Hammerhead-6U-cPCI

Eight ADSP-21160 64-bit, 66 MHz 6U cPCI Board

User's Guide

BittWare

BittWare, Inc.
33 N Main St.
Concord, NH 03301 USA
603.226.0404

If you have comments or suggestions about this manual or find any errors in it, please contact us via e-mail at techpubs@bittware.com.

For technical support, contact us using any of the following methods:
Phone: 603.226.0404
FAX: 603.226.6667
E-mail: support@bittware.com

BittWare also maintains the following Internet sites:

http://www.bittware.com	Contains product information, technical notes, support files available for download, and answers to frequently asked questions (FAQ).
ftp://ftp.bittware.com	Contains technical notes and support files. Login as "anonymous" and use your e-mail address for the password.

Hammerhead-6U-cPCI User's Guide

Hardware Revision 1

The information in this manual has been carefully checked and is believed to be accurate and reliable. However, BittWare assumes no responsibility for any inaccuracies, errors, or omissions that may be contained in this manual. In no event will BittWare be liable for direct, indirect, special, incidental, or consequential damages resulting from any defect or omission in this manual. BittWare reserves the right to revise this document and to make changes from time to time in the content hereof without obligation of BittWare to notify any person or persons of such revision or changes.

Printed in the USA
October 15, 2001 Edition UG-HH6U01-00

Contents

Chapter 1 Introduction

Chapter 2 Preparing the Board for Operation

Chapter 3 Overview of the Hardware Components

Chapter 4 Hammerhead-6U-cPCI Board Architecture

Chapter 5 ADSP-21160 Programming Details

Chapter 6 SharcFIN Programming Details

Appendix A Debugging Your DSP Programs

Appendix B Setting Up for Standalone Operation

Appendix C Troubleshooting Tips

Tables

Figures

Chapter 1
Introduction

BittWare’s Hammerhead-6U-cPCI board packs the processing power of eight ADSP-21160 SHARC® DSPs and the speed of a 64-bit, 66 MHz PCI interface onto a 6U CompactPCI board. The board supports up to three banks of 64–512 MB of SDRAM, two 2 MB banks of Flash memory, and two PMC+ mezzanine sites. The board also features two BittWare Sharc®FIN™ ASICs, which flexibly interface the ADSP-21160s to the 64-bit, 66 MHz cPCI interface, the SDRAM, the Flash memory, and a general-purpose expansion bus.

This chapter covers the following topics:

- The basic architecture of the Hammerhead-6U-cPCI system
- An overview of each chapter in this user’s guide
- Additional documents that provide more information about the Hammerhead-6U-cPCI’s components and software

1.1 Overview of the Hammerhead-6U-cPCI System

This section gives a brief overview of the architecture of the board and describes its software.

1.1.1 Hammerhead-6U-cPCI Features

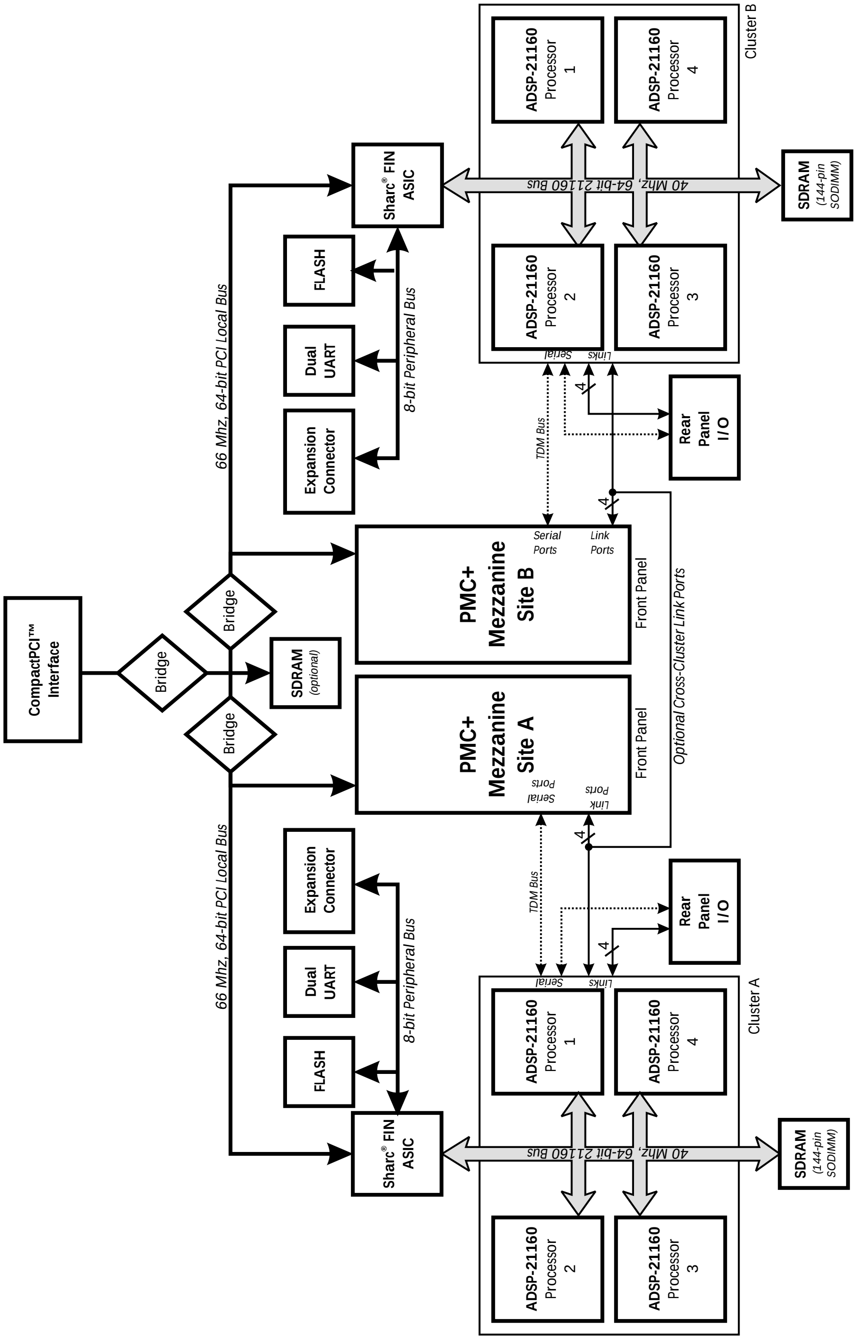
The following is a list of the Hammerhead-6U-cPCI's features:

