPA signal / DSPB signal**
1	*TXSOC*	*TXSOC*	*TXSOC*
3	*GND*	*GND*	*GND*
5	*TXCLAV*	*TXCLAV*	*TXCLAV*
7	*RXADR3*	*RXADR3*	*RXADR3*
9	**N/C**	**DSPA_TIMER_OUT1 (R100)**	**N/C (N/A)**
11	*GND*	*GND*	*GND*
13	**N/C**	**DSPA_EXT_INT5 (R103)**	**FSR0 / FXR0 (R105)**
15	*GND*	*GND*	*GND*
17	*TXADR3*	*TXADR3*	*TXADR3*
19	*TXADR2*	*TXADR2*	*TXADR2*
21	**N/C**	**DSPB_TIMER_OUT1 (R98)**	**DX0 / DR0 (R101)**
23	*GND*	*GND*	*GND*
25	*TXCLK*	*TXCLK*	*TXCLK*
27	*GND*	*GND*	*GND*
29	*TXADR0*	*TXADR0*	*TXADR0*
31	**N/C**	**DSPB_TIMER_IN2/OUT2 (R93)**	**DR0 / DX0(R96)**
33	*GND*	*GND*	*GND*
35	*TXD7*	*TXD7*	*TXD7*
37	*TXD6*	*TXD6*	*TXD6*
39	**N/C**	**DSPB_EXT_INT5 (R99)**	**CLKR0 / CLKX0 (R102)**
41	*GND*	*GND*	*GND*
43	*RXCLK*	*RXCLK*	*RXCLK*
45	*GND*	*GND*	*GND*
47	*TXD5*	*TXD5*	*TXD5*
49	*TXD4*	*TXD4*	*TXD4*
51	*GND*	*GND*	*GND*
53	*TXD3*	*TXD3*	*TXD3*
55	*TXD2*	*TXD2*	*TXD2*
57	**N/C**	**BOARD_RESET_L (R94) ***	**CLKS0 / CLKS0 (R97)**
59	*TXD1*	*TXD1*	*TXD1*
61	*TXD0*	*TXD0*	*TXD0*
63	*GND*	*GND*	*GND*

* BOARD_RESET_L is connected to an on-board driver source S_RST_L from the PCI bridge via R91. Either R91 or R94 is assembled but not both.

Table 14 – Special Configuration for and McBSP0, Interrupt and other Signals on UserIO PMC Connector P14 (Even Row)

McBSP0, Interrupt and other USERIO Signal Options for P14			
Signal Option 1 (Rx=0W)	**Signal Option McBSP0 (Ry=0W) DSPA signal / DSPB signal**	**Standard**	**Pin**
GND	*GND*	*GND*	2
RXADR4	*RXADR4*	*RXADR4*	4
TXADR4	*TXADR4*	*TXADR4*	6
GND	*GND*	*GND*	8
GND	*GND*	*GND*	10
DSPA_TIMER_IN2/OUT2 (R92)	**FSX0 / FSR0 (R95)**	**N/C**	12
GND	*GND*	*GND*	14
RXENB#	*RXENB#*	*RXENB#*	16
RXCLAV	*RXCLAV*	*RXCLAV*	18
GND	*GND*	*GND*	20
TXENB#	*TXENB#*	*TXENB#*	22
RXADR2	*RXADR2*	*RXADR2*	24
GND	*GND*	*GND*	26
TXADR1	*TXADR1*	*TXADR1*	28
RXADR1	*RXADR1*	*RXADR1*	30
GND	*GND*	*GND*	32
RXADR0	*RXADR0*	*RXADR0*	34
N/C (N/A)	**CLKX0 / CLKR0 (R104)**	**N/C**	36
GND	*GND*	*GND*	38
RXD7	*RXD7*	*RXD7*	40
RXD6	*RXD6*	*RXD6*	42
GND	*GND*	*GND*	44
RXD5	*RXD5*	*RXD5*	46
RXD4	*RXD4*	*RXD4*	48
GND	*GND*	*GND*	50
RXD3	*RXD3*	*RXD3*	52
RXD2	*RXD2*	*RXD2*	54
GND	*GND*	*GND*	56
RXD1	*RXD1*	*RXD1*	58
RXD0	*RXD0*	*RXD0*	60
GND	*GND*	*GND*	62
RXSOC	*RXSOC*	*RXSOC*	64

5 Programming Considerations

This chapter provides basic information useful in programming the VT-142x. This includes memory maps, PCI detection and configuration, register definitions, library functions, diagnostic software, and other programming considerations.

5.1 DSP Memory Map

The DSP memory map is shown in Table 15. In this memory map, general memories (internal RAM, SDRAM and Flash) are in bold for easy identification. In this memory map, the SDRAM and Flash external memory devices begin at the base address of the corresponding CE space. For example, a 16MB SDRAM in Bank0 (EMIFA CE0) has the address range from 0x8000_0000 to 0x80FF_FFFF or a 1MB Flash ROM Bank (EMIFA CE1) has the address range from 0x6400_0000 to 0x640F_FFFF. One exception to this is in the case of the 2MB Flash. For the Flash, the DSP memory map still only sees a 1MB footprint. The second (u pper) 1MB of Flash is paged via the DSP's GPIO0 signal (see section 3.2.1.3 Flash and EMIFB). See Table 16 for more information on Flash address.

In Table 16, the Flash Sector Address Map provides more detail for EMIFB CE1of Table 15 and shows the sector address ranges for both the 1MByte and 2MByte Flash devices. This table is useful when you erase or program by sectors. Notice that the second half of this table (SA19-SA34) has sector information that is only available to the 2MByte Flash devices when the upper page is selected.

Since both the DSP and PCI memory spaces are byte addressable, the developer will find it easier to develop code for both the DSP and the host processors when they address the same memory locations in the DSP memory map. There is no need for byte and word address translation between DSP and host processor code.

Table 15 – DSP Memory Map Summary

Memory Block Description	Block Size	DSP Address Range
Internal RAM (L2)	**1 MB**	**0x0000_0000 – 0x000F_FFFF**
Reserved	23 MB	0x0010_0000 – 0x017F_FFFF
External Memory Interface A (EMIFA) Registers	256 KB	0x0180_0000 – 0x0183_FFFF
L2 Registers	256 KB	0x0184_0000 – 0x0187_FFFF
HPI Registers	256 KB	0x0188_0000 – 0x018B_FFFF
McBSP 0 Registers	256 KB	0x018C_0000 – 0x018F_FFFF
McBSP 1 Registers	256 KB	0x0190_0000 – 0x0193_FFFF
Timer 0 Registers	256 KB	0x0194_0000 – 0x0197_FFFF
Timer 1 Registers	256 KB	0x0198_0000 – 0x019B_FFFF
Interrupt Selector Registers	256 KB	0x019C_0000 – 0x019F_FFFF
EDMA RAM and EDMA Registers	256 KB	0x01A0_0000 – 0x01A3_FFFF
McBSP2 Registers	256 KB	0x01A4_0000 – 0x01A7_FFFF
EMIFB Registers	256 KB	0x01A8_0000 – 0x01AB_FFFF
Timer 2 Registers	256 KB	0x01AC_0000 – 0x01AF_FFFF
GPIO Registers	256 KB	0x01B0_0000 – 0x01B3_FFFF
UTOPIA Registers	256 KB	0x01B4_0000 – 0x01B7_FFFF
TCP/VCP Registers[1]	256 KB	0x01B8_0000 – 0x01BB_FFFF
Reserved	256 KB	0x01BC_0000 – 0x01BF_FFFF
PCI Registers	256 KB	0x01C0_0000 – 0x01C3_FFFF
Reserved	4 MB – 256 KB	0x01C4_0000 – 0x01FF_FFFF
QDMA Registers	52B	0x0200_0000 – 0x0200_0033
Reserved	736 MB – 52B	0x0200_0034 – 0x2FFF_FFFF
McBSP 0 Data	64MB	0x3000_0000 – 0x33FF_FFFF
McBSP 1 Data	64MB	0x3400_0000 – 0x37FF_FFFF
McBSP 2 Data	64MB	0x3800_0000 – 0x3BFF_FFFF
UTOPIA Queues	64MB	0x3C00_0000 – 0x3FFF_FFFF
Reserved	256MB	0x4000_0000 – 0x4FFF_FFFF
TCP/VCP [1]	256MB	0x5000_0000 – 0x5FFF_FFFF
EMIFB CE0 – Unused	64MB	0x6000_0000 – 0x63FF_FFFF
EMIFB CE1 – 1 MB Flash ROM Bank[2,3]	**64MB**	**0x6400_0000 – 0x67FF_FFFF**
EMIFB CE2 – Unused	64MB	0x6800_0000 – 0x6BFF_FFFF
EMIFB CE3 – Unused	64MB	0x6C00_0000 – 0x6FFF_FFFF
Reserved	256MB	0x7000_0000 – 0x7FFF_FFFF
EMIFA CE0 – 16 or 32 MB SDRAM Bank 0	**256MB**	**0x8000_0000 – 0x8FFF_FFFF**
EMIFA CE1 – 16 or 32 MB SDRAM Bank 1	256MB	0x9000_0000 – 0x9FFF_FFFF
EMIFA CE2 – Unused	256MB	0xA000_0000 – 0xAFFF_FFFF
EMIFA CE3 – Unused	256MB	0xB000_0000 – 0xBFFF_FFFF
Reserved	1GB	0xC000_0000 – 0xFFFF_FFFF

[1]TMS320C6416 Only – Reserved on TMS320C6415

[2]Refer to Table 16 for Flash sector addresses.

