TEXAS INSTRUMENTS

PCI2040 EVM

Hardware Guide

User's Guide

1999

PCIBus Solutions

SCPU003

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

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

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

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI 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 TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Copyright © 1999, Texas Instruments Incorporated

Preface

Read This First

About This Manual

This manual is provided to assist the developer in designing the PCI2040 in a system which incorporates a DSP-PCI bridge controller.

How to Use This Manual

This document contains the following chapters:

Chapter 1, *Introduction*, lists the key features of the PCI2040 evaluation module (EVM) and provides functional overviews of the EVM hardware and software.

Chapter 2, *Getting Started*, explains how to get started using the EVM. It provides step-by-step hardware and software installation instructions.

Chapter 3, *Running the Board Confidence Test*, explains how to run the board Confidence Test included with the PCI2040 EVM board.

Chapter 4, *PCI2040 EVM Hardware*, describes the EVM hardware, its key components and how they operate, and its various interfaces. Detailed programmer interface information such as memory maps, register definitions, and interrupt usage are also provided.

Chapter 5, *EVM Confidence Test*, describes the automated command line utility that enables the user to verify proper installation and operation of the board.

Chapter 6, *Host CPLD Registers*, describes the five host CPLD registers that are mapped into the PCI2040's GP bus address space.

Chapter 7, *DSP CPLD Registers*, describes the eight DSP CPLD registers that are mapped into the DSP's lower I/O address space.

Chapter 8, *Jumper Definitions*, describes the pinout information for connectors J4 through J10.

Related Documentation From Texas Instruments

PCI2040 DSP-PCI Bridge Controller Data Manual

PCI2040 EVM Software Guide

Trademarks

TI is a trademark of Texas Instruments Incorporated.
† Texas Instruments Customer Response Center

Contents

Figures

Tables

Chapter 1

Introduction

This chapter lists the key features of the PCI2040 evaluation module (EVM) and provides functional overviews of the EVM hardware and software.

The PCI2040 EVM is a low-cost, general-purpose platform for the evaluation of the PCI2040 PCI-to-DSP bridge and the development, analysis, and testing of C54x digital signal processor (DSP) algorithms and applications. The EVM allows you to develop applications that interface with the PCI2040 that run on both a host PC and the C54x DSP to determine if they meet your application requirements. The EVM hardware design information and software application programming interfaces (APIs) also provide a reference design that facilitates your own PCI2040 hardware and software development.

The EVM is bundled with software support software that includes a debugger, Windows 95/98 and NT 4.0 device drivers, host PC and DSP software APIs, example applications with source code, and various utility applications. This hardware/software bundle provides an integrated package that allows you to quickly evaluate the PCI2040 and C54x DSP devices and develop applications.

Topic **Page**

1.1 Key Features

The PCI2040 EVM bundle has the following key features:

- ❑ PCI2040 PCI-to-DSP bridge
- ❑ 3.3-V/5-V, PCI Local Bus Specification Revision 2.2-Compliant Interface
- ❑ 100 MHz TMS320VC5410 DSP (100 MIPS)
- ❑ 64 kW Internal SRAM
- ❑ 64 kW Program SRAM
- ❑ 64 kW Data SRAM
- ❑ Stereo 8–96 kHz, 16-bit Audio (TI's TLC320AD77C)
- ❑ Microphone & Line In/Out Interfaces
- ❑ JTAG Debugging via PCI Bus or XDS510 Emulator
- ❑ Memory & Peripheral Expansion Connectors
- ❑ Support for 32-bit C6x Daughterboards
- ❑ PCI Status and User-Defined LEDs
- ❑ Code Composer Studio Debugging Support (driver provided)
- ❑ Windows and DSP Drivers, APIs, and Utilities
- ❑ Plug and Play Device
- ❑ PCI2040 PCI-to-DSP Reference Design

1.1.1 User Control and Indicators

The PCI2040 EVM has the following user controls and indicators:

- ❑ JTAG Emulation (embedded or external)
- ❑ Clock Mode Selection (CLKMD1–3)
- ❑ Microprocessor/Microcomputer Mode Selection (MP/MC)
- ❑ Hot Swap Insert/Eject Control
- ❑ HPI/Daughterboard Boot Selection
- ❑ Two User-Defined Switches
- ❑ Line Input Level Selection (1-V/2-Vrms)
- ❑ Line Output Filtering Selection
- ❑ Manual Reset
- ❑ Power-On LED

- ❑ D3 Status LED
- ❑ Power Management Event (PME) Status LED
- ❑ $\overline{\text{ENUM}}$ Status LED
- ❑ Hot Swap Status LED
- ❑ User-Defined LED

1.1.2 External Interfaces

The PCI2040 C54x EVM has the following external interfaces:

- ❑ PCI Local Bus Specification Revision 2.2-Compliant Interface (3.3-V/5-V, 32-bit, 33 MHz)
- ❑ 14-pin External JTAG Header
- ❑ Three 3.5-mm Audio Jacks (line in, line out, and microphone)
- ❑ Two Low-Profile 80-pin (0.050-inch) Daughterboard Connectors
- ❑ Molex 4-pin External Power Connector (5V, 12V, –12V, and GND)

1.2 The PCI2040 EVM Board

The EVM is a short PCI board with approximate dimensions of 4.2 inches wide, 8.66 inches long, and 0.45 inch high. The EVM is intended for use in a PCI expansion slot on your computer's motherboard. The EVM can be operated stand-alone on a desktop with the use of an external power supply and XDS510 or XDS510WS emulator. The power supply and emulator required for stand-alone operation are not included since this board is intended for PCI evaluation purposes.

Note:

Make sure that there is an unobstructed slot open for the PCI card. See Section 2.1.2, *Avoiding Obstructions to the EVM Board*, for more information.

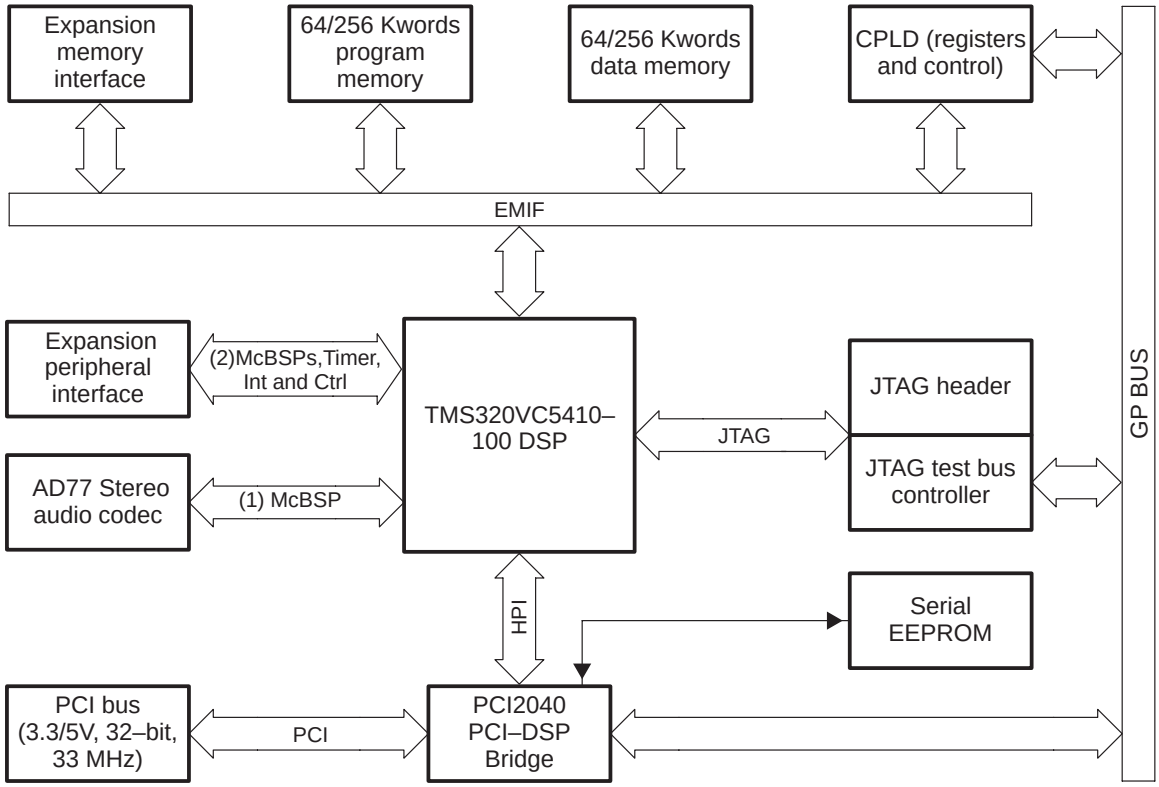
The EVM has a 5410 DSP onboard that enables full-speed verification of C54x code with the included source debugger. The EVM provides a PCI interface, external memory, an audio codec, and embedded JTAG emulation support. Connectors on the EVM provide the DSP's external memory interfaces and peripheral signals that enable its functionality to be expanded with custom or third-party daughterboards.

The EVM provides a PCI2040 hardware reference design that can assist you in the development of your own PCI2040-based products. In addition to providing a reference for interfacing the PCI2040 to various devices, it also provides a reference for interfacing the 5410 DSP to other devices.

1.3 Hardware Functional Overview

Figure 1–1 shows the basic functional block diagram of the EVM.

Figure 1–1. PCI2040 EVM Block Diagram



The EVM hardware can be divided into ten functional areas. This section provides an overview of each of these ten areas. Detailed descriptions are provided in Chapter 4, *PCI2040 EVM Hardware*.

- PCI Interface. The EVM is a PCI target device that is compliant with *PCI Local Bus Specification, Revision 2.2*. The PCI interface is provided by the PCI2040 PCI-to-DSP bridge which provides a glueless interface to the DSP's host port interface (HPI), JTAG test bus controller (TBC), and programmable logic. The PCI interface is target-only, meaning that all communications are initiated by a PCI bus master. For more information, see Section 4.1, *PCI Interface.*
- Clocks. The EVM provides onboard clocks for the JTAG TBC, DSP, CPLD, and audio codec. For more information, see Section 4.2, *Clocks*.
- Logic. The EVM provides logic for user control over the board from both the host and DSP. For more information, see Section 4.3, *Logic*.
- JTAG Emulation. The EVM provides embedded JTAG emulation using an onboard JTAG TBC, as well as a header to support an XDS510 JTAG emulator. This allows source debugging over the PCI bus without requiring

an emulator or by using an XDS510 emulator when operating stand-alone on a desktop. For more information, see Section 4.4, *JTAG Emulation*.

- Digital Signal Processor. The EVM includes a 100-MHz TMS320VC5410 digital signal processor (DSP) that provides the processing unit on the board. For more information, see Section 4.5, *Digital Signal Processor*.
- External Memory. The EVM provides 64Kx16 each of external program and data memory. This external memory complements the DSP's 64Kx16 of on-chip memory. For more information, see Section 4.6, *External Memory*.
- Stereo Audio Interface. The EVM includes a CD-quality, 16-bit audio interface with stereo microphone and line-level inputs and a stereo line-level output. A multimedia audio codec is used to provide popular sample rates from 8 to 96 kHz. The audio circuit includes a microphone preamplifier with microphone biasing. Three 3.5-mm audio jacks are located on the board's mounting bracket. For more information, see Section 4.7, *Stereo Audio Interface.*
- Expansion Interfaces. The EVM provides external memory and peripheral interface connectors that enable the use of custom or third-party daughterboards. For more information, see Section 4.8, *Expansion Interfaces*.
- Power Supplies. The EVM uses voltage regulators to provide 2.5 V for the DSP core, 3.3 V for the DSP I/O and onboard low-voltage devices, and 3.3 V for the analog components. Voltage supervision is also provided. For more information, see Section 4.9, *Power Supplies*.
- User Options and Indicators. The EVM supports user option control via eight on-board DIP switches, direct control over the PCI bus and the use of onboard jumpers. For more information, see Section 4.10, *User Options and Indicators*.

1.4 Software Functional Overview

The EVM software support consists of host support software and DSP support software. The host software supplied with the EVM includes the following host utilities and libraries:

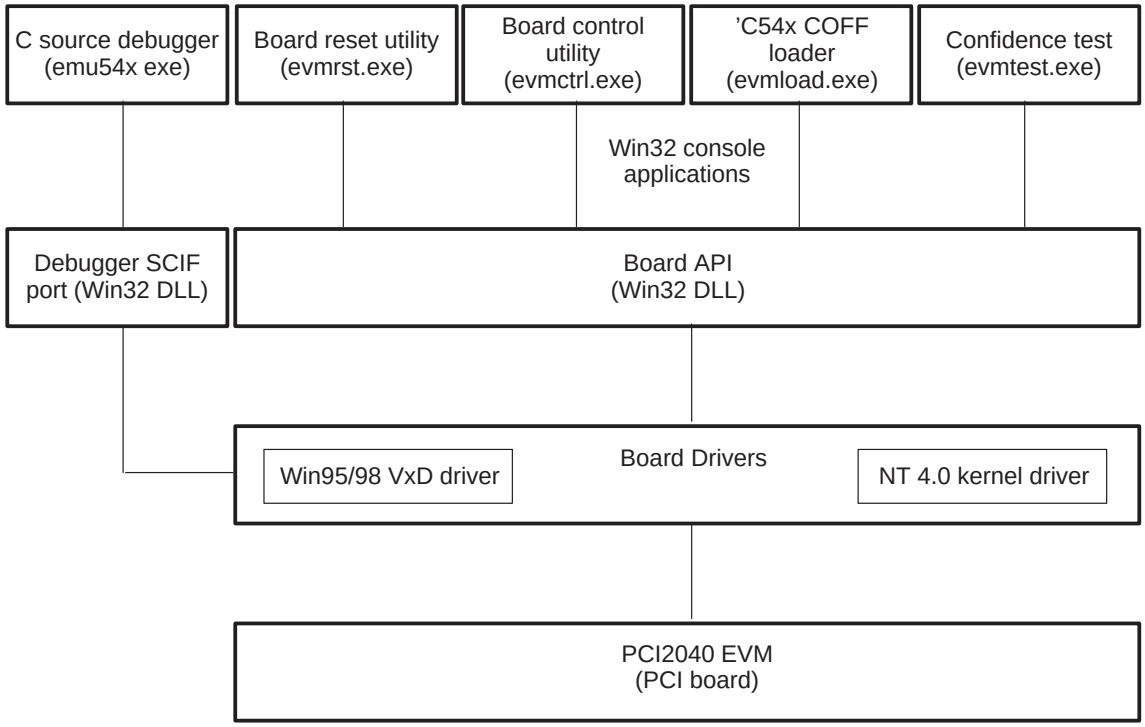
- C source debugger (emu54x.exe). The debugger software tools help you debug ‘C54x software on the board.
- EVM board reset utility (evmrst.exe). This utility resets the board.
- EVM confidence test utility (evmtest.exe). This utility tests the basic operation of the board.
- EVM board control utility (evmctrl.exe). This utility performs various control and status operations with the board.
- EVM COFF loader utility (evmload.exe). This utility loads and executes ‘C54x software on the board.
- EVM Win32 DLL (pci2040.dll). The Win32 host libraries consist of a user-mode dynamic linked library (DLL) that supports Windows 95, Windows 98, and Windows NT. This DLL provides user software access for control and communication with the EVM board.
- Example source code. Example code that illustrates how to use the Win32 DLL functions is provided with the PCI2040 EVM.