- Eight 80 MHz ADSP-21160 SHARC processors
- 64-bit, 66 MHz PCI interface
- Up to three 64–512 MB banks of SDRAM (standard 144-pin SODIMM)
- Two PMC sites with PMC+ extensions for BittWare's PMC+ I/O modules
- Eight link ports and two serial TDM buses for integrating PMC+ I/O with on-board SHARCs
- Eight 80 MB/s external link ports
- Two 40 Mb/s external serial TDM buses
- Two BittWare SharcFIN ASICs
- Two RS-232 UARTs
- Two 2 MB banks of Flash memory
- 6U CompactPCI form factor
- Standalone operation

1.1.2 Hammerhead-6U-cPCI System Architecture

This section gives a basic overview of the Hammerhead-6U-cPCI system. Figure 1–1 is a detailed block diagram of the Hammerhead-6U-cPCI board and its features.

Figure 1–1 *Block Diagram of the Hammerhead-6U-cPCI System*



BittWare's Hammerhead-6U-cPCI board features eight ADSP-21160 SHARC DSPs and a 64-bit, 66 MHz PCI interface. It also features two BittWare SharcFIN ASICs, up to three banks of SDRAM, two 2 MB banks of Flash memory, and two PMC+ sites.

SharcFIN ASIC

The Hammerhead-6U-cPCI incorporates a BittWare SharcFIN ASIC for each cluster. The SharcFIN flexibly interfaces the ADSP-21160 DSPs to the 64-bit, 66 MHz PCI bus, the SDRAM, and a peripheral bus, which interfaces to the board's UARTs and Flash memory. The SharcFIN also provides a feature-rich set of DMA functions and interrupt options to support very high-speed, real-time data flow with minimum processor overhead.

ADSP-21160 DSPs

The Hammerhead-6U-cPCI is configured with eight 80 MHz ADSP-21160 DSPs, arranged in two clusters of four processors. Each cluster shares a common 40 MHz, 64-bit cluster bus, which gives it access to a bank of up to 512 MB SDRAM, the PCI bus interface, and the other three ADSP-21160 processors in the cluster. For additional I/O, each processor also has four flags, three interrupts, six link ports, and two serial ports.

PMC+ Mezzanine Sites

The Hammerhead-6U-cPCI features two PMC+ (PCI Mezzanine Card) sites. The PMC+ sites have front-panel access and allow you to attach standard PMC modules to the board, adding I/O or additional processors and memory. The PMC+ sites also function as proprietary interfaces that allow you to attach BittWare's PMC+ I/O modules for low-latency, high-performance I/O via four 100 MB/s link ports and a serial TDM bus from each mezzanine site.

I/O Options

In addition to the PMC+ interfaces, the Hammerhead-6U-cPCI has several other options for I/O: external link ports, external serial ports, and RS-232 ports. Eight 80 MB/s link ports, one from each DSP, extend from the ADSP-21160s to rear panel I/O; and a 40 Mb/s TDM serial bus extends from each cluster to the ADSP-21160s and to rear panel I/O. Two RS-232 ports allow the DSPs to communicate with external serial devices, facilitating remote debugging, command, and control.