[2]Memories (Flash, SDRAM) begin at the base address of the corresponding CE space.

Table 16 – EMIF B CE 1 Flash Sector Address Map

Memory Block Description	Block Size	DSP Address
SA0	16KB	0x6400_0000 – 0x6400_3FFF
SA1	8KB	0x6400_4000 – 0x6400_5FFF
SA2	8KB	0x6400_6000 – 0x6400_7FFF
SA3	32KB	0x6400_8000 – 0x6400_FFFF
SA4	64KB	0x6401_0000 – 0x6401_FFFF
SA5	64KB	0x6402_0000 – 0x6402_FFFF
SA6	64KB	0x6403_0000 – 0x6403_FFFF
SA7	64KB	0x6404_0000 – 0x6404_FFFF
SA8	64KB	0x6405_0000 – 0x6405_FFFF
SA9	64KB	0x6406_0000 – 0x6406_FFFF
SA10	64KB	0x6407_0000 – 0x6407_FFFF
SA11	64KB	0x6408_0000 – 0x6408_FFFF
SA12	64KB	0x6409_0000 – 0x6409_FFFF
SA13	64KB	0x640A_0000 – 0x640A_FFFF
SA14	64KB	0x640B_0000 – 0x640B_FFFF
SA15	64KB	0x640C_0000 – 0x640C_FFFF
SA16	64KB	0x640D_0000 – 0x640D_FFFF
SA17	64KB	0x640E_0000 – 0x640E_FFFF
SA18	64KB	0x640F_0000 – 0x640F_FFFF
SA19[1]	*64KB*	*0x6400_0000 – 0x6400_FFFF (upper page)*
SA20[1]	*64KB*	*0x6401_0000 – 0x6401_FFFF (upper page)*
SA21[1]	*64KB*	*0x6402_0000 – 0x6402_FFFF (upper page)*
SA22[1]	*64KB*	*0x6403_0000 – 0x6403_FFFF (upper page)*
SA23[1]	*64KB*	*0x6404_0000 – 0x6404_FFFF (upper page)*
SA24[1]	*64KB*	*0x6405_0000 – 0x6405_FFFF (upper page)*
SA25[1]	*64KB*	*0x6406_0000 – 0x6406_FFFF (upper page)*
SA26[1]	*64KB*	*0x6407_0000 – 0x6407_FFFF (upper page)*
SA27[1]	*64KB*	*0x6408_0000 – 0x6408_FFFF (upper page)*
SA28[1]	*64KB*	*0x6409_0000 – 0x6409_FFFF (upper page)*
SA29[1]	*64KB*	*0x640A_0000 – 0x640A_FFFF (upper page)*
SA30[1]	*64KB*	*0x640B_0000 – 0x640B_FFFF (upper page)*
SA31[1]	*64KB*	*0x640C_0000 – 0x640C_FFFF (upper page)*
SA32[1]	*64KB*	*0x640D_0000 – 0x640D_FFFF (upper page)*
SA33[1]	*64KB*	*0x640E_0000 – 0x640E_FFFF (upper page)*
SA34[1]	*64KB*	*0x640F_0000 – 0x640F_FFFF (upper page)*

[1]Flash sectors SA19 to SA34 correspond to the upper page of a 2MB Flash. SA19-SA34 are not available on a 1MB Flash.

5.2 Configuring the Host and Local PCI buses

The VT-142x has two PCI buses, a host PCI bus, sometimes referred to as the VT-142x PCI bus, and a local PCI bus, sometimes referred to as the on-board bus, bridged by a non-transparent bridge. Because of this non-transparent bridge, the host PCI bus and processors on this bus are not aware of the local PCI bus. In other words, to PCI devices on the host PCI bus the VT-142x simply appears to be an INTEL 21555 non-transparent bridge (see Figure 7 for PCI Configuration Space Register Map and Table 21 for DeviceID and VendorID information). Special configuration cycles are required to configure the PCI devices on the local PCI bus of the VT-142x. There are three basic parts to configuring the host and local PCI buses of the VT-142x:

- PCI Bridge serial ROM program
- PCI host controller configuration of the primary port of the PCI bridge
- User's host software which configures the secondary port of the PCI bridge and the PCI interfaces of the DSPs

The VT-142x is shipped with the serial ROM of the PCI bridge pre-programmed. The data file for this program is provided in the distribution with this product. The Windows drivers and utility that The Evergreen Group provides for this product can read and modify the contents of the serial ROM. After power up, the contents of the serial ROM are automatically written into the PCI bridge prior to the occurrence of any other PCI activity. Some of these values set by the serial ROM are unique and can only be set by the serial ROM while other values can later be modified by the host processor. For more information on bridge parameters, refer to the appropriate bridge documentation.

In most host computers, the BIOS or other similar function will automatically configure the primary port of the PCI bridge upon system boot up. The primary port of the bridge is the VT-142x's external PCI interface and is present on the host PCI bus. This host bus configuration occurs sometime after the serial ROM activity and usually before the occurrence of any local (on-board) PCI bus configuration. In the case where the host computer does not automatically perform this primary (host bus) port configuration, see section 5.3.

So far so good, right? Up to this point, everything has been performed automatically upon power up without any user interaction. However, to talk to the DSPs via the PCI bus you must configure all devices on the PCI local bus of the VT-142x. This includes both the secondary port of the bridge as well as the PCI interfaces of the DSPs. For the platforms supported by The Evergreen Group, there are API libraries and utilities to perform all of the necessary local PCI bus configurations. For the platforms not supported by The Evergreen Group, you will need to develop the same functions. Section 5.4 is provided to assist you in this task.

In general, you can use the Windows drivers and utility that The Evergreen Group provides for this product to read and modify all of the necessary configuration values for both the host and local PCI buses. When invoked, the Windows utility (Command Line Interface or CLI) from The Evergreen Group automatically configures all required registers that are not automatically configured by the serial ROM and host computer at boot up. If you need to set different values or if you intend to use the VT-142x with a different operating system, you can use the CLI to identify the values that you need. When using the VT-142x with another operation system, at minimum, you can start by using the same default PCI configuration values programmed into the VT-142x by the serial ROM and the CLI. Simply use the CLI to read the values in the serial ROM and then, the configuration space registers of the 21555 bridge and the DSP. Section 5.2.1 provides a summary of the default PCI configuration for the VT-142x.

5.2.1 Factory Default PCI Configuration and BAR Mapping

This section provides a summary of the default PCI bus configurations for both the host and local PCI buses of the VT-142x. The discussion generally assumes that the platform used is a Windows-based (WINTEL) platform.

The PCI bus uses Base Address Registers (BARs) to map memory space and I/O space of PCI devices. Refer to section 5.3.1 and the PCI specification (Table 34) for more information on BARs. When you generate application software for the host controller to communicate with a PCI device, typically you access the PCI devices via pointers to BARs. In addition a single PCI device can have multiple BARs. From the factory, the VT-142x (the 21555 PCI bridge) is configured with five BARs and each C6400 DSP on the local PCI bus has three BARs as illustrated in Figure 7 and Figure 8, respectively. Since the VT-142x uses a non-transparent bridge between the host and local PCI buses, the bridge translates PCI addresses on its primary port (host PCI bus) to its secondary port (local PCI bus) in order for host-side PCI devices to access the DSPs. This translation is performed via BARs 2-5 and the factory default translations provided by the Windows CLI are shown in Table 17. The CLI changes the translation value depending on which DSP is being accessed as described in the description column for the Bridge BARs in Table 17. The CLI automatically sets the bridge's translation registers appropriately when accessing DSP A and then changes the values when accessing DSP B.