The host software supplied with an EVM board provides utilities to configure the board, to debug PCI2040 ‘C54x software on the board, to load and execute ‘C54x software on the board, and to test the basic operation of the board. Also supplied with the EVM board is a Win32 dynamic link library (DLL) that provides user software access for control and communication with the EVM board. The host utilities and host libraries run on an Intel PC under either Windows 95, Windows 98, or Windows NT 4.0. Figure 1–2 provides a block diagram of the EVM host software components and their relationships.

The far left side of the figure shows the components involved in the C source debugger support. The TI C source debugger makes calls to the debugger SCIF component. The SCIF component calls the JTAG TBC API functions to perform emulation functions. The JTAG TBC API calls the low-level (Ring 0) Win 95 and NT drivers to access the JTAG TBC device on the EVM board.

The remainder of the figure shows the Win32 host utilities and the components they use to access the hardware. The Win32 applications call the Windows 95, Windows 98, or Windows NT DLL that implement a consistent Ring 3 API. These DLLs make calls to the Ring 0 drivers which provide access to the hardware.

Figure 1–2. EVM Host Software Block Diagram



The DSP support software developed for the EVM board includes the following components:

- Board support library (board.c and board.h). This library provides 'C54x board specific routines for EVM configuration and control.
- Codec library (codec.c and board.h). This library is a collection of routines that configures and controls the operation of the AD77 audio codec device.
- Example source code. Code examples are provided to demonstrate the use of the codec and board support functions.

Chapter 2

Getting Started

This chapter explains how to get started using the EVM. It provides step-by-step hardware and software installation instructions.

Topic **Page**

2.1 Before Starting the Installation Process

2.1.1 Controlling Static Electricity

To help avoid equipment damage, read and follow the instructions in the following sections for controlling static electricity and avoiding obstructions that could prevent you from installing your EVM.

- ❏ Store the board in its anti-static bag until you are ready to use it.
- ❏ Keep the anti-static bag for storing or transferring the board between computers.
- ❏ Wear a properly connected ground strap at all times when handling the EVM.

2.1.2 Avoiding Obstructions to the EVM Board

Before installing the EVM, make sure that your PC has an available PCI slot that does not have obstructions. In some computers, the location or height of the central processing unit (CPU) or the location of the PCI-bus slot may prevent you from installing the EVM. Possible obstructions include memory modules, processor fans, power supply circuits, and cable connections. Failure to avoid obstructions when installing the EVM could permanently damage the card.

2.2 EVM Hardware Installation

To install the EVM board into your PC, follow these steps:

1) Turn off your PC's power and unplug the power cord.

2) Remove the cover from your PC.

3) Locate an existing unused PCI slot. Make sure that this slot will allow the installation of the EVM card. If the slot is a shared ISA/PCI slot that shares mechanical space, then make sure that an ISA card is not already installed.

4) If installed, remove the PCI slot's cover from the back of the computer by unscrewing it from the chassis.

5) Remove the EVM from the anti-static bag.

6) Insert the EVM board into the empty PCI slot and secure its mounting bracket to the chassis with the previously removed screw. There are no jumpers or switches on the board that have to be set for PCI operation.

7) Replace the PC cover.

8) Connect the 3.5-mm audio cables to the EVM's microphone, line in, and line out connectors as needed. The connectors are labeled with the MIC IN, LINE IN, and LINE OUT designations. MIC IN is closest to the status LED at the top, LINE IN is in the middle, and LINE OUT is at the bottom.

9) Plug in the PC's power cord and turn on the power.

If your operating system is Windows NT, then the hardware installation is finished. Proceed to Section 2.3, *EVM Software Installation*. If your operating system is Windows 95 or 98, then please read the following information.

During system startup, the PC's PCI BIOS will automatically detect the presence of the EVM and Windows will prompt you for a driver disk. Insert the EVM support software CD-ROM and follow the instructions on the screen. After the driver is installed, reboot the PC. To install the EVM support software, proceed to Section 2.3*, EVM Software Installation*.

2.3 EVM Software Installation

On most Windows 95 or NT 4.0 systems, setup will automatically start when the **PCI2040 Software** CD is loaded. However, if AutoPlay is not enabled on your system, then you will need to explicitly run the program setup.exe in the top-level directory of the CD. For example, you may do this in the following way:

1) Insert the **PCI2040 Software** CD in your CD-ROM drive.

2) Choose "Run..." from the Start Menu.

3) In the "Open:" box, type `d:\setup`, where d is the letter assigned to your CD-ROM drive.

4) Click OK and follow the instructions on the screen.

```
PATH=c:\pci2040;%path%
```

In Windows 95, this is set up in the `autoexec.bat` file.

In Windows NT, this is set up in the registry under:

```
HKEY_CURRENT_USER\ENVIRONMENT
```

If you choose not to have the installation program modify your environment, then you will need to modify it yourself. In addition to this modification of the `PATH` variable, you may wish to set up or modify other environment variables for your convenience in accessing files, setting code generation options, etc.

After you have completed installation of the EVM hardware and software, you should run the EVM confidence test to verify correct installation of the EVM hardware and software. This test is described in Section 5.1, *EVM Confidence Test*.

Chapter 3

Running the Board Confidence Test

This chapter explains how to run the board confidence test included with the PCI2040 EVM board. This test can only be run after the EVM board hardware and software has been successfully installed. Refer to Chapter 2 for installation instructions.

Topic **Page**

3.1 About the Board Confidence Test

The confidence test verifies proper operation of the major components of the EVM board and its support software and displays board configuration information. The EVM hardware components tested include:

- Memory
- C54x interrupts
- Audio codec
- CPLD semaphores
- PCI controller and board LEDs
- JTAG

The installation of these EVM software components are tested:

- Win32 DLL
- Host low-level board driver

The following section explains how to run the board confidence test utility to verify that the board and software have been installed properly. For additional information about the confidence test utility, see Section 5.1, *EVM Confidence Test.*

3.2 Verifying the EVM Installation

To run the board confidence test utility, follow these steps:

- ❑ Install the PCI2040 EVM board and software as described in Chapter 2, *Getting Started*.
- ❑ Open a command window and change directory to the conftest subdirectory of the EVM installation directory. This directory is created during the install process and contains additional files needed for the execution of the confidence test utility.
- ❑ Execute the confidence test utility in a command window by entering the command line:
  - evmtest test.log
  - This command starts the confidence test utility for the first EVM board and logs the test results to the file test.log.
- ❑ The output from the confidence test will look similar to that below.

```
C:>evmtst test.log
PCI2040 'C54x EVM Confidence Test, Version 1.00
04:20:25 PM   May 12, 1999
Board Index                         : 0
Board Type                          : PCI2040 'C54x EVM, Rev 0
Beginning Confidence Test ...
Testing Win32 DLL and Low-Level Driver access..........PASSED.
CPLD Dump....
=================================================
CPLD Revision                       : 7
Board Voltage                       : GOOD
Board Reset                         : NO
TBC Reset                           : NO
DSPINT3                             : NOT SET
NMI                                 : NOT SET
HINT2                               : NOT PENDING
User Switch 1                       : SET
User Switch 2                       : SET
User LED                            : OFF
DSP mode                            : MICROCONTROLLER
Daughter board                      : ABSENT
User semaphore 0                    : NOT OWNED
User semaphore 1                    : NOT OWNED
=================================================
***** PCI2040 Confidence Tests.
Observe the following PCI2040 LEDs
---------------------------------------
PME LED.....................................................PASSED.
Toggle dip switch 6 to observe the Hot Swap Enum LED.....PASSED.
Hot Swap LED................................................PASSED.
***** JTAG Confidence Tests.
JTAG Test...................................................PASSED.
***** EVM Clock Confidence Tests.
DSP 0:Clock Test............................................PASSED.
***** EVM LED Confidence Tests.
DSP 0: Observe the user LED on the bracket blinking......PASSED.
***** DSP Semaphore Confidence Tests.
DSP 0: Semaphore Test.......................................PASSED.
***** DSP Interrupt Confidence Tests.
DSP 0: HOST-to-DSP Interrupt Tests..........................PASSED.
DSP 0: DSP-to-HOST Interrupt Tests..........................PASSED.
***** Memory Tests.
DSP 0: A5 Test    Test: page 1: addr 0x8000: len 0x8000..PASSED.
DSP 0: WALK Test  Test: page 1: addr 0x8000: len 0x8000..PASSED.
DSP 0: ADDR Test  Test: page 1: addr 0x8000: len 0x8000..PASSED.
DSP 0: PULSE Test Test: page 1: addr 0x8000: len 0x8000..PASSED.
DSP 0: A5 Test    Test: page 2: addr 0x8000: len 0x8000..PASSED.
DSP 0: WALK Test  Test: page 2: addr 0x8000: len 0x8000..PASSED.
DSP 0: ADDR Test  Test: page 2: addr 0x8000: len 0x8000..PASSED.
DSP 0: PULSE Test Test: page 2: addr 0x8000: len 0x8000..PASSED.
***** Codec Confidence Tests.
DSP 0: Codec Left Channel Tone Test.........................PASSED.
DSP 0: Codec Right Channel Tone Test........................PASSED.
DSP 0: Codec Left/Right Channel Tone Test...................PASSED.
DSP 0: Codec Line In Loopback Test..........................PASSED.
DSP 0: Codec Mic In Loopback Test...........................PASSED.
***** PCI2040 'C54x EVM Confidence Test Complete.......No Errors.
C:>
```

Chapter 4

PCI2040 EVM Hardware

This chapter describes the EVM hardware, its key components and how they operate, and its various interfaces. Detailed programmer interface information such as memory maps, register definitions, and interrupt usage are also provided.

The hardware can be divided into ten functional areas. Each of these functional areas is discussed in this chapter.

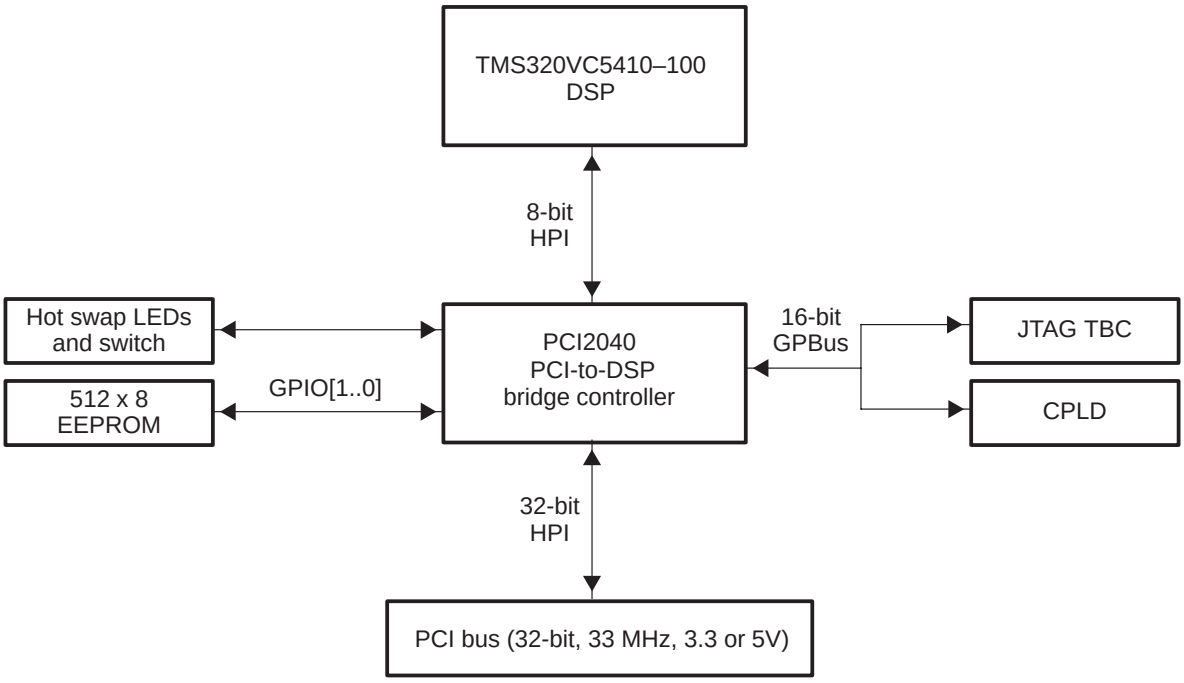
Topic **Page**

4.1 PCI Interface

The EVM has a 3.3-V/5-V, 32-bit universal card edge connector which allows it to be used in PCI slots that use either 3.3-V or 5-V signaling. The PCI2040 PCI-to-DSP bridge controller provides a *PCI Local Bus Specification*, Revision 2.2 compliant interface on the EVM.