1.1.3 Hammerhead-6U-cPCI Software Architecture

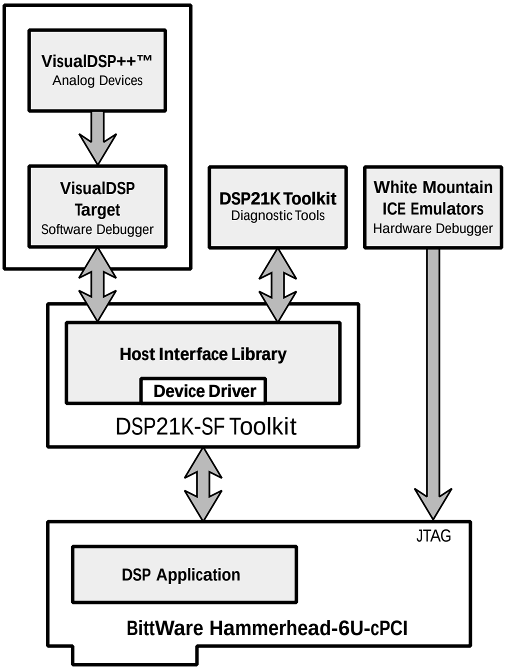
You will need three types of software development tools for the Hammerhead-6U-cPCI: code development tools, debugging tools, and host interface tools. Figure 1–2 is a general block diagram of the software development tools.

To begin developing code for the Hammerhead-6U-cPCI, use Analog Devices' VisualDSP++® Integrated Development Environment (IDE). VisualDSP++ is an easy-to-use project management environment comprised of an integrated development environment (IDE) and software debugger.

Once you have developed your code, you can debug it using BittWare's VisualDSP Target™, which is a software plug-in for VisualDSP++ that allows the VisualDSP++ debugger to communicate directly with your BittWare board. You can also use a hardware in-circuit emulator, such as the ICE emulators from Analog Devices to debug your code.

BittWare's DSP21k-SF Toolkit provides your host interface tools. The DSP21k-SF Toolkit is a complete software development kit that allows you to easily develop application code and integrate the Hammerhead-6U-cPCI into your system. The software tools include a comprehensive host interface library (HIL), a standard I/O library, and diagnostic utilities.

Figure 1–2 *Block Diagram of the Hammerhead-6U-cPCI and its Software*



1.2 About this User's Guide

This section provides an overview of each chapter's content and describes certain variations in text and naming conventions we have used throughout the manual.

1.2.1 Purpose of this Document

This user's guide covers hardware revision 1 of the Hammerhead-6U-cPCI board, which supports eight ADSP-21160 SHARC processors operating at 80 MHz. The purpose of this document is to provide details about the Hammerhead-6U-cPCI's major hardware components, to describe how to install and properly operate the Hammerhead-6U-cPCI, and to discuss important issues that relate to programming the board.

We assume you are already familiar with the ADSP-21160 architecture, operation, and programming as described in the *ADSP-21160 User's Manual* from Analog Devices, Inc.

1.2.2 Conventions Used in This Document

We have used the following conventions throughout this user's guide.

- Since the Hammerhead-6U-cPCI has eight DSPs arranged in two clusters of four DSPs, we refer to them as 21160-A1, 21160-B1, etc., where the letter "A" or "B" designates the cluster and the numeral (1–4) designates the DSP number within the cluster.
- All signal names appear in small capitals (RESET).
- Active low signals appear in small capitals with an overline ($\overline{\text{RESET}}$).
- A "0x" prefix designates a number as a hexadecimal number (0x01).
- Commands that the user enters (for programs such as Diag21k or DspBad in the DSP21k-SF Toolkit) appear in the **`Courier bold`** font.
- Filenames and directories appear in the `Courier` font.

1.2.3 Chapter Overviews

Chapter 2: Preparing the Hammerhead-6U-cPCI for Operation
This chapter describes the tasks that you must perform to prepare your board for installation, install the software for the board, install the board, and test the installation.

Chapter 3: Overview of the Hardware Components
This chapter shows the location of the Hammerhead-6U-cPCI's major components and connectors and briefly discusses their function.

Chapter 4: Hammerhead-6U-cPCI Board Architecture
This chapter discusses the board's architecture, including the serial ports, link ports, flags and interrupts, and bus interfaces. It also discusses the architecture of the SharcFIN ASIC.

Chapter 5: Programming Details for the ADSP-21160 DSPs
This chapter provides programming details for the ADSP-21160 DSPs, describing how to access memory and boot the DSPs.

Chapter 6: Programming Details for the SharcFIN ASIC
This chapter provides a brief functional overview of the SharcFIN ASIC and describes how to configure its SHARC interface control registers.

Appendix A: Debugging Your DSP Programs
This appendix gives information on debugging DSP programs with a hardware or software emulator.

Appendix B: Setting up for Standalone Operation
This appendix describes how to set the board up to operate in standalone mode.

Appendix C: Troubleshooting Tips
This appendix gives tips for solving common operating problems and discusses how to contact technical support at BittWare.

Appendix D: Glossary of Terms
This appendix defines terms used throughout this manual.

1.3 Other Helpful Documents and Tools

This section gives sources for additional information that applies to the Hammerhead-6U-cPCI. It also lists several third party software development tools you may find useful.