Table 17 – Host PCI BAR Definitions and Default Downstream Mapping to DSP BARs

VT-142x (21555 Primary) BARs on host PCI bus	Bridge BAR Description	C6400 DSP BARs on local PCI bus	DSP BAR Description
BAR 0	Non-prefetch memory access to bridge (21555) internal registers Intel documentation nomenclature: **Primary CSR and Downstream Memory 0 BAR**	N/A	N/A
BAR 1	I/O access to bridge (21555) internal registers Intel documentation nomenclature: **Primary CSR I/O BAR**	N/A	N/A
BAR 2	Mapped to local bus I/O space[1] Intel documentation nomenclature: **Downstream I/O or Memory 1 BAR**	BAR 2	DSP I/O space[4]
BAR 3	Mapped to local bus prefetch memory[2] Intel documentation nomenclature: **Downstream Memory 2 BAR**	BAR 0	DSP prefetch memory space. This is a 4M Byte window. [4]
BAR 4	Mapped to local bus non-pretech memory space[3] Intel documentation nomenclature: **Downstream Memory 3 BAR**	BAR 1	DSP non-prefetch memory space. This is a 8MByte window. [4]
BAR 5	Not used Intel documentation nomenclature: **Downstream Memory 3 Upper 32 Bits**	N/A	N/A

[1] The value in the Downstream I/O Translated Base Register of the bridge [CSR space 0x6C] determines which device is selected; DSP A when the value is 0xA200 and DSP B when the value is 0xB200.

[2] The value in the Downstream Memory 2 Translated Base Register of the bridge [CSR space 0x70] determines which device is selected; DSP A when the value is 0xA000_0000 and DSP B when the value is 0xB000_0000.

[3] The value in the Downstream Memory 23Translated Base Register of the bridge [CSR space 0x74] determines which device is selected; DSP A when the value is 0xA100_0000 and DSP B when the value is 0xB100_0000.

[4] Refer to Figure 8 for more information on the DSP BARs.

On boot-up, the host controller on the host PCI bus sets the address in each of the five VT-142x BARs. These address values will change from system to system and even within a system with the system configuration changes, such as the addition of another PCI board. The windows device driver is used for developers of host code to associate these BAR address values to a VT-142x and bars within this VT-142x. These specific BAR values are normally hidden from the user. However, if you would like to view these system specific PCI settings, you can use the Windows CLI to read the 21555 Configuration Registers or refer to the appropriate Appendix of this manual for other PCI diagnostic tools.

On the local bus, there are three devices that the Windows CLI configures:

- DSP A
- DSP B
- Secondary Port of Bridge.

Table 18 shows the address set in each BAR of the local bus PCI devices by the Windows CLI. Generally, DSP A is mapped into "A" space, DSP B is mapped into "B" space, the Bridge internal registers are mapped into "C" space and upstream translation (provides the DSPs with access to other PCI devices on the host PCI bus) is mapped into "D" space.

Table 18 – Local PCI BAR Address Values for DSPs and 21555 Bridge

Local Bus Device	BAR	BAR Address	Description
21555 Secondary Port	Bar0	0x**C**000_0000	21555 Internal Registers
21555 Secondary Port	Bar1	N/A	Not Used
21555 Secondary Port	Bar2	N/A	Not Used
21555 Secondary Port	Bar3	0x**D**000_0000	Upstream to off-board PCI devices
21555 Secondary Port	Bar4	N/A	Not Used
DSP **A** (C64x)	Bar0	0x**A**000_0000	C64x 4MByte Prefetch Memory
DSP **A** (C64x)	Bar1	0x**A**100_0000	C64x 8MByte non-Prefetch Memory
DSP **A** (C64x)	Bar2	0x**A**200	C64x I/O
DSP **B** (C64x)	Bar0	0x**B**000_0000	C64x 4MByte Prefetch Memory
DSP **B** (C64x)	Bar1	0x**B**100_0000	C64x 8MByte non-Prefetch Memory
DSP **B** (C64x)	Bar2	0x**B**200	C64x I/O

From the information in Table 18 plus the documentation and libraries provided by Texas Instruments for the PCI peripheral, a DSP programmer can determine how to program the DSP's PCI interface to master transfers between the DSP and other PCI devices. The Evergreen Group has provided some example source code in installation distribution for the VT-142x products.

It is fairly straightforward to use the PCI interface to transfer data from one DSP to another or from DSP to internal registers in the 21555 PCI Bridge. However, it is slightly more complex to program the DSP to transfer data to a destination that is not on the VT-142x. There are some additional steps that an external PCI device must perform before a VT-142x can properly transfer data. An external PCI device on the host PCI bus must

- Determine the physical PCI address of the destination
- Program the value determined in the previous step into the PCI Bridge's Upstream translation register
- Or Provide the value determined to the DSP so the DSP can program it into the PCI Bridge's Upstream translation register.

Table 19 shows the BARs on the secondary port of the PCI bridge. In the standard configuration provided with the Windows CLI, the DSP must access BAR 3 at address 0xD000_0000 in order to access PCI devices on the host PCI bus but only after a device on the host PCI bus properly programs the translation value for BAR 3. The Windows CLI and API libraries supplied by The Evergreen Group provide a means to program this translation register.

Table 19 – Local PCI Bridge BAR Definitions and Default Upstream Mapping to External PCI

Bridge (21555 Secondary) BARs on local PCI bus	Description
BAR 0	Provides the DSPs with non-prefetch memory access to bridge (21555) internal registers. Intel documentation nomenclature: Secondary CSR Memory BAR
BAR 1	Not used. The DSPs have no capability to master I/O transactions. Intel documentation nomenclature: Secondary CSR I/O BAR
BAR 2	Not used. Intel documentation nomenclature: Upstream I/O or Memory 0 BAR
BAR 3	**Mapped to local bus prefetch memory space (The user must manually use the CLI to set the translation value to point to external PCI memory.)** **Intel documentation nomenclature: Upstream Memory 1 BAR**
BAR 4	Not used. Intel documentation nomenclature: Upstream Memory 2 BAR

5.2.1.1 Examples of a Host Initiating PCI Accesses to DSP Memory (Downstream)

The information in Table 15, Table 17, and Table 18 provides the details required by a PCI device on the host PCI bus to access DSP memory space. In the PCI bridge terminology this is a downstream access. Using these tables and Table 20 below, follow the series of examples below. In each case, it is assumed that the PCI Configuration registers of each device on the VT-142x have been properly set previously. The CLI performs these setups automatically when it is invoked.

Table 20 – Example of Host PCI BAR Address and Length Values

VT-142x (21555 Bridge Primary Port) BARs on host PCI bus	VT-142x (21555 Bridge Primary Port) BAR Example Address
BAR 0	0xF680_0000 (length=0x0000_1000)
BAR 1	0x0000_E001 (length=0x0000_0100)
BAR 2	0x0000_E401 (length=0x0000_0040)
BAR 3	0xFC00_0008 (length=0x0040_0000)
BAR 4	0xF600_0000 (length=0x0080_0000)
BAR 5	Not used.

PCI Host access to DSP A Internal Memory (Address 0x0)
Assuming that the DSP Page Register (DSPP) of DSP A is equal to 0x0, the PCI Host simply needs to access BAR 3 at address 0xFC00_0000. A PCI access at address 0xFC00_0000 is translated to a local PCI address of 0xA000_0000, which in turn is translated by the DSPP of DSP A to address 0x0 of the DSP memory space. The Windows CLI from The Evergreen Group does all of this for you automatically when you use the "Read, Write, or Fill DSP Memory" options and provide the CLI with a start address of 0x0.

PCI Host access to DSP A Internal Memory (Address 0x0100)
Assuming that the DSP Page Register (DSPP) of DSP A is equal to 0x0, the PCI Host simply needs to access BAR 3 at address 0xFC00_0100. A PCI access at address 0xFC00_0100 is translated to a local PCI address of 0xA000_0100, which in turn is translated by the DSPP of DSP A to address 0x0100 of the DSP memory space. The Windows CLI from The Evergreen Group does all of this for you automatically when you use the "Read, Write, or Fill DSP Memory" options and provide the CLI with a start address of 0x0100.