Figure 4–1 shows the interfaces of the PCI2040 on the EVM.

Figure 4–1. PCI2040 Interfaces on EVM



In addition to providing a direct interface to the PCI bus, the PCI2040 also interfaces to the following devices on the EVM board:

- Glueless connection to the DSP's 8-bit host port interface (HPI) that supports data transfers and host-DSP interrupt capabilities. Although the PCI2040 supports up to four DSPs, only one DSP is on the EVM board.
- Complex programmable logic device (CPLD) that provides EVM and board control and status features.
- JTAG test bus controller (TBC) to provide embedded JTAG debugging of the DSP.
- Serial EEPROM device that stores the PCI configuration of the EVM board such as subsystem vendor and ID.
- Hot swap switch and LEDs

The PCI2040 interfaces to the CPLD and TBC devices via its 16-bit general-purpose I/O (GPIO) bus that supports a memory space of up to 64

addresses. The serial EEPROM interface uses 2 of the PCI2040's GPIO signals (GPIO1 and GPIO0) that are used for the serial clock and data. The PCI2040 provides a dedicated input for a hot swap switch (cPCI ejector handle) and two outputs for status ($\overline{\text{HSENUM}}$ and HSLED) that connect to LEDs for display.

A 512-byte serial EEPROM (AT24C04N) is used on the EVM. The PCI2040 controller only requires the first 32 bytes, so 480 bytes are available for manufacturer and user data.

The PCI2040 controller supports up to four DSPs, but only one DSP is supported on the EVM board. The HPI control signals for the three unused DSP ports are not used (inputs are pulled high). However, the unused reset outputs ($\overline{\text{HRST3}}$–$\overline{\text{HRST1}}$) and the unused GPIO3 signal support JTAG programming of the onboard CPLD over the PCI bus. This enables customer and field upgrades of the board without special programming software and hardware.

The PCI2040 controller provides hot-swap capabilities that are intended for use on a CompactPCI platform. The PC's PCI support does not currently provide hot swap. However, the EVM provides control and monitoring of these capabilities through the use of a switch input and LED output. This enables the EVM to be used for evaluation of the PCI2040's hot swap capabilities in a PCI environment.

The PCI2040 controller is also compliant with the latest *PCI Bus Power Management Interface Specification*. The PCI2040 provides several low-power features that reduce power consumption. The EVM supports the $\overline{\text{PME}}$ (power management event) signal, but it is not connected to the PCI bus by default. The EVM does provide LED indications of the $\overline{\text{PME}}$ and D3 power states to support evaluation of the PCI2040's power management capabilities.

The EVM provides 0-ohm resistors and two-pin jumper headers to support current measurements of the PCI2040's core voltage and PCI/HPI clamping voltages.

The PCI2040 controller presents four memory-mapped regions to the EVM software as shown in Table 4–1. Software interacts with the EVM by dealing with these four memory regions. There are no I/O-mapped regions on the EVM board.

The GP bus region on the EVM is divided into 2 halves, with the lower 32 doubleword addresses dedicated to the JTAG TBC, and the upper 32 doubleword addresses dedicated to the CPLD. Each GP bus access must be a 16-bit WORD access, but on a doubleword boundary. The CPLD only uses the lower 8 bits, so they should be masked appropriately on reads.

The support software for the EVM handles all of the memory accessing, so most users do not need to deal with it directly. The 32-kbyte control space is divided into four 8-kbyte regions to address up to four DSPs. Since only one DSP is supported on the EVM, only the lower 4-kbyte region accesses the DSP's HPIA, HPIC, HPID, and HPID with address autoincrementing. Further details about the four memory regions are provided in the *PCI2040 Data Manual*.

Table 4–1. EVM PCI Memory-Mapped Regions

Memory Region	Size (Bytes)
PCI configuration registers	256
HPI CSR registers	4 K
Control space (HPI registers)	32 K
GP bus (TBC and CPLD)	256

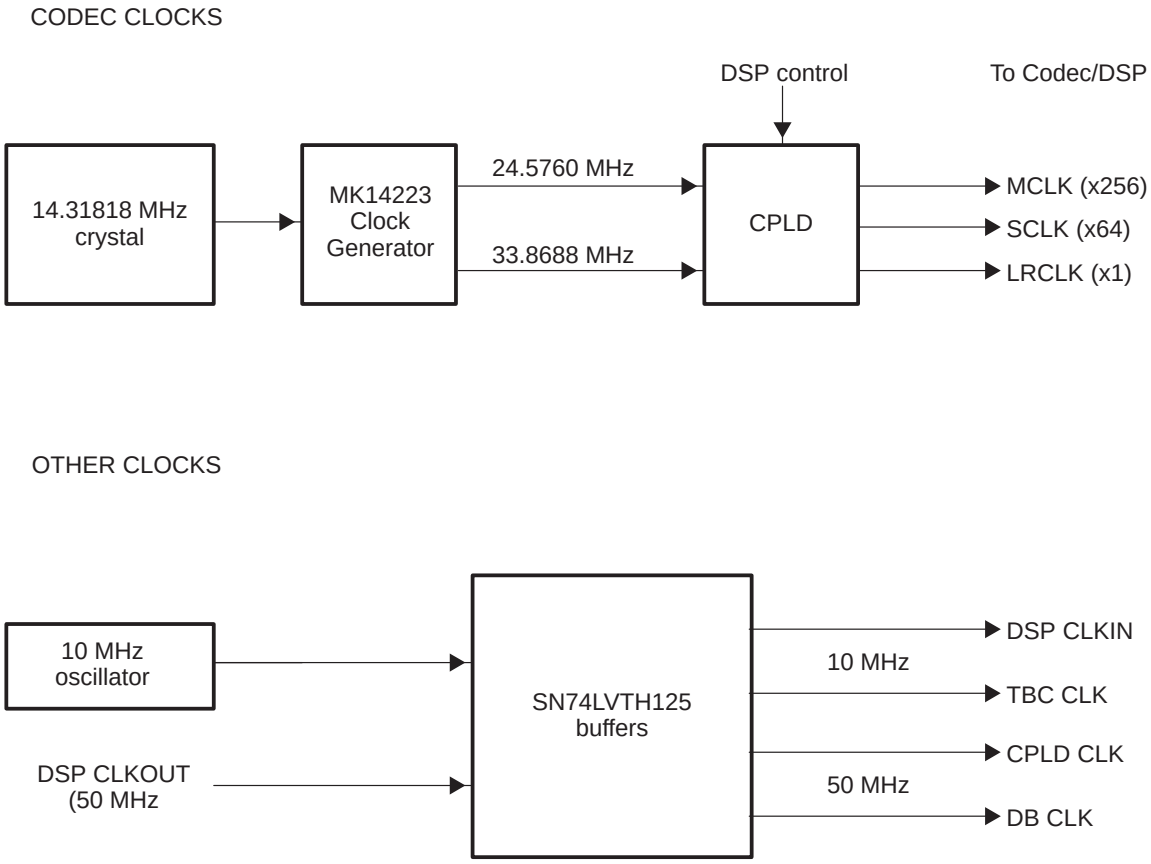
The PCI2040 can interrupt the host PC using $\overline{\text{INTA}}$ whenever certain events occur including HPI error, GP Bus error, GPIO2 interrupt, JTAG TBC interrupt, or a DSP-to-host interrupt generated by with the DSP's $\overline{\text{HINT}}$ output. The EVM's software support provides interrupt masking and handling capabilities. The host can interrupt the DSP by writing setting the DSPINT bit in the DSP's HPI control register (HPIC).

The PCI2040 HPI and GP bus interfaces are brought out to an AMP Mictor (2-767004-2) connector for observation. Note that this connector is not installed on the production EVM board. Additionally, the GPIO signals are provided on a jumper header, with an option for selecting their levels on another jumper header. Note that these two jumper headers are not installed on the production EVM board.

4.2 Clocks

The PCI2040 controller gets its clock directly from the PCI bus, but the audio codec, JTAG TBC, DSP, and CPLD clocks are provided on the EVM. Figure 4–2 summarizes the clock generation and distribution on the EVM.

Figure 4–2. EVM Clocks



The audio codec requires 3 different clocks including a master clock (MCLK) that runs at 256 times the sample rate, a serial clock (SCLK) that runs at 64 times the sample rate, and a left/right clock that runs at the sample rate. MCLK is used internal to the codec for filtering and oversampling. SCLK is used by both the codec and DSP to transmit and receive the serial data streams between the 2 devices. Data is transmitted on the rising edge and sampled on the falling edge of SCLK. The LRCLK is the frame sync signal that indicates when the first bit of the serial data streams is valid. It is active high for one SCLK period.

In order to support the range of common multimedia audio sample rates, two different source clocks are needed. The MK14223 clock generator device is designed specifically for this task. It accepts a common 14.31818 MHz crystal input and produces the 24.576 and 33.688 MHz clocks that are required as audio codec source clocks. Under DSP control, the CPLD generates the MCLK, SCLK, and LRCLK signals by selecting the appropriate source clock and dividing it down as necessary.

A single 10 MHz oscillator is used as the source clock for the DSP and TBC, and indirectly for the CPLD and expansion peripheral (daughterboard) interface. An SN74LVTH125 quad buffer provides 5-V to 3.3-V translation and provide an acceptable edge rate. The DSP and TBC are provided with 10 MHz buffered clocks. The DSP's on-chip PLL can select a DSP operating clock rate in the range of 2.5 to 150 MHz. Since a 100 MHz DSP is used, the software defaults the DSP's internal clock to 100 MHz by setting the on-chip PLL to x10 mode. The JTAG TBC runs at 10 MHz, so the JTAG serial data is transmitted and received at 10 Mbps.

The DSP's CLKOUT signal is buffered by the other two SN74LVTH125 buffers to provide buffered clocks to the CPLD and daughterboard. The CPLD does not currently use this clock, but it is available for future use. The clock is made available to daughterboards for synchronization to the DSP bus rate. This can be useful for state machine clocks and the generation of hardware wait states. The 5410 DSP can generate a CLKOUT signal at a rate of 1, 1/2, 1/3, or 1/4 of its internal clock rate. By default, the EVM support software selects a 1/2-rate CLKOUT signal, so at 100 MHz, the clocks to the CPLD and daughterboard are 50 MHz.

4.3 Logic

The EVM uses fast, discrete TTL logic and a CPLD to provide memory decoding and glue logic functions on the board.

An SN74LVC32A quad-OR device provides fast memory decoding for the TBC and external memory. The PCI2040 controller can directly interface to the TBC device with no logic required. However, since the PCI2040's GP bus is shared between the TBC and the CPLD, minimal logic is required to decode between TBC and CPLD accesses. Simple decoding is performed based on the most-significant address bit on the GP Bus (GPA5). TBC accesses occur whenever a read or write occurs when GPA5 is low. CPLD accesses occur whenever a read or write occurs when GPA5 is high. Two of the four OR gates provide qualified RD/WR strobes to the TBC. The other two OR gates generate memory read and write enables for the daughterboard interface that are compatible with the C6x DSP.

A low-voltage Cypress CY37128VP CPLD provides the following functions on the EVM:

- ❏ Reset generation to DSP, daughterboard, and JTAG TBC
- ❏ Host, DSP, and daughterboard interrupt control and status
- ❏ Host registers mapped in PC's memory space
- ❏ DSP registers mapped in DSP's I/O space
- ❏ Daughterboard data transceivers control
- ❏ Daughterboard control signal generation
- ❏ Codec clock generation

Reset signals can come from multiple sources including the PCI2040 controller, reset pushbutton, voltage supervisor, and under host and DSP software control. Logic manages all of these reset inputs and resets the appropriate devices.

There are also multiple interrupt sources from the host, DSP, and daughterboard that must be managed. Software can generate, enable, select, and poll various interrupts on the board by accessing CPLD registers.

The CPLD provides five, 8-bit registers that are mapped into the host's memory space starting at offset 0x80 of the GP bus memory region. The registers are aligned on 32-bit (doubleword) address boundaries (0x80, 0x84, … 0x90). These registers provide board control and status, CPLD revision information, and two hardware semaphores that can be used for resource sharing or synchronization between the host and the DSP. Chapter 6, *Host CPLD Registers*, provides details on the CPLD's host registers.

The CPLD provides eight, 8-bit registers that are mapped into the DSP's I/O space from 0x00 to 0x07. These registers provide board control and status, data memory control, daughterboard general-purpose I/O, codec control and

clock selection, and two hardware semaphores shared with the host. Chapter 7, *DSP CPLD Registers*, provides details on the CPLD's DSP registers.

The CPLD controls the daughterboard interface including the generation of data transceiver enables and strobes and memory interface control signals. In order for the expansion memory interface to be compatible with existing daughterboards designed for the C6x, differences between C54x and C6x memory interface signals have to be resolved. The CPLD and logic mentioned earlier combine to handle this resolution. A registered transceiver is used on the upper 16 data lines of the expansion memory interface, so the CPLD controls the 16-to-32-bit and 32-to-16-bit conversions required for memory accesses to 32-bit daughterboards.

The CPLD also generates the clocks required by the audio codec and DSP to run the codec device and pass data between them. The CPLD includes multiple clock divisor circuits that provide 50% duty cycle clocks required for codec operation. Refer to the register definitions in Chapter 7, *DSP CPLD Registers*, for details on how the CPLD clocks are generated under DSP control.