1.3.1 Documents for Further Reference

The documents in the list below provide additional information about the Hammerhead-6U-cPCI components and software.

- *ADSP-21160 SHARC User's Guide* – Analog Devices, Inc.
- *Intel 21154 Chip Data Sheet* – Intel Corporation
- *SharcFIN ASIC User's Guide* – BittWare, Inc.
- *QL5064 User's Guide* – Quick Logic Corporation
- *DSP21k-SF Toolkit User's Guide* (Version 6.0 and up) – BittWare, Inc.

1.3.2 Software DevelopmentTools

VisualDSP++® and BittWare VisualDSP Target

The Hammerhead-6U-cPCI is compatible with the VisualDSP++ development tools from Analog Devices. VisualDSP++ is an easy-to-use project management environment comprised of an integrated development environment (IDE) and software debugger. The IDE provides access to Analog Devices' 4.0 SHARC C compiler, C runtime library, assembler, linker, loader, simulator, and splitter. The debugger has an easy-to-use interface and many features that reduce debugging time by enabling you to set breakpoints, single step through code, and perform many other debugging operations.

BittWare offers the VisualDSP Target, a plug-in to the VisualDSP++ IDE that allows the VisualDSP++ debugger to communicate directly with BittWare's DSP boards. The VisualDSP Target lets you debug your DSP application without a hardware emulator, allowing you to set breakpoints, single-step through your code, view memory, and run code on multiple processors.

White Mountain In-Circuit Emulators

The ICE in-circuit emulators from Analog Devices provide real-time hardware emulation and debugging. Analog Devices offers emulators that are compatible with VisualDSP++ in ISA bus, PCI bus, USB, and Ethernet formats. With ICE emulators, you can load programs, start and stop program execution, observe and alter registers and memory, and perform other debugging operations. If you plan to use an in-circuit emulator with the Hammerhead-6U-cPCI, refer to the documentation that comes with the emulator and to the information in Appendix A of this manual.

BittWare Host Interface Support

BittWare supplies host interface support for the Hammerhead-6U-cPCI with the DSP21k-SF Toolkit. Using the Toolkit's C-callable library of routines for DOS and Windows programs, you can download and start programs, read from and write to the Hammerhead-6U-cPCI memory, and control other board functions. Another library gives your DSP programs standard I/O routines such as screen display, keyboard input, and disk file access. The *DSP21k-SF Toolkit User's Guide* (Version 6.0 and up) from BittWare, Inc. contains complete information about the DSP21k-SF Toolkit.

SpeedDSP Optimized Libraries for SHARC DSPs

SpeedDSP is a collection of highly optimized routines for the ADSP-21xxx family of SHARC DSP chips that includes SIMD optimizations for the ADSP-2116x family of DSPs. The functions are written in ADSP-21xxx assembly language and are callable from high-level languages such as C. SpeedDSP includes functions for manipulating large arrays of floating-point numbers and for performing Fast Fourier Transforms (FFTs), windowing, statistics, sorting, histogramming, trigonometry, and timing. Since the functions in the library are coded in ADSP-21xxx assembly language and take full advantage of the ADSP-21xxx architecture, they are much faster than high-level language implementations, delivering optimum speed and performance. SpeedDSP integrates easily with the Analog Devices C compiler and is completely compatible with the program/data memory specifiers and the complex data type.

BittWare's SharcLAB MATLAB Interface

SharcLAB, developed exclusively for BittWare by SDL, works with The MathWorks MATLAB, Simulink, Stateflow, and Real-Time Workshop to allow you to prototype and test DSP applications on your BittWare SHARC DSP boards. SharcLAB integrates seamlessly with the standard MATLAB environment, allowing a nearly automatic transition from MATLAB-based algorithm development to executable DSP code.

You can develop your applications in the Simulink graphical flow-chart-based simulation environment and use SharcLAB to automatically compile, download, and run the algorithms on your BittWare SHARC DSP hardware in real-time. SharcLAB allows you to change application parameters interactively and view data streams in real time in the native Simulink environment for debugging and verification without interrupting the DSP application.

Chapter 2
Preparing the Board for Operation

This chapter describes the tasks necessary to prepare your Hammerhead-6U-cPCI board for installation, install the software for the board, install the board, and test the installation. This chapter does not provide comprehensive instructions for each task; instead, it provides a sequence of steps for you to follow. In addition to the information in this chapter, you will also need to refer to the documentation for the Analog Devices software, the BittWare DSP21k-SF Toolkit, and your CompactPCI chassis.

To prepare the Hammerhead-6U-cPCI for operation, complete the following steps:

1. Unpack the Hammerhead-6U-cPCI (section 2.1).
2. Set the board's configuration jumpers (section 2.2.1).
3. Configure the board's serial ports (section 2.2.2).
4. Connect any desired external devices to the board (section 2.2.3).
5. Install the VisualDSP++ software tools (section 2.3.1).
6. Install BittWare's DSP21k-SF Toolkit (section 2.3.2).
7. Insert the board in a 6U slot in a CompactPCI chassis (section 2.3.3).
8. Run the example software included with the board (section 2.4.1).
9. Run diagnostic tests on the board to ensure that it is operating properly (section 2.4.2).