PCI Host access to DSP A SDRAM (Address 0x8000_0000)
Assuming that the DSP Page Register (DSPP) of DSP A is equal to 0x0000_0200 and that the EMIF registers of DSP A are properly configured, the PCI Host simply needs to access BAR 3 at address 0xFC00_0000. A PCI access at address 0xFC00_0000 is translated to a local PCI address of 0xA000_0000, which in turn is translated by the DSPP of DSP A to address 0x8000_0000 of the DSP memory space. The Windows CLI from The Evergreen Group does all of this for you automatically when you use the "Read, Write, or Fill DSP Memory" options and provide the CLI with a start address of 0x8000_0000.

PCI Host access to DSP A SDRAM (Address 0x8000_0400)
Assuming that the DSP Page Register (DSPP) of DSP A is equal to 0x0000_0200 and that the EMIF registers of DSP A are properly configured, the PCI Host simply needs to access BAR 3 at address 0xFC00_0400. A PCI access at address 0xFC00_0400 is translated to a local PCI address of 0xA000_0400, which in turn is translated by the DSPP of DSP A to address 0x8000_0400 of the DSP memory space. The Windows CLI from The Evergreen

Group does all of this for you automatically when you use the "Read, Write, or Fill DSP Memory" options and provide the CLI with a start address of 0x8000_0400.

PCI Host access to DSP **B** SDRAM (Address 0x8000_0400)
Assuming that the DSP Page Register (DSPP) of DSP B is equal to 0x0000_0200, that the EMIF registers of DSP B are properly configured, and that the BAR 3 translation value is set to DSP B (0xB0000_0000) the PCI Host simply needs to access BAR 3 at address 0xFC00_0400. A PCI access at address 0xFC00_0400 is translated to a local PCI address of 0xB000_0400, which in turn is translated by the DSPP of DSP B to address 0x8000_0400 of the DSP memory space. The Windows CLI from The Evergreen Group does all of this for you automatically when you select DSP A and use the "Read, Write, or Fill DSP Memory" options and provide the CLI with a start address of 0x8000_0400.

PCI Host non-prefetch access to DSP A EMIF Registers (Register Address 0x0180_0000)
Assuming that the BAR 4 translation value is set to select DSP A (0xA100_0000), the PCI Host simply needs to access BAR 4 at address 0xF600_0000. A PCI access at address 0xF600_0000 is translated to a local PCI address of 0xA100_0000, which in turn is translated by the BAR 1 of the DSP to address 0x0180_0000 of the DSP memory space. The Windows CLI from The Evergreen Group does this for you automatically when you use the "Read or Write 21555 Memory" options, select BAR 4, and provide the CLI with a start address of 0x0.

PCI Host I/O access to DSP A PCI I/O Register DSPP (I/O Address 0x08)
Assuming that the BAR 2 translation value is set to select DSP A (0xA200), the PCI Host simply needs to access bridge's BAR 2 (0xE400) at offset address 0x08. A PCI I/O access at address 0xE408 is translated to a local PCI I/O address of 0xA208, which in turn is translated by the BAR 2 of the DSP to address 0x08 of the DSP I/O space. The Windows CLI from The Evergreen Group does this for you automatically when you use the "Read or Write 21555 Memory" options, select BAR 2, and provide the CLI with a start address of 0x08.

5.2.1.2 Examples of a DSP PCI Initiating Accesses to Local PCI Device Memory

The information in Table 15 and Table 18 provides the details required by a developer to program the DSP to access memory space on the local PCI bus. Using these tables, follow the series of examples below. In each case, it is assumed that the PCI Configuration registers of each device on the VT-142x have been properly set previously. The CLI performs these setups automatically when it is invoked.

Note that DSP master accesses have a limitation where the DSP must access PCI space on doubleword (8 byte) address boundaries. Refer to the PCI section in the SPRU190 (Table 33) documentation from Texas Instruments for information on the PCI capabilities of the DSP and how to program the registers DSPMA, PCIMA and PCIMC of the DSP to initiate master transfers on the PCI bus. Also refer to the sample code provided by The Evergreen Group.

DSP A internal memory (0x000E_0000) write to DSP B internal memory (0x000F_0000)
Assuming that the DSP Page Register (DSPP) of DSP B is equal to 0x0, DSP A simply needs to access BAR 0 of DSP B at base address 0xB000_0000. DSP A needs to be programmed with both the address for the master DSP (DSPMA=0x000E_0000) and the local PCI address for DSP B (PCIMA=0xB00F_0000). As the DSP code executes, it will read bytes from its internal address 0x000E_0000 and write to them to the local PCI bus at 0xB00F_0000, where 0xB00F_0000 is translated by the DSPP of DSP B to address 0x000F_0000.

DSP A read from PCI Bridge Scratchpad Registers to internal memory (0x000E_0000)
DSP A simply needs to access BAR 0 (0xC000_0000 see Table 18 and Table 19) on the secondary port of the PCI bridge. The eight 32-bit scratchpad registers are located at the CSR byte offset addresses 0xA8 to 0xC7. DSP A needs to be programmed with both the address for the master DSP (DSPMA=0x000E_0000) and the local PCI address for scratchpad registers (PCIMA=0xC000_00A8). As the DSP code executes, it will read bytes from the local PCI bus at 0xC000_00A8, where 0xC000_00A8 is translated by the BAR 0 of the PCI bridge to address 0x0000_00A8 and write these values into its internal address 0x000E_0000.

5.2.1.3 Examples of a DSP PCI Initiating Access to External (Upstream) PCI Device Memory

The information in Table 15,Table 18, and Table 19 provides the details required by a developer to program the DSP to access memory space on the local PCI bus. Using these tables, follow the series of examples below. The assumptions and referenced information from section 5.2.1.2 are also pertinent to these examples.

DSP A internal memory (0x000E_0000) write to off-board memory at (0xF700_0000)
The assumption is that memory is accessible on the host PCI bus at 0xF700_0000. In this case, the PCI bridge Upstream Memory 1 Translated Base Register [CSR byte offset 0x7C] must be set to 0xF700_0000. This can be performed by either the host or the DSP[1]. Then, the DSP A simply needs to access the PCI Bridge BAR 3 (upstream memory 1) at the PCI local bus base address 0xD000_0000. DSP A needs to be programmed with both the address for the master DSP (DSPMA=0x000E_0000) and the PCI address for upstream transfers (PCIMA=0xD000_0000). As the DSP program executes, it will read bytes from its internal memory at address 0x000E_0000 and write them to the local PCI bus at starting at 0xD000_0000, where 0xD000_0000 is translated by the BAR 3 of the PCI bridge to address 0xF700_0000 of the host PCI bus.

The 21555 PCI bridge also has a lookup translation table for upstream transactions which can be quite useful in scatter gather type of activities. The lookup table at BAR4 (currently

[1] The CLI provides an option to this translation value. In addition, source code provided by The Evergreen Group for the both the host and the DSP show how this is performed.

unused for upstream memory 2) could be used in place of upstream memory 1. It is beyond the scope of this manual to provide such an example and instead is left to the reader as a trivial academic exercise.

5.3 Detecting and Configuring a VT-142x on the host PCI bus

In order for software to determine the presence of a VT-142x on the host PCI bus, it must examine the configuration space of the board. A board with the configuration values specified in Table 21 is a VT-142x and you may safely configure it as described in the following sections.

Table 21 – Defined Values for VT-142x PCI Configuration Space Registers

Register	Value	Description
Device ID	0xB555	The Device ID identifies the bridge device as the 21555.
Vendor ID	0x8086	The Vendor ID identifies Intel® as the vendor of the bridge device.
Subsystem ID	0x1420	The Subsystem ID identifies the subsystem that carries the bridge device as the VT-142x. Similarly, 0x1423 is used for the VT -1423.
Subsystem Vendor ID	0x15E5	The Subsystem Vendor ID identifies Valley Technologies, Inc. as the vendor of the VT-142x subsystem that carries the bridge device.

5.3.1 Configuring the VT-142x PCI Base Address Registers (BARs)

The PCI bus allows device independent software to determine what devices are present in the system and build a consistent address map for them. This means it has to determine how much memory space and I/O space the PCI devices in the system require. After determining this information, the software can map the PCI devices into reasonable locations. In order to do this mapping in a device independent manner, the PCI devices place the base address registers or BARs for this mapping in the predefined header portion of Configuration Space.