4.4 JTAG Emulation

The EVM provides embedded JTAG emulation that is accessible via the PCI bus, as well as support for an external XDS510 emulator. The selected JTAG method is user-configurable via the DIP switches when the board is operated outside the PC, or via the software switches when it is in the PC.

The TI SN74ACT8990 JTAG test bus controller (TBC) provides memory-mapped control of the DSP's JTAG interface. This allows the C source debugger to be used with the EVM without an external emulator.

The EVMs embedded emulation support provides several benefits:

- Emulation is supported without external cabling, monitor software, or consumption of user resources.
- Easy access to the DSP supports high-level language (HLL) debuggers, factory testing, and field diagnostics.
- System boot ROMs are not needed. The host can download all necessary program and data through the emulation port.

The TBC is presented to the PC host software as 24 memory-mapped registers. Each register is mapped at 32-bit (doubleword) address boundaries, but only the lower 16 bits of data words are connected to the 16-bit TBC device. The PCI2040 controller's GP bus accesses the TBC device's 24 registers in the lower half of the 256-byte GP bus memory region (0x00, 0x04, … 0x5C).

A 14-pin (2 rows of 7 pins) header is included on the EVM to support an external XDS510 or XDS510WS emulator connection. This connection is required to use the EVM outside the PCI and may be used for debugging the EVM in the PC with the XDS510. Two TI CBT quad 2:1 multiplexer devices (SN74CBT3257) provide the +5-V to +3.3-V translation required from the TBC to the DSP, and the selection between internal and external JTAG emulation.

4.5 Digital Signal Processor

The 100 MHz TMS320VC5410 DSP is the processor used on the EVM. The 5410 provides a set of enhanced peripherals and functionality that were not available on previous members of the 54x family. The following is a list of key features of the 5410 DSP that differentiate it from previous 54x DSPs:

- 64K x 16-bit on-chip RAM
- Enhanced HPI-8 with support for host access to all on-chip memory
- Three multichannel buffered serial ports (McBSPs)
- Six-channel direct memory access (DMA)
- CLKOUT selection of 1, 1/2, 1/3, or 1/4 of the internal clock rate
- Extended addressing of up to 8M x 16-bit memory

The 5410's ability to address a large address space and its three McBSP serial ports make it an ideal processor for this EVM since it can be compatible with C6x daughterboards.

The DSP can run in a variety of modes. The EVM hardware and software support provides you with the ability to use them. DIP switches can select the clock mode (CLKMD) and boot mode (MP/MC). A jumper is provided to select between booting from the HPI or from a daughterboard. DSP software library support provides software control over various operating parameters such as clock speed, bus rate, wait states, and memory map.

The PCI2040 controller interfaces directly to the DSP's HPI for data transfers and interrupt control. The host has access to all of the DSP's on-chip memory via the HPI. The DSP's JTAG interface can be connected directly to an XDS510 emulator or to the onboard JTAG TBC under user control for debugging.

The DSP's three McBSPs are allocated on the EVM as indicated in Table 4–2. The 5410 bootloader supports McBSP bootloading through McBSPs 0 and 2, so McBSP 1 was selected as the dedicated serial port for the audio codec.

Table 4–2. DSPs McBSP Allocation on EVM

McBSP No.	Function
0	Expansion peripheral interface (McBSP0)
1	AD77 audio codec
2	Expansion peripheral interface (McBSP1)

The DSP has four external maskable interrupts ($\overline{\text{INT3}}$–$\overline{\text{INT0}}$) and one non-maskable interrupt ($\overline{\text{NMI}}$). $\overline{\text{INT2}}$ is used at boot time to indicate whether HPI booting should be performed. If $\overline{\text{INT2}}$ is connected to $\overline{\text{HINT}}$ at boot time, then HPI boot is selected. A jumper is provided to make this selection. $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ are routed to the expansion peripheral interface, so a daughterboard can interrupt the DSP with these two interrupts. $\overline{\text{INT3}}$ can be generated by

either the host or the daughterboard under DSP software selection. A $\overline{\text{NMI}}$ interrupt can be generated by the host to the DSP. The DSP can mask this interrupt inside of the CPLD (it is not maskable in the DSP). Table 4–3 summarizes the DSP interrupt allocation.

Table 4–3. DSPs Interrupt Allocation

Interrupt	Allocation
$\overline{\text{NMI}}$	Host-generated, DSP-maskable in CPLD
$\overline{\text{INT3}}$	Host or daughterboard-generated, DSP selects source in CPLD. Both sources can also be polled in CPLD.
$\overline{\text{INT2}}$	HPI boot select or daughterboard-generated (if not in HPI boot mode)
$\overline{\text{INT1}}$	Daughterboard-generated
$\overline{\text{INT0}}$	Daughterboard-generated

The DSPs $\overline{\text{HOLD}}$ and $\overline{\text{HOLDA}}$ are not used on the EVM. $\overline{\text{HOLD}}$ is pulled high.

The DSPs CLKIN is provided by a buffered 10 MHz oscillator. Software can control the frequency of the internal DSP (CPU) clock using the on-chip, programmable PLL to obtain internal clock rates from 2.5 to 150 MHz. The DSP's CLKOUT can be selected by the DSP software to be 1, 1/2, 1/3, or 1/4 of the internal DSP clock. All external memory accesses are relative to the CLKOUT frequency. The CLKOUT signal is provided to the CPLD and the expansion peripheral interface for daughterboard usage.

The DSPs timer output TOUT, and general-purpose input ($\overline{\text{BIO}}$) and output (XF) are routed to the expansion peripheral connector for daughterboard usage.

4.6 External Memory

The EVM extends the 5410's 64k x 16 on-chip RAM with external memory consisting of 64k x 16 program memory and 64k x 16 data memory. Both of these memories could be upgraded to 256k x 16-bit devices since the layout supports them. The memories are 10-ns devices which support consecutive (CONSEC/ = 0) 50 MHz accesses and nonconsecutive 100 MHz accesses at 100 MHz with 0 wait states.

Page 0 of the external program memory can be mapped into the DSP's program memory space from 0x0000–0xBFFF if the DSP's OVLY bit is set to 0. If OVLY = 1, then external memory in Page 0 is only available from 0x8000–0xBFFF. The DSP's 16K x 16 ROM is always resident in the program memory space from 0xC000 to 0xFFFF. Extended pages of program memory access the external program memory in 64K-word pages if OVLY = 0 or 32K-word pages if OVLY = 1. Refer to the TMS320VC5410 data sheet for detailed memory maps. Onboard external program memory is accessed only when A22 is low. The daughterboard external program memory space is accessed when A22 is high.

The 5410 addresses external data memory only when the DSPs DROM bit is 0. External data memory is accessible only in a 32K x 16 window in data memory from 0x8000x–0xFFFF since the DSP does not have built-in extended data memory page support. However, the EVM's CPLD provides a data memory page selection register that supports extension of the data memory space to thirty-two 32K x 16 windows, for a total address space of 1M data memory words. Each window resides at 0x8000 in the data memory space. The EVM ships with a 64K x 16 memory device, so there are two external data memory pages available. The CPLD provides an additional data memory select (DMSEL) bit that determines if external data memory accesses are intended for the onboard memory or the daughterboard. When DMSEL = 0, onboard memory is accessed. When DMSEL = 1, daughterboard memory is accessed. The CPLD defaults DMSEL to 1 to support daughterboard parallel booting at reset.

4.7 Stereo Audio Interface

The EVM includes a stereo audio interface that has the following characteristics:

- ❑ 16-bit stereo audio codec with sample rates from 8 kHz to 96 kHz
- ❑ Support for stereo electric microphones with voltage bias
- ❑ Stereo line-level input
- ❑ Stereo line-level output
- ❑ Three 3.5-mm (1/8-inch) stereo audio jacks

The TI TLC320AD77C stereo audio codec is a 24-bit codec which operates in 16-bit DSP mode. It is a delta-sigma codec that is designed for consumer applications that demand excellent audio performance. It is typically used in applications such a mini-disk players, audio/video receivers, musical instruments and other end-equipment requiring high-performance digital audio conversion.

The AD77 codec requires three clocks to be provided to it consisting of MCLK (x256), SCLK (x64) and LRCLK (x1), as described in detail in Section 4.2, *Clocks*. The conversion rate is defined by MCLK, the serial port shift clock is defined by SCLK, and the left/right frame synchronization is defined by LRCLK. As discussed earlier, the CPLD provides the necessary clocks to the codec. Additionally, the DSP software can enable and disable both the codec clocks and the codec itself with CPLD register bits.

The EVM provides support for the following popular sample rates (in kHz):

8.000	9.600	11.025	12.000	13.230	16.000
16.5375	22.050	24.000	32.000	33.075	44.100
48.000	66.150	96.000			

The AD77 codec is connected to the DSP's McBSP1. Since clocks are provided by the DSP, the McBSP should be initialized for external clocks and frame syncs, with a data delay of 1. EVM support software provides DSP library functions that initialize and control the codec.

The industry-standard 3.5-mm stereo audio jacks are located on the board's mounting bracket. The tip of each jack is the left audio channel and the ring of each jack is the right audio channel. Mono connections only use the left channel.

TI TLV2462C op amps are used for the microphone preamplifier and filtering. A voltage bias is provided to support the use of battery-powered and electric microphones. Passive filtering is included on the EVM between the audio jacks and the codec for increased performance. Ferrite beads and capacitors are used on all audio interfaces to reduce noise coupling.

The microphone input is designed for electric microphones that require a bias voltage. A dynamic microphone can be used if capacitors block the bias

voltage. The maximum allowable signal level from the microphone is 500 mV (350 mVrms). There is a 10-dB preamplifier gain on the EVM.

The line input signal level can be a maximum of 6 Vpp (2 Vrms). The line input signal is attenuated 9 dB before it is presented to the codec. A jumper option can disable the 9 dB attenuation for maximum dynamic range with 1 Vrms or less signals.

The line output signal level can be a maximum of 1.9 Vpp (0.67 Vrms). The line output is designed to drive 10-kohm loads, not low-impedance headphones or speakers.

The DSP software can select either the microphone or line-level input for each of the channels (left/right) by controlling bits in a CPLD register. There are additional control bits to enable codec de-emphasis for 32, 44.1, and 48 kHz sample rates and the codec speed mode for greater than 48 kHz sample rates. Refer to Section 7.5 for details on the codec control register bits.

Additional filtering is required for 8 kHz operation due to the nature of the AD77's DAC output at this lower sample rate. The AD77 is specified to operate from 16 to 96 kHz. However, if the additional external filtering is enabled using a jumper option, then 8 kHz operation can be used without added noise from the DAC.

4.8 Expansion Interfaces

The EVM provides two expansion connectors that allow a daughterboard to be connected to extend the capabilities of the board. Daughterboards can extend the capabilities of the EVM and provide custom and application-specific I/O. One expansion connector provides the DSP's memory interface and the other provides access to the DSPs peripherals and control/status signals. Both connectors also provide power to the daughterboard.

Most of the expansion connector signals are buffered so that the daughterboard cannot directly influence the operation of the EVM board. The use of TI 'LVTH and 'CBT interface devices allows the use of either 5- or 3.3-V devices to be used on the daughterboard.

Mating connectors for the 80-pin connectors on the EVM are available in four different heights; all of which meet the PCI height requirement. The recommended mating connector provides 0.465" spacing allowing sufficient space for components.

The expansion memory interface connector has a reference designator on the EVM of J7. The expansion peripheral interface connector is J8. Refer to Chapter 8 for the pinouts of the expansion connectors.

4.8.1 Expansion Memory Interface

The expansion memory interface provides the DSP's memory interface signals to a daughterboard. External asynchronous memories and memory-mapped devices can be added to the EVM, including nonvolatile memory that can be used to boot the EVM upon reset.

The expansion memory interface includes:

- Twenty-two external address signals (X_A21–X_A0). All of the DSPs external address signals, except A22, are available on the expansion memory interface, allowing up to 4M words of external program memory to be addressed. A22 is not provided since daughterboard program memory accesses are only valid when A22 is high. Similarly, daughterboard I/O accesses are only valid when A15 is high (1M I/O addresses), and data memory accesses are only valid when DMSEL is high (1M addresses). Table 4–4 summarizes the address ranges for daughterboard accesses from the DSP.

Table 4–4. Daughterboard Access Address Ranges

Access	Address Range	Conditions
Program	0x400000 – 0x7FFFFF	A22 is high, DSPs XPC selects page
Data	0x8000 – 0xFFFF	DMSEL=1, DM_PG[4..0] selects 1 of 32, 32k pages
I/O	0x8000 – 0xFFFF	A15 is high, DM_PG[4..0] selects 1 of 32, 32k pages

- Thirty-two external data signals (X_D31–X_D0). All of the DSP's 16 external data signals, plus 16 additional data signals for 32-bit daughterboard compatibility, are available on the expansion memory interface.
- Data memory space enable ($\overline{\text{X\_DS}}$). The DSP's data memory space enable is asserted only when DMSEL = 1 and the DSP asserts $\overline{\text{DS}}$. This qualifies daughterboard data memory space accesses.
- Five asynchronous memory control signals. Asynchronous memory control signals that are from the DSP, or derived from DSP signals to provide C6x daughterboard compatibility, are available on the expansion memory interface. The signals include $\overline{\text{X\_RE}}$, $\overline{\text{X\_OE}}$, $\overline{\text{X\_MSTRB}}$, $\overline{\text{X\_WE}}$ and X_RDY. X_RDY can be used by the daughterboard to control hardware wait states.
- Power Signals. The expansion memory interface also provides ground, 5-V, and 3.3-V power signals to the daughterboard.