2.1 Unpacking the Hammerhead-6U-cPCI

Warning!

The Hammerhead-6U-cPCI contains electro-static discharge (ESD) sensitive devices. Be sure to follow the standard handling procedures for ESD sensitive devices, taking proper precautions to ground yourself and the work area before removing the board from its anti-static bag. If you fail to follow proper handling procedures, you could damage the board.

To unpack the Hammerhead-6U-cPCI board,

1. Carefully remove the board from the shipping box. Save the box and packing materials in case you need to reship the board.
2. Remove the board from the plastic bag. Observe all precautions described in the warning above to prevent damage from electro-static discharge (ESD).
3. Carefully examine the board, checking for damage. If the board is damaged, *do not* install it. Call BittWare technical support.

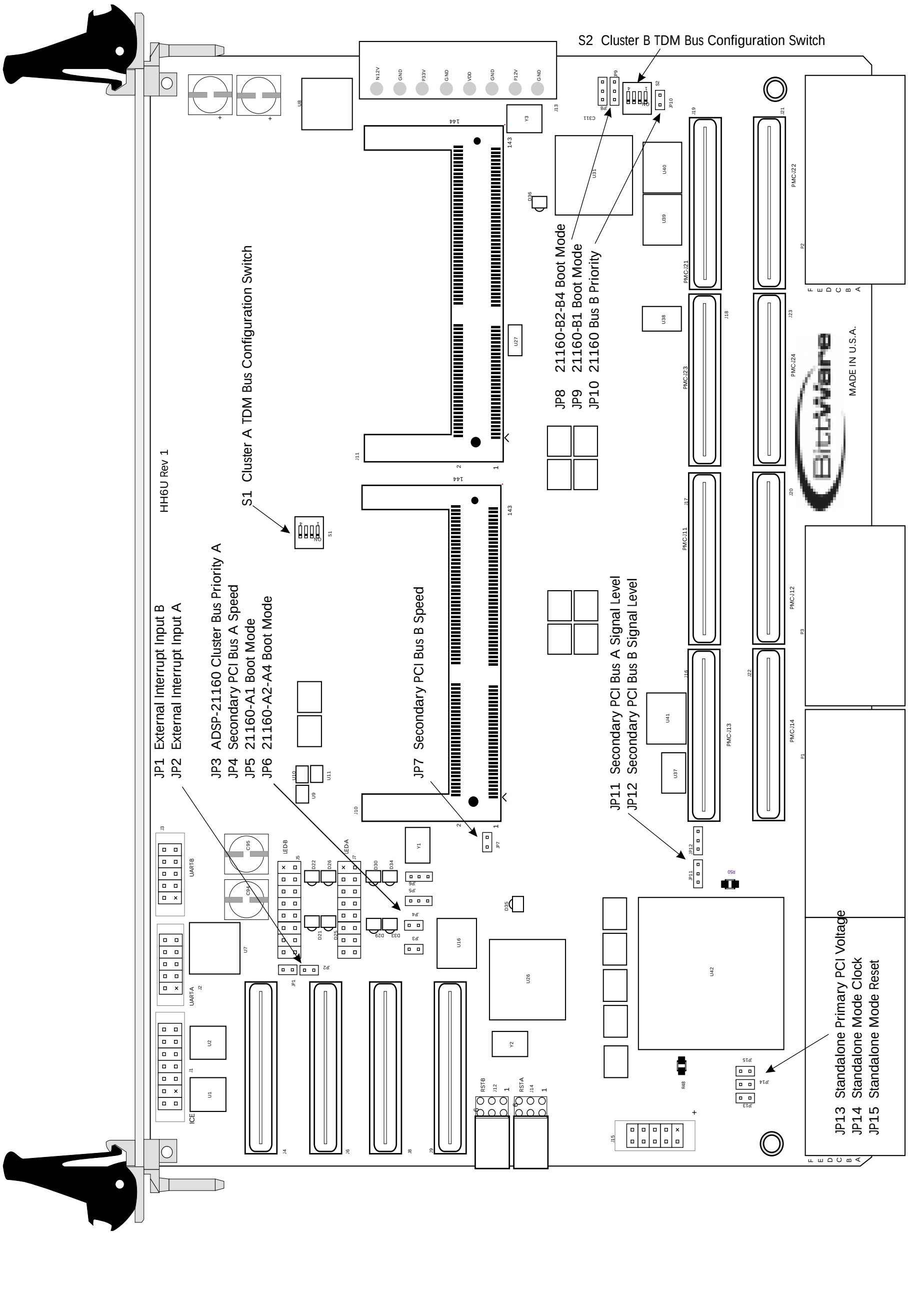
2.2 Configuring the Hammerhead-6U-cPCI

This section explains how to set up the physical features of the board to get it ready for installation. It discusses setting configuration jumpers, configuring the external serial ports, and connecting external devices to the board.