Software can determine whether a Base Address register indicates memory or I/O space by examining bit 0 of the register. In all Base Address registers this bit is read-only and indicates whether the register maps into Memory or I/O Space. Base Address registers that map to Memory Space will return a 0 in bit 0. Base Address registers that map to I/O Space will return a 1 in bit 0.

Software can determine how much address space an address register requires by writing a value of all 1's to the register and then reading the value back. The device will return 0's in all don't-care address bits, effectively specifying the address space required.

Unimplemented Base Address registers are hardwired to zero. This design implies that all address spaces are a power of two in size and are naturally aligned.

The steps involved in determining the size requested by an address register are as follows:

1. Software disables decoding of the Base Address register using the command register at offset 0x04 in the configuration header before sizing the Base Address register. To do this, software must clear the I/O Space Enable field in bit 0 of the command register and also clear the Memory Space Enable field in bit 1 of the command register.

2. Software reads and saves the original value of the Base Address register.

3. Software writes 0xFFFF_FFFF to the Base Address register.

4. Software reads back the Base Address register. Note that the software should ignore the upper 16 bits of the result if the Base Address register is for I/O and bits 16 through 31 returned zero when read.

5. Software clears the encoding information from the Base Address register: bit 0 for I/O and bits 0 through 3 for memory Base Address registers.

6. Software inverts all of the bits in the register (logical NOT).

7. Software increments the resulting value by 1. The resulting value is the size of the memory or I/O range that the register decodes.

   At this point, software should determine an appropriate location in its address map where a naturally aligned block of the requested size is available and write the starting address of that block to the Address Register. If software was simply trying to determine the size of a Base Address register whose value was already assigned, it should restore that value now.

8. Software enables decoding of the Base Address register using the command register before sizing a Base Address register. To do this, software must set the I/O Space Enable field in bit 0 of the control register and also set the Memory Space Enable field in bit 1 of the control register.

As shipped from the factory, the VT-142x presents five BARs to the host.

Figure 7 – VT-142x PCI Configuration Space Register Map

Address	Byte 3	Byte 2	Byte 1	Byte 0
0x00 read only	Device ID		Vendor ID	
0x04 read/write	Status		Command	
0x08 read only	Class Code			Revision ID
0x0C read/write	BIST	Header Type	Latency Timer	Cache Line Size
0x10 read/write	BAR 0			
0x14 read/write	BAR 1			
0x18 read/write	BAR 2			
0x1C read/write	BAR 3			
0x20 read/write	BAR 4			
0x24 read/write	BAR 5[1]			
0x28 read only	Reserved			
0x2C read only	Subsystem ID		Subsystem Vendor ID	
0x30 read/write	Expansion ROM BAR			
0x34 read only	Reserved			Capabilities Pointer
0x38 read only	Reserved			
0x3C read/write	Max Latency	Min_Grant	Interrupt Pin	Interrupt Line
0x40 read only	Secondary Port Device ID[2]		Secondary Port Vendor ID[2]	
0x44 read/write	Secondary Port Status[2]		Secondary Port Command[2]	
0x48 read only	Secondary Port Class Code[2]			Revision ID
0x4C read/write	BIST	Header Type	Latency Timer	Cache Line Size
0x50 read/write	Secondary Port BAR 0[2]			
0x54 read/write	Secondary Port BAR 1[2]			
0x58 read/write	Secondary Port BAR 2[2]			
0x5C read/write	Secondary Port BAR 3[2]			
0x60 read/write	Secondary Port BAR 4[2]			
0x64 read/write	Reserved			
0x68 .	Unused			
0x6C .	Unused			
0x70 read only	Reserved			
0x74 .	Unused			
0x78 .	Unused			
0x7C read/write	Max Latency	Min_Grant	Interrupt Pin	Interrupt Line
0x7F- 0xFF read/write	Other Secondary Port and Enhanced Capabilities Registers[2]			

[1]The VT-142x does not implement BAR 5. Reading BAR 5 returns all zeroes.

[2]Refer to the 21555 PCI bridge documentation from INTEL for more information.

5.3.1.1 VT-142x Base Address Register 0 (BAR 0)

BAR 0 is the 21555 Primary CSR and Downstream Memory 0 BAR of the bridge. Used as a Primary CSR BAR, it requests 4KB of memory space. This BAR allows access to the command and status registers within the bridge device from the host PCI bus using memory transactions.

Table 22 – VT-142x Base Address Register 0

Bit	Name	R/W	Description
0	Space Indicator	R	Indicates the type of address space to setup. Reads as 0 to indicate that memory space is requested.
2:1	Type	R	Indicates size and location of this address space. Reads as 00 indicating that this is a 32-bit address register and the space can be mapped anywhere in 32-bit memory.
3	Prefetchable	R	Indicates whether the region is prefetchable. Reads as 0 to indicate that nonprefetchable memory is requested.
11:4	Base Address Reserved LSBs	R	Used during sizing to determine requested size. Reads as 0.
31:12	Base Address	R/W	These bits indicate the size of the requested address range and set the base address of the range. These bits are writable and return the value last written when read. The low 4 KB of this address range map the 21555 CSRs into memory space.

5.3.1.2 VT-142x Base Address Register 1 (BAR 1)

BAR 1 is the 21555 Primary CSR I/O BAR of the bridge. It requests 256 bytes of I/O space. This BAR allows access to the command and status registers within the bridge device from the host PCI bus using I/O transactions.

Table 23 – VT-142x Base Address Register 1

Bit	Name	R/W	Description
0	Space Indicator	R	Indicates the type of address space to setup. Reads as 1 to indicate that I/O space is requested.
1	Reserved	R	Reserved. Reads as 0.
7:2	Base Address Reserved LSBs	R	Used during sizing to determine requested size. Reads as 0.
31:8	Base Address	R/W	These bits indicate the size of the requested address range and set the base address of the range. These bits are writable and return the value last written when read. The low 256 bytes of this address range map the 21555 CSRs into I/O space.

5.3.1.3 VT-142x Base Address Register 2 (BAR 2)

BAR 2 is the Downstream I/O or Memory 1 BAR of the bridge. Used as the Downstream I/O BAR, it requests 64 bytes of I/O space. This BAR allows the host to initiate I/O transactions on the VT-142x local PCI bus, typically intended for the DSPs. Please note that the TMS320C6415/16 DSPs only present 16 bytes of I/O space. Accesses within the 64-byte space of BAR 2 but outside the space of the DSP devices may result in unpredictable behavior.

Table 24 – VT-142x Base Address Register 2

Bit	Name	R/W	Description
0	Space Indicator	R	Indicates the type of address space to setup. Reads as 1 to indicate that I/O space is requested.
1	Reserved	R	Reserved. Reads as 0.
5:2	Base Address Reserved LSBs	R	Used during sizing to determine requested size. Reads as 0.
31:6	Base Address	R/W	These bits indicate the size of the requested address range and set the base address of the range. These bits are writable and return the value last written when read.

5.3.1.4 VT-142x Base Address Register 3 (BAR 3)

BAR 3 is the Downstream Memory 2 BAR of the bridge. It requests 4MB of memory space. This BAR allows the host to initiate memory transactions on the VT-142x local PCI bus, typically intended for the DSPs. The memory mapped by this BAR is prefetchable, as indicated by the assertion of bit 3_0 in the BAR.

Table 25 – VT-142x Base Address Register 3

Bit	Name	R/W	Description
0	Space Indicator	R	Indicates the type of address space to setup. Reads as 0 to indicate that memory space is requested.
2:1	Type	R	Indicates size and location of this address space. Reads as 00 indicating that this is a 32-bit address register and the space can be mapped anywhere in 32-bit memory.
3	Prefetchable	R	Indicates whether the region is prefetchable. Reads as 1 to indicate that prefetchable[1] memory is requested.
11:4	Base Address Reserved LSBs	R	Used during sizing to determine requested size. Reads as 0.
31:12	Base Address	R/W	These bits indicate the size of the requested address range and set the base address of the range. These bits are writable and return the value last written when read.

[1]A device can mark a range as prefetchable if there are no side effects on reads, the device returns all bytes on reads regardless of the byte enables, and host bridges can merge processor writes into this range without causing errors.

5.3.1.5 VT-142x Base Address Register 4 (BAR 4)

BAR 4 is the Downstream Memory 3 BAR of the bridge. It requests 8MB of memory space. This BAR allows the host to initiate memory transactions on the VT-142x local PCI bus, typically intended for the DSPs. The memory mapped by this BAR is non-prefetchable, as indicated by the negation of bit 3_0 in the BAR.