4.8.2 Expansion Peripheral Interface

The expansion peripheral interface provides the DSP's peripheral signals to a daughterboard. This peripheral expansion capability allows serial devices such as other codecs and communication devices to be added to the EVM via a daughterboard.

The expansion peripheral interface includes:

- Six signals for each serial port. The DSP's six McBSP signals for McBSP0 and McBSP2 are available on the expansion peripheral interface (mapped to serial ports 0 and 1, respectively). These signals are buffered on the EVM to support both 5-V and 3.3-V serial devices. Each McBSP has CLKX, FSX, DX, XLKR, FSR, and DR signals.
- Timer output signal. The DSP's timer output signal TOUT is available on the expansion peripheral interface.
- Interrupts and interrupt acknowledge. The four external DSP interrupts ($\overline{\text{INT0}}$–$\overline{\text{INT3}}$), a DSP-to-daughterboard interrupt ($\overline{\text{X\_DBINT}}$), and the DSP's interrupt acknowledge ($\overline{\text{IACK}}$) are available on the expansion peripheral interface. Note that $\overline{\text{INT2}}$ is only available if it is not used on the EVM as the HPI boot indicator.
- DSP general-purpose I/O signals. The DSP's XF and $\overline{\text{BIO}}$ signals are available on the expansion peripheral connector.
- Three asynchronous memory control signals. Asynchronous memory control signals that are from the DSP are available on the expansion memory interface. The signals include $\overline{\text{X\_PS}}$, $\overline{\text{X\_IS}}$, and $\overline{\text{X\_IOSTRB}}$.
- Four general-purpose input/output flags. Four signals provided by the EVM's CPLD can be individually programmed as inputs or outputs. By default, for compatibility with C6x EVM daughterboards, two signals

default as inputs (X_STAT1 and X_STAT0) and two signals default as outputs (X_CNTL1 and X_CNTL0). However, for C54x daughterboards, these signals can be inputs or outputs.

- Reset signal. The expansion peripheral interface also provides a reset signal that is active low when the board is in the reset state. This allows the circuitry on the daughterboard to be set to a known state. The reset signal is asserted for a minimum of 140 milliseconds upon power-up, via a manual reset pushbutton or under software control. A CPLD register bit (DB_RST) allows DSP software to directly control this reset signal when the DSP is not held in reset.

- CLKOUT signal for the synchronization clock. The DSP's CLKOUT signal (CPU clock divided by 1, 2, 3, or 4) is brought out to the expansion peripheral interface for synchronization needs on daughterboards.

- Daughterboard detect. Pin 75 on the expansion peripheral connector should be grounded on the daughterboard to indicate that the daughterboard is present.

- Two miscellaneous signals. The DSP's $\overline{\text{MSC}}$ and $\overline{\text{IAQ}}$ signals are also provided on the expansion peripheral interface.

- Power signals. The expansion peripheral interface also provides ground, 12-V, –12-V, 5-V, and 3.3-V power signals to the daughterboard.

4.8.3 Daughterboard

The EVM supports the mating of a daughterboard that has two 80-pin 0.050”x0.050” TFM-series connectors from Samtec. The recommended mating connector, whose part number is TFM-140-32-S-D-LC, is a surface-mount connector that provides a 0.465” mated height.

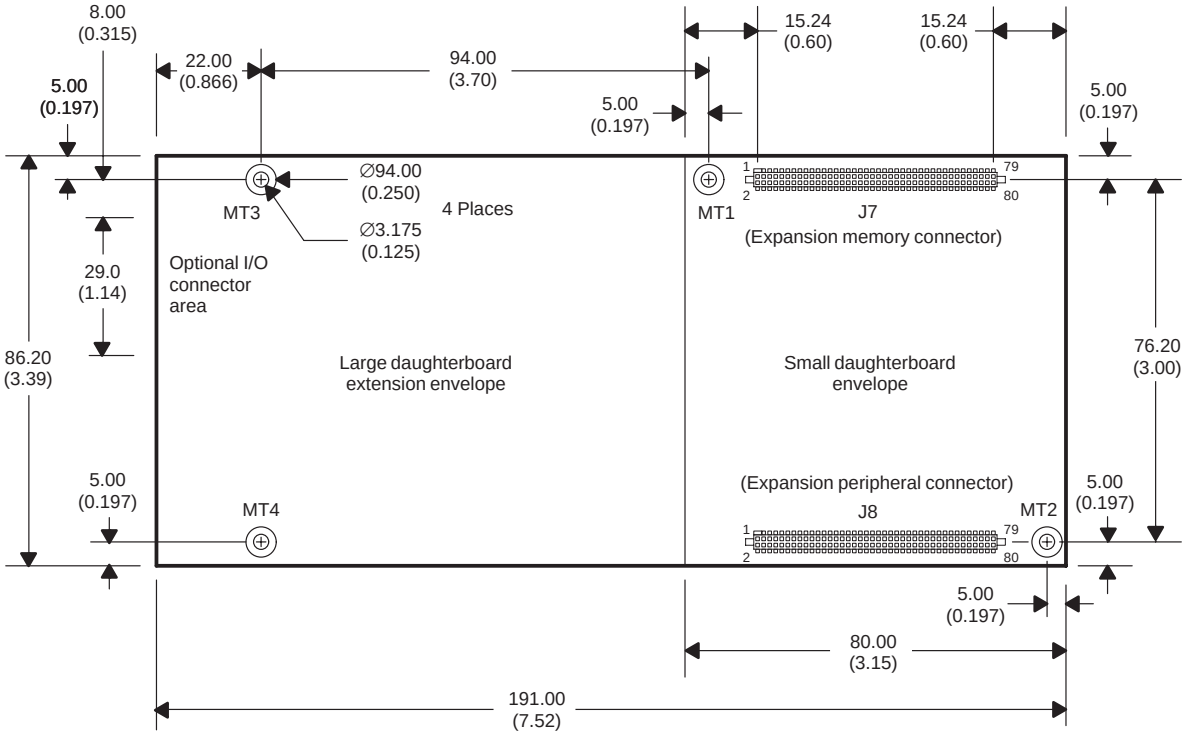
The EVM supports two sizes of daughterboards which both use the two 80-pin connectors. The small daughterboard measures approximately 3.15” long and 3.4” wide and mounts in the center of the board over the low-profile buffers and memories. This format is intended for daughterboards that do not require an I/O connection on the mounting bracket.

The large daughterboard measures approximately 7.5” long and 3.4” wide and mounts from the center of the board over to the mating connector end of the board. This format is intended for daughterboards that require an I/O connection on the mounting bracket or need more space than the small daughterboard provides.

Daughterboards mount with their primary component sides down. This ensures that the PCI height requirement is met and no components are exposed to possible damage due to board insertions and extractions.

Figure 4–3 shows the location of the expansion connectors and the dimensions of the EVM daughterboards. This figure shows the component side of the EVM.

Figure 4–3. Daughterboard Envelopes and Connections on EVM



Notes:

1. All dimensions are shown in millimeters. Inch dimensions are shown in parentheses.
2. Drawing shows daughterboard envelopes and connections on components side of EVM board.
3. Standard size daughterboard is 80.0 x 86.2 mm (3.15 x 3.39”).
4. Full–size daughterboard is 191.0 x 86.2 mm (7.52 x3.39”).
5. Daughterboard connectors are Samtec .050 x .050” Micro Strips (SFM–140–L2–S–D–LC).
6. Daughterboard mating connectors are Samtec TFM–140–32–S–D–LC.
7. Mating height is 0.465” (11.81 mm).
8. There are four plated holes (MT1–MT4) on the EVM for standoff mounting.
9. Mounting holes MT1–MT3 are electrically connected to digital ground
10. Mounting hole MT4 is floated with no electrical connection.

4.8.4 Compatibility with C6x EVM Daughterboards

The PCI2040 EVM’s expansion interfaces are designed to be compatible with the C6x EVM daughterboards. This is advantageous because it allows these daughterboards to be used on this EVM also. This presents a few challenges, since the C6x is a 32-bit processor with a different memory interface. However, with some additional logic and a registered transceiver on the EVM, the PCI2040’s EVM should be compatible with virtually all existing daughterboards.

32-bit memory accesses from the 16-bit 5410 DSP are accomplished in software by making two accesses to the least and most significant words. In hardware, this involves the use of a 16-bit registered transceiver and control logic in the CPLD.

DSP software must enable the 32-bit access mode by setting the DB_WIDE bit in the CPLD and setting the DB_32ODD bit for the appropriate target address (0 = even, 1 = odd). For reads, the LSW is read first, followed by the MSW. For writes, the MSW is written first, followed by the LSW. Accesses to the LSW cause the memory access to take place with a 32-bit daughterboard.

The C6x has four byte enables since it can access individual bytes. The C54x does not support individual byte accesses. This shouldn't be a problem with most daughterboards. The PCI2040 EVM does ground BE1 and BE0 so that they are always enabled. BE3 and BE2 are mapped to the DSP's upper address lines A21 and A20 which are always zero for memory accesses to C6x daughterboards. Therefore, all four bytes enables are enabled when accessing C6x daughterboards.

The C6x DSP has different memory interface signals than the C54x DSP. Since the daughterboard interface is based on the C6x memory interface, some simple logic is included on the EVM to convert C54x memory control signals to C6x memory control signals for compatibility. The signals that are created include $\overline{\text{X\_OE}}$, $\overline{\text{X\_RE}}$, and $\overline{\text{X\_WE}}$. Additionally, the C54x MSTRB signal is provided on a pin that is a no-connect on C6x daughterboards.

The C6x DSP has four chip enables ($\overline{\text{CE0}}$–$\overline{\text{CE3}}$). C6x daughterboards are provided with $\overline{\text{CE1}}$, $\overline{\text{CE2}}$, and $\overline{\text{CE3}}$. The C54x does not provide chip enables, but it does provide memory space enables. Therefore, the C6x enables were mapped to the DSP's $\overline{\text{DS}}$, $\overline{\text{PS}}$, and $\overline{\text{IS}}$ signals. Since $\overline{\text{CE1}}$ is the default space for C6x daughterboards, the 54x's data memory space is the default space, so it can support bootloading from a daughterboard.

The C5410 does not provide the McBSP signal CLKS, so it is not available on the daughterboard interface. These signals are no connects on the C54x EVM. Since the C54x EVM still provides the CLKX and CLKR signals, independent transmit and receive clocks are available at the daughterboard, so this should meet most daughterboards' requirements.

The C6x has two timers, with each timer having an input and an output. The C5410 only has one timer output. The C54x EVM has a no connect for the first timer input. It places the software-controlled DB_INT signal on the timer 1 output and the DSP's $\overline{\text{BIO}}$ signal on the timer 1 input pins.

The C54x provides all four external DSP interrupt signals ($\overline{\text{INT0}}$–$\overline{\text{INT3}}$). The C6x EVM only provided a single interrupt on the pin where $\overline{\text{INT0}}$ is assigned. Therefore, the PCI2040 EVM has a superset of interrupts on pins that were no-connects on the C6x EVM. Note that these interrupts are compatible with TI's new daughterboard specification that allocates these extra pins to additional interrupts.

The C54x does not have interrupt identification codes like the C6x (INUM0–INUM3). Therefore, these pins provide C54x signals including $\overline{\text{IACK}}$, $\overline{\text{MSC}}$, and $\overline{\text{IOSTRB}}$. The interrupt ID codes are not used on any existing C6x daughterboards, so this shouldn't be a problem.

The C54x XF output signal is assigned the same pin as the PD pin on the C6x EVM. Therefore, if a daughterboard supports power down mode, then the XF signal can be used to assert this mode (XF=1).

The C6x EVM defines four I/O signals (two inputs and two outputs). The PCI2040 EVM defaults to this same arrangement but provides the additional capability to assign the direction to each of the four signals as either input or output.

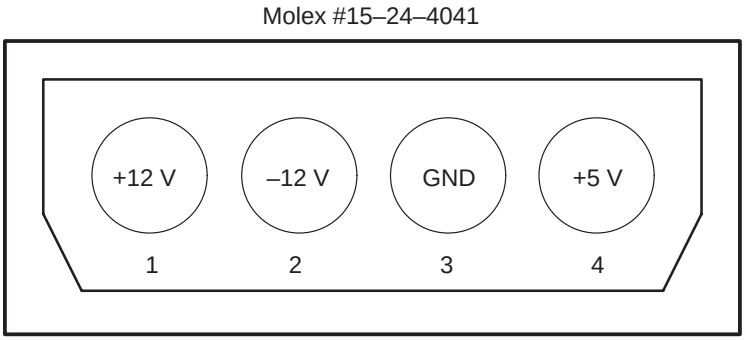
The PCI2040 EVM defines an additional signal $\overline{\text{DB\_DET}}$ which indicates to the DSP that a daughterboard is attached. This pin is a ground on C6x daughterboards, so it is directly compatible.

In summary, the PCI2040 EVM daughterboard interface provides maximum compatibility with C6x daughterboards. However, there are some minor differences identified above that should be considered.

4.9 Power Supplies

The EVM only requires a single 5-V input power supply. It generates the 2.5-V DSP core, 3.3-V digital, and 3.3-V analog voltages with onboard linear voltage regulators. When the EVM is installed in a PCI slot, 5 V is provided from the PCI bus. Also, the PCI bus provides 12 V and –12 V which are routed to the expansion peripheral interface for daughterboard use, but they are not used on the EVM board itself. When the EVM is operated standalone, power is provided through a 4-pin Molex power connector as shown in Figure 4–4. A standard disk-drive power cable can be used to provide the 5 V and 12 V. Some power supplies also provide the –12 V if needed for a daughterboard.

Figure 4–4. External Power Connector



Note: Not to scale.