2.2.1 Setting the Configuration Jumpers

The Hammerhead-6U-cPCI has fifteen configuration jumpers that allow you to control and enable certain features on the board. Before installing the Hammerhead-6U-cPCI in the CompactPCI chassis, make sure you have properly set all of the configuration jumpers. Figure 2–1 shows where each of the jumpers is located.

Figure 2–1 *Location of the Hammerhead-6U-cPCI Configuration Jumpers and Switches*


S2 Cluster B TDM Bus Configuration Switch
JP8 21160-B2-B4 Boot Mode
JP9 21160-B1 Boot Mode
JP10 21160 Bus B Priority
S1 Cluster A TDM Bus Configuration Switch
HH6U Rev 1
JP1 External Interrupt Input B
JP2 External Interrupt Input A
JP3 ADSP-21160 Cluster Bus Priority A
JP4 Secondary PCI Bus A Speed
JP5 21160-A1 Boot Mode
JP6 21160-A2-A4 Boot Mode
JP7 Secondary PCI Bus B Speed
JP11 Secondary PCI Bus A Signal Level
JP12 Secondary PCI Bus B Signal Level
JP13 Standalone Primary PCI Voltage
JP14 Standalone Mode Clock
JP15 Standalone Mode Reset
N12V
GND
P33V
GND
VDD
GND
P12V
GND
PMC-J21
PMC-J22
PMC-J23
PMC-J24
PMC-J11
PMC-J12
PMC-J13
PMC-J14
UARTB
UARTA
ICE
LEDB
LEDA
RSTB
RSTA
U42
U26
U16
MADE IN U.S.A.
BittWare

Using the External Interrupt Input

The external interrupt input connectors, JP1 and JP2, are used to send external interrupts directly to the SharcFIN. Do not use a jumper on JP1 and JP2; they are two-pin connectors. See section 3.2.11 for details about using JP1 and JP2.

Selecting the ADSP-21160 Cluster Bus Priority

The Hammerhead-6U-cPCI has two jumpers for configuring the rotating priority bus arbitration (RPBA) mode for the 64-bit ADSP-21160 cluster buses. JP3 configures the RPBA mode for ADSP-21160 cluster bus A, and JP10 configures the RPBA mode for ADSP-21160 cluster bus B.

The ADSP-21160 cluster buses have two RPBA modes: fixed priority scheme and rotating priority scheme. The *ADSP-21160 User's Guide* (Analog Devices) explains the RPBA modes in more detail. The fixed priority scheme for bus arbitration gives priority to the ADSP-21160 processor with the lowest multiprocessor ID. With the fixed priority scheme, 21160-1 (A1 or B1, depending on the cluster) would always have priority. The rotating priority scheme for bus arbitration gives priority to the ADSP-21160 processors on a rotating schedule. For example, 21160-1 (A1 or B1) would have priority, then 21160-2 (A2 or B2) would have priority, and so on.

Table 2–1 *Selecting the Rotating Priority Bus Mode for the ADSP-21160s*

Jumper	Jumper Position	Setting	Default
JP3 (Cluster A)	IN	Fixed priority scheme	
	OUT	Rotating priority scheme	✔
JP10 (Cluster B)	IN	Fixed priority scheme	
	OUT	Rotating priority scheme	✔

Selecting the Boot Mode

Four jumpers configure the boot mode for the ADSP-21160 DSPs. The processors can boot in three boot modes:

- link booting
- host booting
- Flash booting

JP5 and JP6 configure the boot mode for the ADSP-21160 cluster A processors, and JP8 and JP9 configure the boot mode for the ADSP-21160 cluster B processors. Table 2–2 below shows their settings.

Table 2–2 *Settings for the Boot Mode Selection Jumpers*

Jumper	Jumper Position	Setting	Default
JP5	No Jumper	21160-A1 will boot from host computer	✔
	Pins 1–2	21160-A1 will boot from on-board Flash	
	Pins 2–3	21160-A1 will boot via link port	
JP6	No Jumper	21160-A2–A4 will boot from host computer	✔
	Pins 1–2	21160-A2–A4 will boot from on-board Flash	
	Pins 2–3	21160-A2–A4 will boot via link port	
JP9	No Jumper	21160-B2–B4 will boot from host computer	✔
	Pins 1–2	21160-B2–B4 will boot from on-board Flash	
	Pins 2–3	21160-B2–B4 will boot via link port	
JP8	No Jumper	21160-B1 will boot from host computer	✔
	Pins 1–2	21160-B1 will boot from on-board Flash	
	Pins 2–3	21160-B1 will boot via link port	

Configuring the Speed of the Secondary PCI Buses

The Hammerhead-6U-cPCI has two jumpers that allow you to configure the secondary PCI buses for each cluster to run at either 33 MHz or 66 MHz. JP4 configures the speed of the secondary PCI bus for cluster A, and JP7 configures the speed of the secondary PCI bus for cluster B. Table 2–3 shows the settings for the jumpers.