Table 26 – VT-142x Base Address Registe r 3

Bit	Name	R/W	Description
0	Space Indicator	R	Indicates the type of address space to setup. Reads as 0 to indicate that memory space is requested.
2:1	Type	R	Indicates size and location of this address space. Reads as 00 indicating that this is a 32-bit address register and the space can be mapped anywhere in 32-bit memory.
3	Prefetchable	R	Indicates whether the region is prefetchable. Reads as 0 to indicate that non-prefetchable memory is requested.
12:4	Size	R	Used during sizing to determine requested size. Reads as 0.
31:13	Base Address	R/W	These bits indicate the size of the requested address range and set the base address of the range. These bits are writable and return the value last written when read.

5.4 Initializing the VT-142x Local PCI Bus

The VT-142x uses a non-transparent bridge to connect the PCI interfaces of the DSP chips to the host PCI bus. This bridge allows the local PCI bus of the VT-142x to operate in isolation from the host bus. In addition, the non-transparent bridge offers a number of features for the VT-142x, like the ability to operate on a 64-bit, 66-MHz, universal PCI bus, among others. Unlike a transparent bridge, the computer BIOS does not completely configure a non-transparent bridge and the PCI devices on the other side. Additional configuration is necessary. For Windows users, The Evergreen Group. provides this functionality within its drivers, libraries and utilities. For users who are interested in writing their own drivers, we provide the following instructions. Please refer to the INTEL documentation listed in Table 33 for details on downstream indirect configuration accesses.

Like the VT-142x itself, each DSP implements a standard PCI configuration header. From Figure 8, you can see that the DSP implements three Base Address Registers or BARs. In order for software to determine the presence of a C6400 DSP on the local PCI bus, it must examine the configuration space of the local bus. A device with the configuration values specified in Table 21 is a C6400 DSP and you may safely configure it as this section describes.

Figure 8 – DSP PCI Configuration Space Register Map

Address	Byte 3	Byte 2	Byte 1	Byte 0
0x00 read only	Device ID		Vendor ID	
0x04 read/write	Status		Command	
0x08 read only	Class Code			Revision ID
0x0C read/write	Reserved	Header Type	Latency Timer	Cache Line Size
0x10 read/write	BAR0 (4 Mbyte prefetchable)[1]			
0x14 read/write	BAR1 (8 Mbyte non-prefetchable)[2]			
0x18 read/write	BAR2 (4 words I/O)[3]			
0x1C read/write	BAR3[4]			
0x20 read/write	BAR4[4]			
0x24 read only	BAR5[4]			
0x28 read only	Reserved			
0x2C read only	Subsystem ID		Subsystem Vendor ID	
0x30 read only	Reserved			
0x34 read only	Reserved			Capabilities Pointer
0x38 read only	Reserved			
0x3C read only	Max Latency	Min_Grant	Interrupt Pin	Interrupt Line
0x40 read only	Power Management Capabilities		Next_Item_Ptr	Cap_ID
0x44 read only	Power Data	Reserved	Power Management Control/Status	
0x48 – 0xFF	Reserved			

[1] Base 0: 4M-byte prefetchable maps to all of DSP memory with the window provided by the DSP Page register. Prefetch reads have all bytes valid.
[2] Base 1: 8M-byte non-prefetchable maps to DSP's memory-mapped registers. Windows starts at 0x0180_0000. Non-prefetch supports byte enables.
[3] Base 2: 16-byte I/O contains I/O registers for the PCI host

[4] The DSP does not implement BAR 3-5. Reading BAR 3-5 returns all zeroes.

Table 27 – Defined Values for DSP PCI Configuration Space Registers

Register	Value	Description
Device ID	0xA106	The Device ID identifies the DSP device as a C6000 DSP.
Vendor ID	0x104C	The Vendor ID identifies TI as the vendor of the DSP device.
Subsystem ID	0x0000	The Subsystem ID identifies the subsystem that carries the DSP device. This is unnecessary for the VT-142x and is left unprogrammed.
Subsystem Vendor ID	0x0000	The Subsystem Vendor ID identifies a vendor of the subsystem. This is unnecessary for the VT-142x and is left unprogrammed.

5.4.1 Performing Configuration Transactions on the VT-142x Local PCI Bus from the PCI Host Bus

One of the differences between a transparent and a non-transparent bridge is that a non-transparent bridge does not *transparently* pass a type 0 configuration transaction on to its secondary bus. This means that the host must perform a special set of actions in order to configure the devices on the VT-142x local PCI bus. Software generates these configuration transactions using downstream[2] indirect configuration[3] transactions.

To generate a downstream configuration transaction, the Downstream Configuration Control bit in the Configuration CSR must first be set.

To perform a configuration transaction on the VT-142x local PCI bus, software writes the address to the Downstream Configuration Address at offset 0x80 in configuration space. Software must write the Downstream Configuration Address register with the address to be driven before it accesses the corresponding data register. This address is driven on the AD lines of the VT-142x local PCI bus exactly as written in the register. Therefore, since all configuration transactions on the VT-142x local PCI bus are type 0 configuration transactions (AD[1:0] = 0) a Type 0 format must be used to generate a Type 0 configuration transaction. The upper 21 bits of a Type 0 address format are used as IDSEL signals. The VT-142x uses AD30 to select DSP A and AD31 to select DSP B.

When software reads or writes the Downstream Configuration Data register at offset 0x84 in configuration space the bridge initiates the configuration transaction on the VT-142x local PCI bus. Software must access this register using either a configuration transaction or an I/O transaction to initiate the transaction. The local transaction uses the same byte enables that the host used to read or write the Downstream Configuration Data register. The bridge responds to the access of the Downstream Configuration Data register with a target retry until the access is completed on the VT-142x local PCI bus. When the access is completed, the bridge returns the corresponding target termination and, if a read, the read data on a subsequent attempt of the transaction by the host.

---

[2] Downstream refers to transactions that are initiated from the host PCI bus as opposed to upstream transactions that are initiated from the local VT-1420 PCI bus.

[3] Indirect configuration is a two-step (address then data) process that allows software to initialize the secondary side (the side facing the VT-1420 local PCI bus) of the bridge from the primary side (the side facing the host PCI bus). This is necessary because the TMS320C6415/16 cannot perform PCI configuration cycles. The host CLI features functions that perform indirect configuration accesses to the DSPs.

Figure 9 – Pseudo-code for Indirect Configuration of the VT-142x Local PCI Bus

```
Extern PCIWriteConfig(uint32 offset, uint32 value);
Extern uint32 PCIReadConfig(uint32 offset);

Typedef enum {
    kDSPA,
    kDSPB,
} tTarget;

#define kDSPA_IDSEL (0x40000000)
#define kDSPB_IDSEL (0x80000000)

#define kCONFIG_CSR_OFFSET (0x92)
#define kCONFIG_ADDR_OFFSET (0x80)
#define kCONFIG_DATA_OFFSET (0x84)

void VT1420EnableIndirectConfig() {
    uint32 value = pciReadConfig(kCONFIG_CSR_OFFSET);
    pciWriteConfig(kCONFIG_CSR_OFFSET, (value | 0x2));
}

void VT1420ConfigWrite(tTarget target, uint32 offset, uint32 value) {
    switch (target) {
    case kDSPA;
        offset |= kDSPA_IDESL; break;
    case kDSPB;
        offset |= kDSPB_IDESL; break;
    }
    pciWriteConfig(kCONFIG_ADDR_OFFSET, offset);
    pciWriteConfig(kCONFIG_DATA_OFFSET, value);
}

uint32 VT1420ConfigRead(tTarget target, uint32 offset) {
    switch (target) {
    case kDSPA;
        offset |= kDSPA_IDESL; break;
    case kDSPB;
        offset |= kDSPB_IDESL; break;
    }
    pciWriteConfig(kCONFIG_ADDR_OFFSET, offset);
    return pciReadConfig(kCONFIG_DATA_OFFSET);
}
```

5.4.2 Configuring the VT-142x Local Bus

Follow these basic steps to configure the VT-142x local PCI bus. Steps 8 through 10 may not be directly part of the local PCI bus, but they are required for PCI transactions between devices on the host PCI bus and devices on the local PCI bus.

1. Enable the bridge to perform downstream indirect configuration transactions. To do this, perform normal configuration transactions to read the Bridge Configuration CSR from offset 0x92 in the Bridge PCI configuration header, set bit 1_0, and write the modified value back.