A 3.3-V linear regulator provides up to 1.5 A of current to the EVM board circuitry and daughterboard. The daughterboard interface is specified up to 1 A of current on the 3.3 V. If additional current is required by a custom daughterboard, then the D^{2}PAK regulator (U34) can be replaced with drop-in Micrel devices that can provide 3, 5, and 7 A. The EVM layout supports the use of an Aavid Thermal Products surface-mount heat sink (PN 573300) on the D^{2}PAK device, if required, for these higher current devices.

4.10 User Options and Indicators

The EVM has user options that are selected on the board via DIP switches and jumpers. Several LEDs on the EVM provide board and PCI controller status information.

The EVM includes an 8-position DIP switch (SW1–1 through SW1–8) that allows selection of user options. The user options include JTAG selection, DSP mode, clock mode, hot swap, and user-defined options.

Table 4–5 summarizes the eight DIP switches and their functions in both the ON and OFF settings. Table 4–6 summarizes the clock mode selections. The factory default switch settings are all switches in the ON position as indicated in the table.

Table 4–5. User Option DIP Switches

Switch	Name	ON Selection	OFF Selection
SW1–1	JTAGSEL	Onboard JTAG TBC (default)	External XDS510 emulator
SW1–2	MP/MC	Microcomputer mode (default)	Microprocessor mode
SW1–3	CLKMD3	See Table 4–6	See Table 4–6
SW1–4	CLKMD2	See Table 4–6	See Table 4–6
SW1–5	CLKMD1	See Table 4–6	See Table 4–6
SW1–6	HOTSWAP	Ejector handle closed (default)	Ejector handled opened
SW1–7	USER1	Read as 1 in CPLD (default)	Read as 0 in CPLD
SW1–8	USER0	Read as 1 in CPLD (default)	Read as 0 in CPLD

Table 4–6. Clock Mode Switch Selections

CLKMD3 (SW1–3)	CLKMD2 (SW1–4)	CLKMD1 (SW1–5)	Clock Mode
ON (default)	ON (default)	ON (default)	Divide-by-2, external source
OFF	ON	ON	Divide–by-2, external source
ON	OFF	ON	Divide-by–2, external source
OFF	OFF	ON	Stop mode
ON	ON	OFF	Divide-by-2, internal oscillator
OFF	ON	OFF	PLLx1, external source
ON	OFF	OFF	Divide-by-2, external source
OFF	OFF	OFF	Reserved

The EVM also has some installed jumper options that select line input attenuation for the left/right channels, line output filtering for left/right channels, and DSP boot selection. Table 4–7 summarizes the jumper options on the

EVM board, indicating the selection when the jumper is in the 1–2 and 2–3 positions.

Table 4–7. EVM Jumper Options

Jumper	Name	1–2 Selection	2–3 Selection
JP1	Left line in attenuation	–9 dB (2 Vrms max) (default)	0 dB (1 Vrms max)
JP2	Right line in attenuation	–9 dB (2 Vrms max) (default)	0 dB (1 Vrms max)
JP3	Left line out filter	Bypass (default)	Enable
JP4	Right line out filter	Bypass (default)	Enable
JP5	DSP boot selection	HPI (default)	Daughterboard

Note that the line out filters are only required for 8 kHz operation.

The EVM has six LEDs on it to provide indications of board and PCI controller status. Table 4–8 summarizes the user indicators on the EVM board.

Table 4–8. EVM User Indicators

Ref Des	Name	LED Color	Location	On Meaning	Off Meaning
D1	USER LED	Red	Bracket	CPLD bit = 1	CPLD bit = 0
D2	POWER	Green	Top card edge	Power on	Power off
D3	D3STAT	Red	Top card edge	D3 state	Not D3 state
D4	PME	Yellow	Top card edge	PM event	No PM event
D5	ENUM	Green	Top card edge	HS event	No HS event
D6	HOTSWAP	Red	Top card edge	HS-okay	HS-not yet

Chapter 5

EVM Confidence Test

Topic Page

5.1 EVM Confidence Test

This confidence test is a command line utility that provides a way to test the PCI2040 EVM board, enabling the user to verify proper installation and operation of the board. The testing includes checkout of the PCI2040 PCI bridge, DSP, external memory, and audio codec. This automated utility provides pass/fail indications for each of these items. This utility provides the user with confidence that the board is working properly.

This test must be executed from the directory into which it was installed since several support files are required for the test operations.

The board confidence test utility is invoked with the following syntax:

```
evmtest [options]

   evmtest  Command that invokes the board confidence
   test utility

   options  Options that affect the way the utility be-
   haves. Options are not case-sensitive and can appear
   anywhere on the command line following the command.
```

If no options are entered, the utility continues to run.

Table 5–1 summarizes the DSP application loader utility command parameters. Command parameters can be used in any order on the command line.

Table 5–1. Board Confidence Test Utility Command Parameters

Command Parameter	Description
Log_filename	Name of optional file for test results logging.
–? or –h	Utility help display
–b *num*	Selects specific EVM target board; *num* is zero-based relative board index, ranging from 0 (first board) to n–1 (last board), where n boards are installed. Default is 0.
–d	Selects specific DSP on the board.
–i	Enables information only mode. In this mode, only the board configuration information is displayed, bypassing the board tests.

The confidence test checks for proper installation of the Win32 DLL and low-level driver by opening a connection to the board. With successful access to the board, the board's configuration information is retrieved and displayed to the command window. If the information only mode is requested (–i option), then the program terminates at this time.

The configuration information includes the:

- ❑ Board index
- ❑ Board type and revision

The tests performed on the board include:

- Memory tests
- ‘C54x interrupts tests
- Audio codec
  - Left channel tone test
  - Right channel tone test
  - Left/right channel tone test
  - Line in loopback test
  - Microphone in loopback test
- CPLD semaphores
- PCI Controller and Board LEDs
- JTAG

Upon completion of the tests, the board is reset. The configuration and test information are written to stdout and optionally recorded to a file specified on the command line.

Chapter 6

Host CPLD Registers

Five host CPLD registers are mapped into the PCI2040's GP bus address space starting at offset 0x80 in the GP bus region. The base address of this region (BAR2) is stored in the PCI2040's PCI configuration space at offset 0x18.

For consistency and ease of programmer usage, register bits are active high, meaning that the intended action is always selected by setting a bit to a 1. If the output signal is active low, then this polarity change is automatically handled in the CPLD.

Topic **Page**

Table 6–1 shows the five registers' bit definitions.

Table 6–1. Host CPLD Register Definitions

Address	Name	Description	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BAR+0x80	CNTL	Misc control	BRD_RST RW 0 (no reset)	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	TBC_RST RW 0 (no reset)	DSP_NMI RW 0 (no reset)	DSP_INT3 RW 0 (no int)
BAR2+0x84	STAT	Misc status	DB_DET R –	VCC2BAD R –	TBC_RDY R –	HINT2 R –	MP_MC R –	USR_LED R –	USR_SW1 R –	USR_SW0 R –
BAR2+0x88	REV	CPLD rev	CREV7 R –	CREV6 4 –	CREV5 R –	CREV4 R –	CREV3 R –	CREV2 R –	CREV1 R –	CREV0 R –
BAR2+0x8C	SEM0	Host Semaphore 0	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	SEM0 RW 0 (not own)
BAR2+0x90	SEM1	Host Semaphore 1	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	SEM1 RW 9 (not own)

6.1 Host CNTL Register (Offset 0x80)

The CNTL register controls the board hardware reset, the JTAG TBC's hardware reset, and the DSP's NMI and INT3 interrupts. By default, the TBC will not be forced in reset, and the interrupts will not be asserted. The TBC is always reset by the voltage supervisor for 200 ms at power up. Note that the DSP can disable the NMI in a DSP CPLD register. The DSP's INT3 can also be selected to be sourced from the daughterboard under DSP software control (see Sections 7.1 and 7.2). When the BRD_RST bit is set, the onboard reset circuitry will reset the board, including the CPLD itself, so the BRD_RST bit will be automatically cleared by hardware. A DSP interrupt is asserted by changing the respective interrupt register bit from low to high (rising edge). Table 6–2 summarizes the function of each bit in the CNTL register.

Table 6–2. Host CNTL Register Bit Definitions

Bit	Name	RW	Description
7	BRD_RST	RW	Board hardware reset 0 = no reset (default) 1 = assert board reset
6–3	RSVD	R	Reserved. Bits 6–3 return 0s when read.
2	TBC_RST	RW	TBC hardware reset 0 = no TBC reset (default) 1 = assert TBC reset
1	DSP_NMI	RW	DSP NMI interrupt 0 = no NMI (default) 1 = assert NMI
0	DSP_INT3	RW	DSP INT3 interrupt 0 = no INT3 (default) 1 = assert INT3

6.2 Host STAT Register (Offset 0x84)

The STAT register provides information about the board's state. This information includes the status of the daughterboard, DSP-to-Host interrupt (HINT2), the 2.5 voltage regulator status, the DSP operating mode, and the current user LED and switch settings. DB_DET indicates to the host software that a daughterboard is installed. VCC2BAD is set when the DSP's core voltage is out of tolerance. TBC_RDY indicates when the JTAG TBC device is ready to accept another access from the host. The HINT2 bit can be used as a general-purpose flag, or for interrupt polling. MP_MC bit indicates to the host software how the memory map is organized. The USR_LED and USR_SW1/USR_SW0 switch settings can be used to monitor the current application or board status. Table 6–3 summarizes the function of each bit in the STAT register.

Table 6–3. Host STAT Register Bit Definitions

Bit	Name	RW	Description
7	DB_DET	R	Daughterboard detect 1 = installed 0 = not installed
6	VCC2BAD	R	DSP core voltage bad indication 0 = okay 1 = bad
5	TBC_RDY	R	JTAG TBC ready indication 0 = not ready 1 = ready
4	HINT2	R	DSP-to-host interrupt status 0 = no interrupt 1 = interrupt
3	MP_MC	R	DSP operating mode 0 = microcontroller 1 = microprocessor
2	USR_LED	R	User LED state 0 = extinguished 1 = illuminated
1	USR_SW1	R	User switch 1 state 0 = off 1 = on. On position is down.
0	USR_SW0	R	User switch 0 state 0 = off 1 = on. On position is down.

6.3 Host REV Register (Offset 0x88)

The REV register simply provides the revision number of the CPLD. An eight-bit value represents revision numbers from 0 to 255. Table 6–4 summarizes the function of each bit in the STAT register.

Table 6–4. Host REV Register Bit Definitions

Bit	Name	RW	Description
7	CREV7	R	CPLD revision bit 7
6	CREV6	R	CPLD revision bit 6
5	CREV5	R	CPLD revision bit 5
4	CREV4	R	CPLD revision bit 4
3	CREV3	R	CPLD revision bit 3
2	CREV2	R	CPLD revision bit 2
1	CREV1	R	CPLD revision bit 1
0	CREV0	R	CPLD revision bit 0

6.4 Host SEM0 Register (Offset 0x8C)

The SEM0 register provides a hardware semaphore that is shared with the DSP SEM0. This semaphore can be used by host and DSP applications to control access to resources such as shared memory. To request the semaphore a 1 should be written to the SEM0 bit. The SEM0 bit should then be read to determine if the semaphore is owned (SEM0 = 1). If SEM0 is 0, then the semaphore is not owned by the host (it is owned by the DSP). Semaphore requests are pending, so the host can continue to poll SEM0 until it is released by the DSP. Pending semaphore requests can be cleared at any time by writing a 0 to the SEM0 bit. When the host wants to release the semaphore ownership, a 0 should be written to the SEM0 bit. Table 6–5 summarizes the function of each bit in the SEM0 register. Note that only the LSB is used.

Table 6–5. Host SEM0 Register Bit Definitions

Bit	Name	RW	Description
7–1	RSVD	R	Reserved. Bits 7–1 return 0s when read.
0	SEM0	RW	Semaphore 0 0 = not owned (default) 1 = owned/request

6.5 Host SEM1 Register (Offset 0x90)

The SEM1 register provides a hardware semaphore that is shared with the DSP SEM1. This semaphore can be used by host and DSP applications to control access to resources such as shared memory. To request the semaphore a 1 should be written to the SEM1 bit. The SEM1 bit should then be read to determine if the semaphore is owned (SEM1 = 1). If SEM1 is 0, then the semaphore is not owned by the host (it is owned by the DSP). Semaphore requests are pending, so the host can continue to poll SEM1 until it is released by the DSP. Pending semaphore requests can be cleared at any time by writing a 0 to the SEM1 bit. When the host wants to release the semaphore ownership, a 0 should be written to the SEM1 bit. Table 6–6 summarizes the function of each bit in the SEM1 register. Note that only the LSB is used.

Table 6–6. Host SEM1 Register Bit Definitions

Bit	Name	RW	Description
7–1	RSVD	R	Reserved. Bits 7–1 return 0s when read.
0	SEM1	RW	Semaphore 1 0 = not owned (default) 1 = owned/request

Chapter 7

DSP CPLD Registers

There are 8 DSP CPLD registers mapped into the DSP's lower I/O address space starting at address 0h. Since only the lower 6 I/O address bits are decoded, the registers will alias to other I/O addresses in 64 address blocks within the lower 32K of the I/O address space. The upper 32K I/O addresses are available for daughterboards.

For consistency and ease of programmer usage, register bits are active high, meaning that the intended action is always selected by setting a bit to a 1. If the output signal is active low, then this polarity change is automatically handled in the CPLD.

Topic **Page**

Table 7–1 shows the eight registers bit definitions.