Table 2–3 *Settings for the Secondary PCI Bus Jumpers*

Jumper	Jumper Position	Setting	Default
JP4 (Secondary PCI Bus A)	IN	33 MHz	
	OUT	66 MHz	✔
JP7 (Secondary PCI Bus B)	IN	33 MHz	
	OUT	66 MHz	✔

Configuring the Signal Level of the Secondary PCI Buses

The Hammerhead-6U-cPCI has two 3-pin jumpers that configure the signal level of the secondary PCI bus for each cluster to either 3.3 volts or 5 volts. JP11 configures the signal level of the secondary PCI bus for cluster A, and JP12 configures the signal level of the secondary PCI bus for cluster B. Table 2–4 below gives the jumpers' settings.

Warning!

Be sure to set these jumpers correctly. If you set these jumpers incorrectly, some power supplies could short together and damage the board.

Table 2–4 *Settings for the Secondary PCI Signal Level Jumpers (JP11, JP12)*

Jumper Position	Setting	Default
No Jumper	Board non-functional*	
Pins 1–2	3.3 Volts	
Pins 2–3	5 Volts	✔

* A jumper must be installed on JP11 and JP12 or the board will not function properly.

Setting the Standalone Mode Jumpers

The Hammerhead-6U-cPCI has three jumpers for configuring the board to operate in standalone mode:

JP13	Standalone primary PCI voltage
JP14	Standalone mode clock
JP15	Standalone mode reset

All three jumpers must be IN for the board to operate properly in standalone mode.

Table 2–5 *Settings for the Standalone Mode Jumpers*

Jumper	Jumper Position	Setting	Default
JP13 (PCI voltage)	IN	Standalone	
	OUT	Normal operation	✔
JP14 (Clock)	IN	Standalone	
	OUT	Normal operation	✔
JP15 (Reset)	IN	Standalone	
	OUT	Normal operation	✔

2.2.2 Configuring the TDM Serial Ports

Two TDM serial buses from each cluster provide a communication route between the ADSP-21160s and synchronous serial devices. One serial bus per cluster connects to rear panel I/O (P3 and P5), and one connects to the PMC+ interface. The Hammerhead-6U-cPCI has two sets of switches, one for each cluster, that allow you to configure the signals of the external TDM serial ports. The switch labeled S1 configures the serial ports for cluster A, and S2 configures the serial ports for cluster B.

Table 2–6 *TDM Serial Port Switch Connections and Usage*

Serial Port	Switch	Connections
TDM SPORT A0	S1	SPORT0 from 21160-A1–A4 have TDM serial connection to PMC+ interface A
TDM SPORT A1	S1	SPORT1 from 21160-A1–A4 have TDM serial connection to rear panel I/O (P3)
TDM SPORT B0	S2	SPORT0 from 21160-B1–B4 have TDM serial connection to PMC+ interface B
TDM SPORT B1	S2	SPORT1 from 21160-B1–B4 have TDM serial connection to rear panel I/O (P5)

Setting the TDM Serial Port Configuration Switches

Table 2–7 gives the pinout of the switches, and Table 2–8 shows their settings.