   Use the procedure described in section 5.4.1 to perform downstream indirect configuration transactions as required in the steps below.

2. Enable each DSP PCI Interface to respond to both Memory and I/O transactions. To do this, perform downstream indirect configuration transactions to read the DSP PCI command register from offset 0x04 in the DSP PCI configuration header, set bits 1_0 and 0_0, and write the modified value back.

3. Properly set the DSP Base Address Registers (BARs). To do this, use a procedure like the one described in section 5.4.3 using downstream indirect configuration transactions. From Figure 8, you can determine the byte offset address for each of the BARs in the DSP PCI configuration header.

4. Enable each DSP PCI Interface to master PCI transactions. To do this, use downstream indirect configuration transactions to read the DSP PCI command register from offset 0x04 in the DSP PCI configuration header, set bit 2_0 and write the modified value back. This step can be combined with step 2 above.

5. Enable the secondary port of the PCI bridge for both Memory and I/O transactions. To do this, use normal configuration transactions to read the bridge PCI secondary port command register from offset 0x44 in the bridge configuration header, set bits 1_0 and 0_0, and write the modified value back.

6. Properly set the secondary port BARs of the PCI bridge. To do this, use a procedure like the one described in section 5.3.1 using normal configuration transactions to the bridge. From the bridge documentation found in Table 33, you can determine the byte offset address for each of the secondary port BARs in the bridge PCI configuration header.

7. Enable the secondary port of the PCI bridge to master PCI transactions. To do this, use normal configuration transactions to read the bridge PCI secondary port command register from offset 0x44 in the DSP PCI configuration header, set bit 2_0 and write the modified value back. This step can be combined with step 5 above.

8. If not already done, enable the primary port of the PCI bridge to master PCI transactions. To do this, use normal configuration transactions to read the bridge PCI primary port command register from offset 0x04 in the DSP PCI configuration header, set bit 2_0 and write the modified value back

9. Set the downstream translation addresses to map PCI transactions on the primary port of the Bridge to DSP BARs. To do this, use memory transactions to set the values of the Translated Base Registers in the CSR space (or BAR 0) of the bridge. The values written into the Translated Base Registers must correlate to the values written into the DSP BARs in step 6 above. The offsets in BAR 0 for three of the Downstream Translated Base Registers are:

CSR Offset	**Bridge Translation Register Name**	**Bridge BAR #**	**Description**
0x6C	Downstream I/O or Memory 1Translated Base Register	BAR 2	Suggest map to DSP I/O space. See Table 17.
0x70	Downstream Memory 2 Translated Base Register	BAR 3	Suggest map to DSP memory space.
0x7C	Downstream Memory 3 Translated Base Register	BAR 4	Suggest map to DSP memory space.

10. Set the upstream translation addresses to map PCI transactions on the secondary port of the Bridge (from the DSPs) to external PCI devices. To do this, use memory transactions to set the desired value in the Upstream Memory 1 Translated Base Register at offset 0x7C of the CSR space in the Bridge. Note: Either the host or the DSP can be set any register in CSR, including this translation register, however the desired value must be determined by the host system.

5.4.3 Configuring the DSP Base Address Registers

Except for the use of downstream indirect configuration accesses, the procedure to configure the DSP BARs is the same as the procedure described in section 5.3.1, Configuring the VT-142x PCI Base Address Registers (BARs). The following sections provide detail of the register bit fields in the DSP BARs.

5.4.3.1 C6400 DSP Base Address Register 0

BAR 0 is one of two DSP Memory BARs. It requests 4MB of memory space that is a window into the DSP memory space. The base of this window is defined by the DSP register DSPP and it allows access to the entire memory space of the DSP.

Table 28 – PCI Base Address Register 0 for the C6400 DSP

Bit	Name	R/W	Description
0	Space Indicator	R	Indicates the type of address space to setup. Reads as 0 to indicate that memory space is requested.
2:1	Type	R	Indicates size and location of this address space. Reads as 00 indicating that this is a 32-bit address register and the space can be mapped anywhere in 32-bit memory.
3	Prefetchable	R	Indicates whether the region is prefetchable. Reads as 1 to indicate that prefetchable memory is requested.
21:4	Base Address (Size)	R	Used during sizing to determine requested size. Reads as 0.
31:22	Base Address	R/W	These bits indicate the size of the requested address range and set the base address of the range. These bits are writable and return the value last written when read. The low 4 MB of this address range map to the DSP 4 Mbyte window into memory space. The Base Address bits <31:22> are translated in the DSP by DSPP register bits <9:0> to DSP memory space.

[1]A device can mark a range as prefetchable if there are no side effects on reads, the device returns all bytes on reads regardless of the byte enables, and host bridges can merge processor writes into this range without causing errors.

5.4.3.2 C6400 DSP Base Address Register 1

BAR 1 is the second of two DSP Memory BARs. It requests 8MB of memory space that is a window into the DSP memory space. The base of this window is defined to 0x0180_000 and it allows access to the registers of the DSP. This region can also be accessed by BAR 0 but BAR 1 is defined to non-prefetchable.

Table 29 – PCI Base Address Register 1 for the C6400 DSP

Bit	Name	R/W	Description
0	Space Indicator	R	Indicates the type of address space to setup. Reads as 0 to indicate that memory space is requested.
2:1	Type	R	Indicates size and location of this address space. Reads as 00 indicating that this is a 32-bit address register and the space can be mapped anywhere in 32-bit memory.
3	Prefetchable	R	Indicates whether the region is prefetchable. Reads as 0 to indicate that nonprefetchable memory is requested.
22:4	Base Address (Size)	R	Used during sizing to determine requested size. Reads as 0.
31:23	Base Address	R/W	These bits indicate the size of the requested address range and set the base address of the range. These bits are writable and return the value last written when read. The low 8 MB of this address range map to the DSP 8 Mbyte window into memory space. The Base Address bits <31:23> are translated in the DSP to 0x0180_0000 of DSP memory space.

[1]A device can mark a range as prefetchable if there are no side effects on reads, the device returns all bytes on reads regardless of the byte enables, and host bridges can merge processor writes into this range without causing errors.

5.4.3.3 C6400 DSP Base Address Register 2

BAR 2 is an I/O BAR. It requests 16 bytes of I/O space. This BAR allows access to the PCI I/O Registers.

Table 30 –PCI Base Address Register 2 for the C6400 DSP

Bit	Name	R/W	Description
0	Space Indicator	R	Indicates the type of address space to setup. Reads as 1 to indicate that I/O space is requested.
1	Reserved	R	Reserved. Reads as 0.
3:2	Base Address (Size)	R	Used during sizing to determine requested size. Reads as 0.
31:4	Base Address	R/W	These bits indicate the size of the requested address range and set the base address of the range. These bits are writable and return the value last written when read.

5.5 Library Functions

The Evergreen Group provides both host and target (C6400 DSP) API libraries and example code using each of the libraries within the VT-142x distribution. These libraries provide useful functions such as toggling LEDs, setting up the EMIF, programming the Flash, and in the case of the host libraries, setting up the PCI interfaces of the devices on the local PCI bus. Refer to the The Evergreen Group software documentation in Table 32 for more information of the library functions.

5.6 Diagnostic Software

The Evergreen Group provides a Command Line Interface (CLI) utility with the host software. This utility can be used to run memory tests, which verifies a good portion of the board. In addition, the Flash comes preprogrammed with some diagnostic code that can be run 'right out of the box' in the Flash boot configuration. This Flash code performs a DSP subsystem self-test and indicates a pass or fail status with a green or red LED, respectively. The C6400 source for the diagnostic code is available with the product distribution. Refer to the The Evergreen Group software documentation in Table 32 for more infomation.

5.7 Other Considerations

Good programming practice dictates that only one processor at a time has control of certain VT-142x control registers over the PCI bus. Of particular note are the following registers:

- Registers that modify the address map (PCI Configuration Space)

- ❑ Interrupt request and general registers
- ❑ DMA registers

In addition, be careful that the host does not access the program space of the DSP while code is executing.

5.8 Setup and Initialization of the Module

5.8.1 Boot mode configuration

Since the DSPs have a variety of modes of operation and boot sequences, a switch provides for easy configuration of these settings. Table 31 describes how the various switch settings affect the boot mode of the DSPs. Refer to the Texas Instruments documentation on boot mode operation for any particular setting. In simplest form, all switches up (ON) means the DSPs will boot from the PCI host and all switches down (OFF) means the DSPs will boot from Flash.