Table 7–1. DSP CPLD Register Definitions

I/O Address	Name	Description	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CNTL	Misc. control	RSVD R 0	DB_RST RW 0 (no reset)	DB_INT RW 0 (no int)	NMIEN RW 0 (no int)	HINT2 RW 0 (no int)	INT3SEL RW 0 (host)	RSVD R 0	USR_LED RW 0 (off)
1	STAT	Misc. status	RSVD R 0	USR_SW1 R –	USR_SW0 R –	NMI R –	DB_INT3 R –	HST_INT3 R –	DB_DET R –	PCI_DET R –
2	DMCNTL	Data memory control	DM_SEL RW 1 (DB)	DB_WIDE RW 0 (16 bits)	DB_32ODD RW 0 (even)	DM_PG4 RW 0 (page 0)	DM_PG3 RW 0 (page 0)	DM_PG2 RW 0 (page 0)	DM_PG1 RW 0 (page 0)	DM_PG0 RW 0 (page 0)
3	DBIO	Daughter-board GP I/O	DB_IODIR3 RW 0 (input)	DB_IODIR2 RW 0 (input)	DB_IODIR1 RW 1 (output)	DB_IODIR0 RW 1 (output)	DB_IO3 RW 0 (inactive)	DB_IO2 RW 0 (inactive)	DB_IO1 RW 0 (inactive)	DB_IC0 RW 0 (inactive)
4	CCNTL	Codec control	C_CLKEN RW 0 (reset)	C_ENABLE RW 0 (reset)	RSVD R 0	MICSEL_L RW 0 (line)	MICSEL_R RW 0 (line)	DEM1 RW 1 (none)	DEM0 RW 1 (none)	SPDMODE RW 0 (<48K)
5	CCLK	Codec clock	CLKSEL RW 0 (24.576M)	RSVD R 0	RSVD R 0	CLKDIV2 RW 0 (16K)	CLKDIV1 RW 1 (16K)	CLKDIV0 RW 0 (16K)	MCLKSEL1 RW 0 (16K)	MCLKSEL0 RW 0 (16K)
6	SEM0	DSP Semaphore 0	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	SEM0 RW 0 (not own)
7	SEM1	DSP Semaphore 1	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	RSVD R 0	SEM1 RW 0 (not own)

7.1 DSP CNTL Register (I/O Address 0)

The CNTL register controls daughterboard signals, interrupts, and the user LED mounted on the EVM's bracket. DB_RST resets the daughterboard under software control. The daughterboard is reset also during power-up and when the manual reset button is depressed. DB_INT provides a way for the DSP to interrupt a daughterboard. The polarity of the signal is inverted from the bit setting. NMIEN selects whether the NMI interrupt from the host is routed to the DSP's NMI pin. HINT2 provides a secondary host interrupt that can be software controlled in addition to the HPIC register's host interrupt (HINT). The polarity of HINT2 is inverted from the bit setting. This interrupt should be useful for background processing, or processing independent from an application such as file I/O coordination. INT3SEL selects whether the DSP's INT3 is generated by the host or the daughterboard. USR_LED controls the user-defined LED on the EVM's bracket. Table 7–2 summarizes the function of each bit in the CNTL register.

Table 7–2. DSP CNTL Register Bit Definitions

Bit	Name	RW	Description
7	RSVD	R	Reserved. Bit 7 returns 0 when read.
6	DB_RST	RW	Daughterboard reset 0 = no reset (default) 1 = reset
5	DB_INT	RW	Daughterboard interrupt 0 = no interrupt (default) 1 = interrupt
4	NMIEN	RW	DSP NMI enable 0 = disable NMI (default) 1 = enable NMI
3	HINT2	RW	DSP-to-host interrupt (GPINT2) 0 = no interrupt (default) 1 = interrupt
2	INT3SEL	RW	DSP INT3 source select 0 = host (default) 1 = daughterboard
1	RSVD	R	Reserved. Bit 1 returns 0 when read.
0	USR LED	RW	User-defined LED control 0 = extinguished (default) 1 = illuminated

7.2 DSP STAT Register (I/O Address 1)

The STAT register provides information about the board's state including the user-defined switches, NMI interrupt status, INT3 interrupt status from DSP and daughterboard, and daughterboard and PCI detection status. USR_SW1 and USR_SW0 provide the user-defined DIP switch settings. NMI provides the host-controlled NMI interrupt state. This can be used as a polled interrupt, if the NMIEN is not set in the CNTL register. DB_INT3 provides the current state of the INT3 interrupt from the daughterboard. HST_INT3 provides the current state of the INT3 interrupt from the host. These two interrupt flags can be used for interrupt polling. DB_DET and PCI_DET are set to indicate whether a daughterboard is installed or the EVM is installed in a PCI slot, respectively. Table 7–3 summarizes the function of each bit in the STAT register.

Table 7–3. DSP STAT Register Bit Definitions

Bit	Name	RW	Description
7	RSVD	R	Reserved. Bit 7 returns 0 when read.
6	USR_SW1	R	User switch 1 state 0 = off 1 = on
5	USR_SW0	R	User switch 0 state 0 = off 1 = on
4	NMI	R	Host-controlled NMI state 0 = not asserted 1 = asserted
3	DB_INT3	R	Daughterboard INT3 state 0 = no interrupt 1 = interrupt
2	HST_INT3	R	Host INT3 state 0 = no interrupt 1 = interrupt
1	DB_DET	R	Daughterboard detect 0 = no daughterboard 1 = daughterboard installed
0	PCI_DET	R	PCI slot detect 0 = standalone 1 = PCI

7.3 DSP DMCTRL Register (I/O Address 2)

The DMCTRL register enables the DSP software to control the data memory space to select between external onboard and daughterboard data memory, the width and access mode of daughterboard data memory and I/O accesses, and multiple pages of data memory and I/O. The 54x DSP has a 16-bit data bus, so a registered transceiver and control logic are included on the EVM to interface to 32-bit daughterboards designed for the C6x EVM. The 54x DSP only provides a 32K window for external data memory. Therefore, to expand on this, external data memory page selection is provided.

DM_SEL selects whether the onboard SRAM or the daughterboard SRAM is active. By default, the daughterboard data memory is selected to be able to support parallel booting from a daughterboard (such as a Flash memory daughterboard).

DB_WIDE determines whether daughterboard data memory and I/O accesses are 16- or 32-bit. When 32-bit (wide) mode is selected, daughterboard data memory and I/O accesses are asserted to the daughterboard based on the DB_32ODD selection. When DB_WIDE is 0, 16-bit accesses are performed, and when DB_WIDE is 1, 32-bit accesses are performed.

DB_32ODD selects whether a daughterboard memory access is to an even (0) or odd (1) address. This selection is required to enable data and I/O selects to the daughterboard at the appropriate time. If a 32-bit daughterboard access is to an odd address, then DB_32ODD should be 1. If a 32-bit daughterboard access is to an even address, then DB_32ODD should be 0. For 32-bit daughterboard writes, the upper 16 bits should be written first to the destination address plus 1, and then lower 16 bits should be written to the destination address. For example, for a 32-bit write to daughterboard memory at 0x8000, DB_32ODD should be 0, the most-significant word (MSW) should be written to 0x8001, and the least-significant word (LSW) should be written to 0x8000. For 32-bit daughterboard reads, the lower 16 bits should be read first from the source address, and then the upper 16 bits should be read from the source address plus 1. For example, for a 32-bit read from daughterboard memory at 0x8001, DB_32ODD should be 1, the LSW should be read from 0x8001 and the MSW should be read from 0x8002.

The DM_PG4–DM_PG0 bits select the current 32K data memory or I/O page. With five bits provided, thirty-two, 32K data memory pages can be accessed onboard, as well as on a daughterboard. This provides a total external data memory and I/O address space of 1M words. All external data memory is accessed in the DSP address region from 0x8000 to 0xFFFF (32K words). Note that for data and I/O accesses that DSP address bit 15 does not get presented to the daughterboard since it is always a 1. Only DSP address bits 0–14 are presented, with address bits 15–19 being provided by the DM_PG4–DM_PG0 bits.

Table 7–4 summarizes the function of each bit in the DMCTRL register.

Table 7–4. DSP DMCTRL Register Bit Definitions

Bit	Name	RW	Description
7	DM_SEL	RW	Data memory selection 0 = onboard SRAM 1 = daughterboard (default)
6	DB_WIDE	RW	Daughterboard data memory width selection 0 = 16 bits (default) 1 = 32 bits
5	DB_32ODD	RW	32-bit daughterboard address access mode 0 = even (default) 1 = odd
4	DM_PG4	RW	Data memory page bit 4 (defaults to 0)
3	DM_PG3	RW	Data memory page bit 3 (defaults to 0)
2	DM_PG2	RW	Data memory page bit 2 (defaults to 0)
1	DM_PG1	RW	Data memory page bit 1 (defaults to 0)
0	DM_PG0	RW	Data memory page bit 0 (defaults to 0)

7.4 DSP DBIO Register (I/O Address 3)

The DBIO register provides four general-purpose bits of I/O to a daughterboard. The four signals can be defined as inputs or outputs under software control. By default, two signals default to inputs and two signals default to outputs to be backwards compatible with existing daughterboards' two STAT input and two CNTL output signals. The DB_IODIR bits select whether the signals are inputs or outputs. The DB_IO bits select the output values or provide the current signal values. Table 7–5 summarizes the function of each bit in the DBIO register.

Table 7–5. DSP DBIO Register Bit Definitions

Bit	Name	RW	Description
7	DB_IODIR3	RW	Daughterboard I/O #3 direction 0 = input (default) 1 = output (XSTAT1)
6	DB_IODIR2	RW	Daughterboard I/O #2 direction 0 = input (default) 1 = output (XSTAT0)
5	DB_IODIR1	RW	Daughterboard I/O #1 direction 0 = input 1 = output (XCNTL1) (default)
4	DB_IODIR0	RW	Daughterboard I/O #0 direction 0 = input 1 = output (XCNTL0) (default)
3	DB_IO3	RW	Daughterboard I/O #3 state 0 = TTL low (default) 1 = TTL high
2	DB_IO2	RW	Daughterboard I/O #2 state 0 = TTL low (default) 1 = TTL high
1	DB_IO1	RW	Daughterboard I/O #1 state 0 = TTL low (default) 1 = TTL high
0	DB_IO0	RW	Daughterboard I/O #0 state 0 = TTL low (default) 1 = TTL high

7.5 DSP CCTRL Register (I/O Address 4)

The CCTRL register enables DSP software to control the AD77 audio codec on the EVM, as well as the input selections for each stereo channel. C_CLKEN controls the codec clocks. For minimum power dissipation, this bit defaults to 0 to disable the codec clocks to the CPLD and the codec. C_ENABLE controls the codec's powerdown input. It is important that when the codec is placed in the power-down state from the active state that the codec clocks be active. The codec clocks can be disabled after the codec is disabled to prevent excess current drain. When the codec is not enabled, the codec is in a low-power state which is the default setting. The codec library must enable the clocks and codec before the codec can be used. The MICSEL bits select the input source for the left and right channels. Each channel can be from the microphone or the line input. The DEM bits select the de-emphasis filtering operation of the codec. The SPDMODE bit selects the sampling mode of the codec which is determined by the sample rate. Table 7–6 summarizes the function of each bit in the CCTRL register.

Table 7–6. DSP CCTRL Register Bit Definitions

Bit	Name	RW	Description
7	C_CLKEN	RW	Codec clocks enable 0 = disable (default) 1 = enable
6	C_ENABLE	RW	Codec enable 0 = power down (default) 1 = enable
5	RSVD	R	Reserved. Bit 5 returns 0 when read.
4	MICSEL_L	RW	Left codec input select 0 = line in (default) 1 = microphone
3	MICSEL_R	RW	Right codec input select 0 = line in (default) 1 = microphone
2	DEM1	RW	Codec de-emphasis select (default to 1) (see Table 7–7 for details)
1	DEM0	RW	Codec de-emphasis select (default to 1) (see Table 7–7 for details)
0	SPDMODE	RW	Codec speed mode 0 =< 48 kHz (default) 1 => 48 kHz

Table 7–7. Codec De-Emphasis Selection

DEM1	DEM0	De-Emphasis Selection
0	0	32 kHz
0	1	44.1 kHz
1	0	48 kHz
1 (default)	1 (default)	Off (default)

7.6 DSP CCLK Control Register (I/O Address 5)

The CCLK register selects the clock source and clock divisors required to generated the 15 supported codec sample rates. CLKSEL selects the source clock frequency of 24.575 MHz (0) or 33.8688 MHz (1). The CLKDIV bits select the clock divisor used inside the CPLD when MCLKSEL = 00. The actual clock division is ((CLKDIV+1)*2) when MCLKSEL = 00. The MCLKSEL bits select which master clock output is provided to the codec and DSP (00 = CLKDIV, 01 = x1, 10 = /2, 11 = /3). Table 7–8 summarizes the function of each bit in the CCLK register. Table 7–9 summarizes the bit selections for the 15 supported sample rates.

Table 7–8. DSP CCLK Register Bit Definitions

Bit	Name	RW	Description
7	CLKSEL	RW	Codec source clock select 0 = 24.576 MHz (default) 1 = 33.8688 MHz
6–5	RSVD	R	Reserved. Bits 6–5 return 0s when read.
4	CLKDIV2	RW	Clock divisor 2 (see Table 15)
3	CLKDIV1	RW	Clock divisor 1 (see Table 15)
2	CLKDIV0	RW	Clock divisor 0 (see Table 15)
1	MCLKSEL1	RW	Clock divisor 1 (see Table 15)
0	MCLKSEL0	RW	Clock divisor 2 (see Table 15)

Table 7–9. Codec Sample Rate Selection

Sample Rate	Value	CLKSEL	CLKDIV2	CLKDIV1	CLKDIV0	MCLKSEL1	MCLKSEL0
8000	0x14	0	1	0	1	0	0
9600	0x10	0	1	0	0	0	0
11025	0x94	1	1	0	1	0	0
12000	0x0C	0	0	1	1	0	0
13230	0x90	1	1	0	0	0	0
16000	0x08	0	0	1	0	0	0
16537.5	0x8C	1	0	1	1	0	0
22050	0x88	1	0	1	0	0	0
24000	0x04	0	0	0	1	0	0
32000	0x03	0	0	0	0	1	1
33075	0x84	1	0	0	1	0	0
44100	0x83	1	0	0	0	1	1
48000	0x02	0	0	0	0	1	0
66150	0x82	1	0	0	0	1	0
96000	0x01	0	0	0	0	0	1

7.7 DSP SEM0 Register (I/O Address 6)

The SEM0 register provides a hardware semaphore that is shared with the host SEM0. This semaphore can be used by host and DSP applications to control access to resources such as shared memory. To request the semaphore a 1 should be written to the SEM0 bit. The SEM0 bit should then be read to determine if the semaphore is owned (SEM0 = 1). If SEM0 is 0, then the semaphore is not owned by the DSP (it is owned by the host). Semaphore requests are pending, so the DSP can continue to poll SEM0 until it is released by the host. Pending semaphore requests can be cleared at any time by writing a 0 to the SEM0 bit. When the DSP wants to release the semaphore ownership, a 0 should be written to the SEM0 bit. Table 7–10 summarizes the function of each bit in the SEM0 register. Note that only the LSB is used.

Table 7–10. DSP SEM0 Register Bit Definitions

Bit	Name	RW	Description
7–1	RSVD	R	Reserved. Bits 7–1 return 0s when read.
0	SEM0	RW	Semaphore 0 0 = not owned (default) 1 = owned/request

7.8 DSP SEM1 Register (I/O Address 7)

The SEM1 register provides a hardware semaphore that is shared with the host SEM1. This semaphore can be used by host and DSP applications to control access to resources such as shared memory. To request the semaphore a 1 should be written to the SEM1 bit. The SEM1 bit should then be read to determine if the semaphore is owned (SEM1 = 1). If SEM1 is 0, then the semaphore is not owned by the DSP (it is owned by the host). Semaphore requests are pending, so the DSP can continue to poll SEM1 until it is released by the host. Pending semaphore requests can be cleared at any time by writing a 0 to the SEM1 bit. When the DSP wants to release the semaphore ownership, a 0 should be written to the SEM1 bit. Table 7–11 summarizes the function of each bit in the SEM1 register. Note that only the LSB is used.

Table 7–11. DSP SEM1 Register Bit Definitions

Bit	Name	RW	Description
7–1	RSVD	R	Reserved. Bits 7–1 return 0s when read.
0	SEM1	RW	Semaphore 1 0 = not owned (default) 1 = owned/request)

Chapter 8

Connector Definitions

Topic Page

8.1 Codec Digital Interface Connector (J4)

The codec's digital interface signals are provided on J4 for external testing of the codec. J4 is not installed on production EVMs.

Table 8–1. Codec Digital Interface Connector J4 Pinout

J4 Pin No.	Signal Name	Description	Type
1	C_MCLK	Codec master clock (x256)	O
2	C_SCLK	Codec serial clock (x64)	O
3	C_LRCLK	Codec left/right (frame sync) clock	O
4	C_SDIN	Codec serial data in	I
5	C_SDOUT	Codec serial data out	O
10	GND	–	–

8.2 CPLD ISR Header Connector (J5)

Connector J5 provides the CPLD's JTAG in-system reprogramming port which allows the EVM's onboard logic to be reprogrammed. This connector is a 10-pin header (two rows of five pins) with connections shown in Table 8–2 to communicate with a Cypress parallel port cable. The 10-pin female connector on the Cypress cable is connected to the male header on the EVM. The pins have 0.025" square posts with 0.100" spacing. Note that this connector is not installed on production EVMs since the CPLD can be programmed over the PCI bus using a console utility (evmcpld.exe).

Table 8–2. CPLD ISR Header Connector J5 Pinout

J5 Pin No.	Signal Name	Description	Type
1	GND	Ground	–
2	TMS	Test mode select	I
3	NC	–	–
4	TCK	Test clock	I
5	NC	–	–
6	TDI	Test data input	I
7	VCC5	+5 V	O
8	NC	–	–
9	TDO	Test data output	O
10	GND	–	–

8.3 Mictor Logic Analyzer Header Connector (J6)

Connector J6 provides the PCI2040s HPI and GP bus signals for observation. J6 is not installed on production EVMs.

Table 8–3. Mictor Logic Analyzer Header Connector J6 Pinout

J6 Pin No.	Signal Name	Description	Type
1–2	NC	–	–
3	GND	Ground	–
4	NC	–	–
5	M_GPIO0	General-purpose I/O #0	I/O
6	$\overline{\text{M\_GPRDY}}$	General-purpose ready	O
7	M_GPIO1	General-purpose I/O #1	I/O
8	H_AD15	Host port address/data bit 15	I/O
9	M_GPIO2	General-purpose I/O #2	I/O
10	H_AD14	Host port address/data bit 14	I/O
11	M_GPIO3	General-purpose I/O #3	I/O
12	H_AD13	Host port address/data bit 13	I/O
13	M_GPIO4	General-purpose I/O #4	I/O
14	H_AD12	Host port address/data bit 12	I/O
15	M_GPIO5	General-purpose I/O #5	I/O
16	H_AD11	Host port address/data bit 11	I/O
17	H_R/$\overline{\text{W}}$	Host port read/write	O
18	H_AD10	Host port address/data bit 10	I/O
19	H_CNTL1	Host port control 1	O
20	H_AD9	Host port address/data bit 9	I/O
21	H_CNTL0	Host port control 0	O
22	H_AD8	Host port address/data bit 8	I/O
23	H_HWIL	Host port word indication	O
24	H_AD7	Host port address/data bit 7	I/O
25	$\overline{\text{H\_BE1}}$	Host port byte enable 1	O
26	H_AD6	Host port address/data bit 6	I/O
27	$\overline{\text{H\_BE0}}$	Host port byte enable 0	O
28	H_AD5	Host port address/data bit 5	I/O

Table 8–3. Mictor Logic Analyzer Header Connector J6 Pinout (Continued)

J6 Pin No.	Signal Name	Description	Type
29	$\overline{\text{H\_DS}}$	Host port data strobe	O
30	H_AD4	Host port address/data bit 4	I/O
31	$\overline{\text{H\_CS0}}$	Host port chip select #0	O
32	H_AD3	Host port address/data bit 3	I/O
33	H_INT0	Host port interrupt (HINT) #0	O
34	H_AD2	Host port address/data bit 2	I/O
35	$\overline{\text{H\_RST0}}$	Host reset #0	O
36	H_AD1	Host port address/data bit 1	I/O
37	H_RDY0	Host ready #0	O
38	H_AD0	Host port address/data bit 0	I/O

8.4 Expansion Memory Interface Connector (J7)

Connector J7 provides the DSP expansion memory interface signals to a daughterboard which can provide additional memory and memory-mapped devices.

Table 8–4. Expansion Memory Interface Connector J7 Pinout

J7 Pin No.	Signal Name	Type	J7 Pin No.	Signal Name	Type
1	5 V	O	2	5 V	O
3	X_A19	O	4	X_A18	O
5	X_A17	O	6	X_A16	O
7	X_A15	O	8	X_A14	O
9	X_A13	O	10	X_A12	O
11	GND	–	12	GND	–
13	X_A11	O	14	X_A10	O
15	X_A9	O	16	X_A8	O
17	X_A7	O	18	X_A6	O
19	X_A5	O	20	X_A4	O
21	5 V	O	22	5 V	O
23	X_A3	O	24	X_A2	O
25	X_A1	O	26	X_A0	O
27	X_A21	O	28	X_A20	O
29	GND	O	30	GND	O
31	GND	–	32	GND	–
33	X_D31	I/O/Z	34	X_D30	I/O/Z
35	X_D29	I/O/Z	36	X_D28	I/O/Z
37	X_D27	I/O/Z	38	X_D26	I/O/Z
39	X_D25	I/O/Z	40	X_D24	I/O/Z
41	3.3 V	O	42	3.3 V	O
43	X_D23	I/O/Z	44	X_D22	I/O/Z
45	X_D21	I/O/Z	46	X_D20	I/O/Z
47	X_D19	I/O/Z	48	X_D18	I/O/Z
49	X_D17	I/O/Z	50	X_D16	I/O/Z
51	GND	–	52	GND	–
53	X_D15	I/O/Z	54	X_D14	I/O/Z
55	X_D13	I/O/Z	56	X_D12	I/O/Z

Table 8–4. Expansion Memory Interface Connector J7 Pinout (Continued)

J7 Pin No.	Signal Name	Type	J7 Pin No.	Signal Name	Type
57	X_D11	I/O/Z	58	X_D10	I/O/Z
59	X_D9	I/O/Z	60	X_D8	I/O/Z
61	GND	–	62	GND	–
63	X_D7	I/O/Z	64	X_D6	I/O/Z
65	X_D5	I/O/Z	66	X_D4	I/O/Z
67	X_D3	I/O/Z	68	X_D2	I/O/Z
69	X_D1	I/O/Z	70	X_D0	I/O/Z
71	GND	–	72	GND	–
73	$\overline{\text{X\_RE}}$	O	74	$\overline{\text{X\_WE}}$	O
75	$\overline{\text{X\_OE}}$	O	76	X_RDY	I
77	$\overline{\text{X\_MSTRB}}$	O	78	$\overline{\text{X\_DS}}$	O
79	GND	–	80	GND	–

8.5 Expansion Peripheral Interface Connector (J8)

Connector J8 provides DSP expansion peripheral interface signals to a daughterboard.

Table 8–5. Expansion Peripheral Interface Connector J8 Pinout

J8 Pin No.	Signal Name	Type	J8 Pin No.	Signal Name	Type
1	12 V	O	2	–12 V	O
3	GND	–	4	GND	–
5	5 V	O	6	5 V	O
7	GND	–	8	GND	–
9	5 V	O	10	5 V	O
11	SPARE (N/C)	–	12	SPARE (N/C)	–
13	SPARE (N/C)	–	14	SPARE (N/C)	–
15	SPARE (N/C)	–	16	SPARE (N/C)	–
17	SPARE (N/C)	–	18	SPARE (N/C)	–
19	3.3 V	O	20	3.3 V	O
21	X_CLKX0	I/O/Z	22	SPARE (N/C)	–
23	X_FSX0	I/O/Z	24	X_DX0	O
25	GND	–	26	GND	–
27	X_CLKR0	I/O/Z	28	SPARE (N/C)	–
29	X_FSR0	I/O/Z	30	X_DR0	I
31	GND	–	32	GND	–
33	X_CLKX1	I/O/Z	34	SPARE (N/C)	–
35	X_FSX1	I/O/Z	36	X_DX1	O
37	GND	–	38	GND	–
39	X_CLKR1	I/O/Z	40	SPARE (N/C)	–
41	X_FSR1	I/O/Z	42	X_DR1	I
43	GND	–	44	GND	–
45	X_TOUT	O	46	SPARE (N/C)	–
47	SPARE (N/C)	–	48	$\overline{\text{X\_INT1}}$	I
49	X_XF	O	50	$\overline{\text{X\_BIO}}$	I
51	GND	–	52	GND	–
53	$\overline{\text{X\_INT0}}$	I	54	$\overline{\text{X\_IACK}}$	O
55	SPARE (N/C)	O	56	$\overline{\text{X\_IOSTRB}}$	O

Table 8–5. Expansion Peripheral Interface Connector J8 Pinout (Continued)

J8 Pin No.	Signal Name	Type	J8 Pin No.	Signal Name	Type
57	$\overline{\text{X\_MSC}}$	O	58	$\overline{\text{X\_IAQ}}$	O
59	$\overline{\text{X\_RESET}}$	O	60	$\overline{\text{X\_DBINT}}$	O
61	GND	–	62	GND	–
63	X_CNTL1	O	64	X_CNTL0	O
65	X_STAT1	I	66	X_STAT0	I
67	$\overline{\text{X\_INT2}}$	I	68	$\overline{\text{X\_INT3}}$	I
69	$\overline{\text{X\_PS}}$	O	70	$\overline{\text{X\_IS}}$	O
71	SPARE (N/C)	O	72	SPARE (N/C)	–
73	SPARE (N/C)	O	74	SPARE (N/C)	–
75	$\overline{\text{DB\_DET}}$	I	76	GND	–
77	GND	–	78	X_CLKOUT	O
79	GND	–	80	GND	–

8.6 DSP JTAG Emulation Header Connector (J9)

Connector J9 provides the DSPs emulation port based on the IEEE 1149.1 standard. This connector is a 14-pin header (two rows of seven pins) with connections shown in Table 8–6 to communicate with an XDS510 emulator. Pin 6 is used for keying to ensure a proper connection.

Table 8–6. DSP JTAG Emulation Header Connector J9 Pinout

J9 Pin No.	Signal Name	Description	Type
1	TMS	Test mode select	I
2	TRST–	Test reset	I
3	TDI	Test data input	I
4	GND	Ground	–
5	PD (VCC)	Presence detect. Indicates that the emulation cable is connected and the target is powered up. PD is tied to 3.3 V on the EVM.	O
6	KEY	Not used. This pin is cut off on the J9 header. This pin is filled in on the XDS510 connector.	–
7	TDO	Test data out	O
8	GND	Ground	–
9	TCK_RET	Test clock return. Test clock input to the emulator.	O
10	GND	Ground	–
11	TCK	Test clock. TCK is a 10.368-MHz clock source from the emulation cable pod.	I
12	GND	Ground	–
13	EMU0	Emulation pin 0	I/O
14	EMU1	Emulation pin 1	I/O

8.7 External Power Connector (J10)

Table 8–7. External Power Connector J10 Pinout

J10 Pin No.	Signal Name	Description	Type
1	12	12 Vdc @ 500 mA	I
2	–12	–12 Vdc @ 100 mA	I
3	GND	Ground	–
4	5	5 Vdc @ 4 A	I