Table 31 – Configuration for Setup Switch (SW1)

SW1 pos.	Signal name	Switch Setting and Description
1:0	DSPB_BOOT_MODE[1:0]	Type of boot process for DSP B. The available boot processes are: SW1<1> SW1<0> Boot Mode OFF OFF Flash ROM Boot (default) OFF ON Reserved ON OFF No Boot ON ON PCI Boot
3:2	DSPA_BOOT_MODE[1:0]	Type of boot process for DSP A. The available boot processes are: SW1<1> SW1<0> Boot Mode OFF OFF Flash ROM Boot (default) OFF ON Reserved ON OFF No Boot ON ON PCI Boot

Appendix A – Troubleshooting the VT-142x Module

Solving Startup and Installation Problems

Installation troubleshooting information is included in the software installation instructions included on the installation distribution.

If you are using Windows as the operating system, you can download a free PCI probe application from BSquare (formerly Blue Water Systems) at www.bsquare.com (formerly www.bluewatersystems.com). The PCI probe application, called PCIview, allows the user to look into the Configuration Space Registers of all PCI devices that are visible on the host bus.

Appendix B – Ordering Related Documentation

The Evergreen Group Documents

The publications listed below are included on the installation distribution, and may be referenced in this document.

Table 32 – The Evergreen Group Documents

Document Title	Document number	Revision
VT-142x Dual C64x DSP PMC Module User's Manual (this manual)	560-000-039	2.2
VT-142x Software Reference Manual for Host API Library	560-002-039	1.2 (or later)
VT-142x Command Line Interface (CLI) User's Manual	560-003-039	1.1 (or later)
VT-142x Series Data Sheets: VT-1420/23 Data Sheet VT-1225/26 Data Sheet		

Manufacturers' Documents

The following manufacturers' documents are included with the VT-142x installation distribution. Please note, that in some cases, the information is preliminary and the revision levels of the documents are subject to change without notice.

Table 33 – Manufacturers' Documents

Document Title	Source	Document name
TMS320C6000 CPU and Instruction Set Reference Guide	TI	SPRU189x[1]
Manual Update Sheet: TMS320C6000 CPU and Instruction Set Reference Guide	TI	SPRZ168x[1]
TMS320C6000 Peripherals Reference Guide	TI	SPRU190x[1]
Manual Update Sheet: TMS320C6000 Peripherals Reference Guide	TI	SPRZ122x[1]
TMS320C6000 Programmer's Guide	TI	SPRU198x[1]
TMS320C6415 Fixed-Point Digital Signal Processor Data Sheet	TI	SPRS146x[1]
TMS320C6415 Silicon Errata	TI	SPRZ010x[1]
TMS320C6416 Fixed-Point Digital Signal Processor Data Sheet	TI	SPRS164x[1]
TMS320C6416 Silicon Errata	TI	SPRZ011x[1]
21555 Non-Transparent PCI-to-PCI Bridge Datasheet	Intel	278320-00x[1]
21555 Non-Transparent PCI-to-PCI Bridge User Manual	Intel	278321–00x[1]
21555 Non-Transparent PCI-to-PCI Bridge Specification Update	Intel	278337-00x[1]
21555 Non-Transparent PCI-to-PCI Bridge Performance Optimization Application Note	Intel	278381-00x[1]

[1]'x' is the Alphanumeric character indicting the revision of the document.

Related Specifications

For additional information, refer to the following table for related specifications. Please note that in some cases, the information is preliminary and the revision levels of the documents are subject to change without notice.

Table 34 – Related Specifications

Document Title and Source	Publication Number/Rev
IEEE Standard for a Common Mezzanine Card Family: CMC	IEEE 1386-2001
IEEE Standard Physical and Environmental Layers for PCI Mezzanine Cards: PMC	IEEE 1386.1-2001
PCI Local Bus Specification	Rev 2.2, December 18, 1998
Utopia Level 1	Version 2.01, March 21, 1994 (af-phy-0017.000)
Utopia Level 2	Version 1.0, June 1995 (af-phy-0039.000)
PCI Telecom Mezzanine/Carrier Card Specification, PICMG® 2.15	Revision 1.0, April 11, 2001

Appendix C – List of Acronyms, Abbreviations and Initializations

API – Application Programming Interface

ATM – Asynchronous Transfer Mode

BAR – Base Address Register

BIST – Built-In Self-Test

CLI – Command Line Interface

CMC – Common Mezzanine Card

DMA – Direct Memory Access

DSP – Digital Signal Processor or Digital Signal Processing

EDMA – Enhanced DMA

EMIF – External Memory Interface

EEPROM – Electrically Erasable Programmable ROM

ESD – Electro-Static Discharge

GPIO – General Purpose Input/Output

HPI – Host Port Interface

IEEE – The Institute of Electrical and Electronics Engineers

JTAG – Joint Test Action Group, referred as Boundary Scan standard IEEE 1149.1

LSB – Least Significant Bit

L1, L2 – Level one, Level two cache memory

McBSP – Multi-channel Buffered Serial Port

MSB – Most Significant Bit

PCI – Peripheral Component Interconnect

PHY – Physical Layer

PMC – PCI Mezzanine Card

QDMA – Quick DMA

RAM – Random Access Memory

ROM – Read Only Memory

RTDX – Real Time Data exchange

SAR – Segmentation and Reassembly

SDRAM – Synchronous Dynamic RAM

TCP/VCP – Turbo Decoder CoProcessor/Viterbi Decoder CoProcessor

Utopia – Universal Test & Operations PHY Interface for ATM

CUSTOMER SUPPORT

Technical Assistance

The Evergreen Group technical support staff is available to assist you with questions that you may have. Please contact us using one of the methods shown below.

Phone: (610) 871-1955 – at anytime

Fax: (610) 680-3305

Email: support@evergreengrp.com

Website: http://www.evergreengrp.com

World Wide Web

The Evergreen Group maintains an active site on the Web. The site contains current information about the company and locations of sales offices, new and existing products, contacts for sales, service and technical support information. You can also send email to The Evergreen Group using the web site. Requests for sales, service, and technical support information will receive a prompt response.

Return for Repair

You must obtain a Returned Material Authorization (RMA) number before sending any product to The Evergreen Group for repair. Please contact The Evergreen Group using one of the methods shown above to obtain an RMA number. Please be ready with your name, telephone number, company name, company address, shipping address, invoicing address, product number, date purchased, and a technical description of the problem. We will apply a service charge to units that are out of warranty. Please pack the unit you are returning in anti-static material and ship in a sturdy cardboard box with adequate packing material. If possible, re-use the original shipping containers and packaging. Please follow good ESD control practices when handling the product. Mark the RMA number clearly on the outside of the box before returning.

Before you ship the product, include the following information:

- ❑ Return address

- Contact names and phone numbers
- RMA number
- A technical description of the problem

Ship the product to:

The Evergreen Group

4460 Bachman Drive

Schnecksville, PA 18078

Warranty Service

For specifics on warranty coverage, please reference the warranty statements posted on our website. Products without fault sent in for warranty repair, are subject to a re-certification charge. The warranty does not cover customer-induced damage resulting from misuse, abuse, or exceeding the specifications of a product.

Non-Warranty Services

There are several classes of non-warranty service. These include:

- Repair of customer-induced problems
- Repairs of failures for products outside the warranty period
- Re-certification (function testing) of a product in or out of warranty
- Procurement of spare parts.

All non-warranty repairs are subject to service charges. The Evergreen Group has determined that the pricing of repairs based on time and materials are more cost-effective for the customer than a flat-rate repair charge. After receiving the product, The Evergreen Group will analyze the product for a fixed fee. Once the repair costs have been determined, The Evergreen Group will request authorization from the customer to proceed with the repairs. After the customer authorizes the repairs and makes the billing arrangements, The Evergreen Group will repair the product and return it to the customer.

Life Support Policy

The Evergreen Group products are not authorized for use as critical components in life support devices or systems without the express written approval of the president of The Evergreen Group. Refer to the following for definitions of critical components and life support devices.

1. A critical component is any component of a life support device or system whose failure to perform can be expected to cause the failure of the life support device or system, affect its safety, or limit its effectiveness.

2. Life support devices or systems are devices or systems which support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

Office Location:

The Evergreen Group
4460 Bachman Drive
Schnecksville, PA 18078

Tel: 610.871.1955
Fax 610.680.3305

Internet:

Web Site: http://www.evergreengrp.com
http://www.pmcmodules.com
E-Mail: info@evergreengrp.com