Table 2–7 *TDM Serial Port Switch Pinout (S1, S2)*

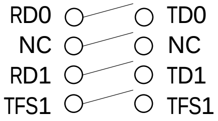
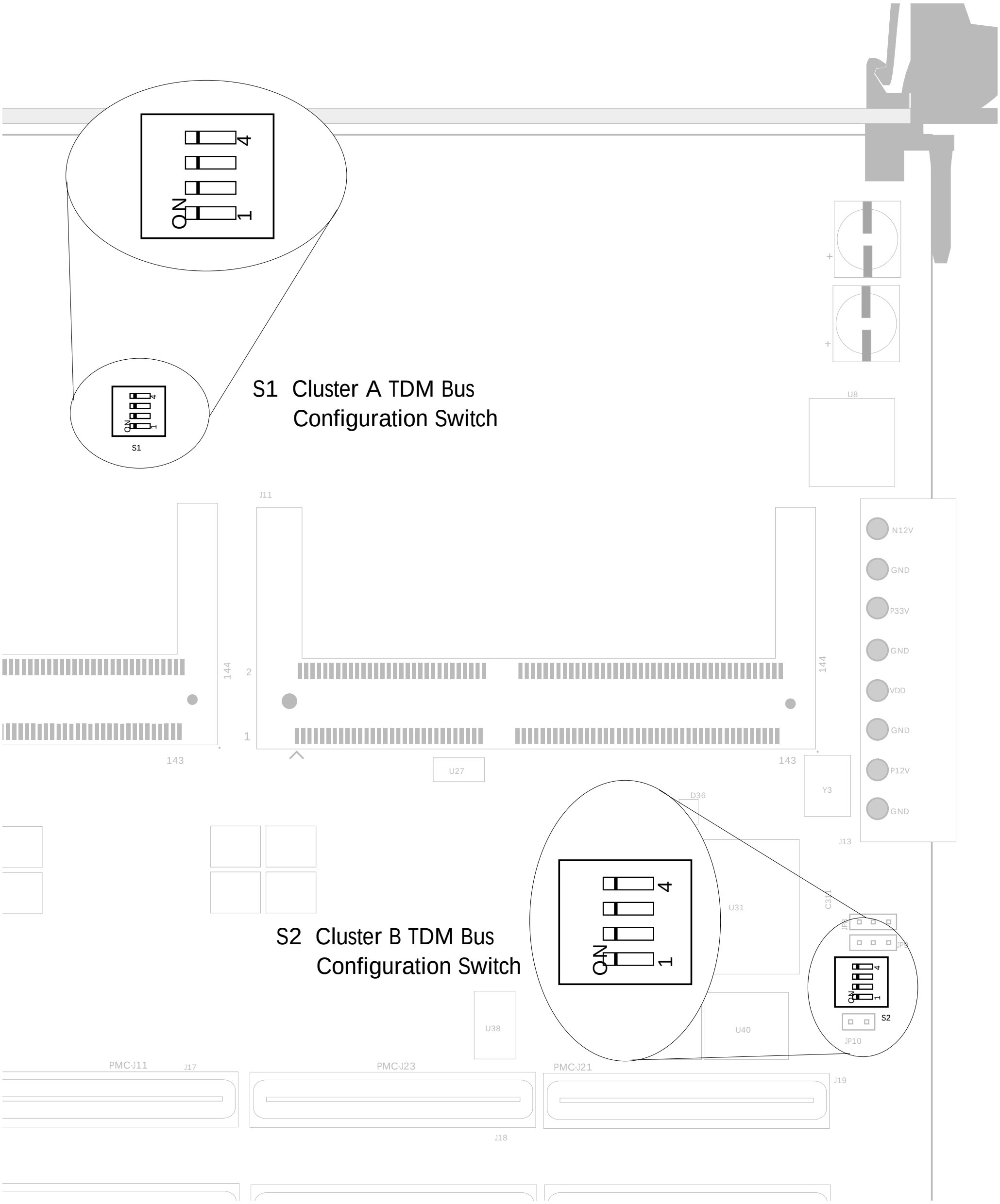


Table 2–8 *TDM Serial Port Switch Settings (S1, S2)*

Switch Pins	Signal	Settings
1/8	TD0/RD0	ON: Connects TD0 and RD0 signals together on external connector for TDM 1-wire mode. Also connects TD and RD signals on ADSP-21160. OFF: TDM 2-wire, standard mode
2/7	NC	Not connected
3/6	TD1/RD1	ON: Connects TD1 and RD1 signals together on external connector for TDM 1-wire mode. Also connects TD and RD signals on ADSP-21160. OFF: TDM 2-wire, standard mode
4/5	TFS	ON: Connects TFS signal on external connector with TFS signal on ADSP-21160. Required for use as a standard serial port with a null-modem cable. OFF: Otherwise

Figure 2–2 shows where the TDM serial port configuration switches are located on the Hammerhead-6U-cPCI board.

Figure 2–2 Location of the TDM Serial Port Configuration Switches



2.2.3 Attaching External Devices to the Board

This section explains how to attach external devices to the Hammerhead-6U-cPCI's external interfaces.

Connecting Link Port Cables to the Rear Panel

Four link ports per DSP cluster on the Hammerhead-6U-cPCI are available externally via the rear panel connectors (P3 and P5), allowing the DSPs to communicate directly with DSPs on other boards. Link ports from cluster A are available via P3, and link ports from cluster B are available via P5. Section 3.2.2 shows the location and pinout of the rear panel connectors.

BittWare offers link port cables for the rear panel connectors. Table 2–9 gives the part numbers of the link port cables that are compatible with the Hammerhead-6U-cPCI's rear panel connectors.

Table 2–9 *External Link Port Cables Available From BittWare*

Location	Cable Description	Part Number
Hammerhead-6U-cPCI (P3, P5)	Straight through, 15-pin; Coax ribbon, 12" or 36"; Connects to rear panel	CACR-S15-12 (12") CACR-S15-36 (36")

Connecting Serial Port Cables to the Rear Panel

One TDM serial port per DSP cluster on the Hammerhead-6U-cPCI is available externally via the rear panel connectors (P3 and P5), providing a communication route between the ADSP-21160 DSPs and other synchronous serial devices. The TDM serial port from cluster A is available via P3, and the TDM serial port from cluster B is available via P5. Section 3.2.2 shows the location and pinout of the rear panel connectors.

BittWare offers serial port cables for the rear panel connectors.Table 2–10 gives the part numbers of the serial port cables that are compatible with the Hammerhead-6U-cPCI's rear panel connectors.

Table 2–10 *Serial Port Cable Available from BittWare*

Location	Cable Description	Part Number
Hammerhead-6U-cPCI (P3, P5)	Straight through, 15-pin; Coax ribbon, 12" or 36"; Connects to rear panel	CACR-S15-12 (12") CACR-S15-36 (36")

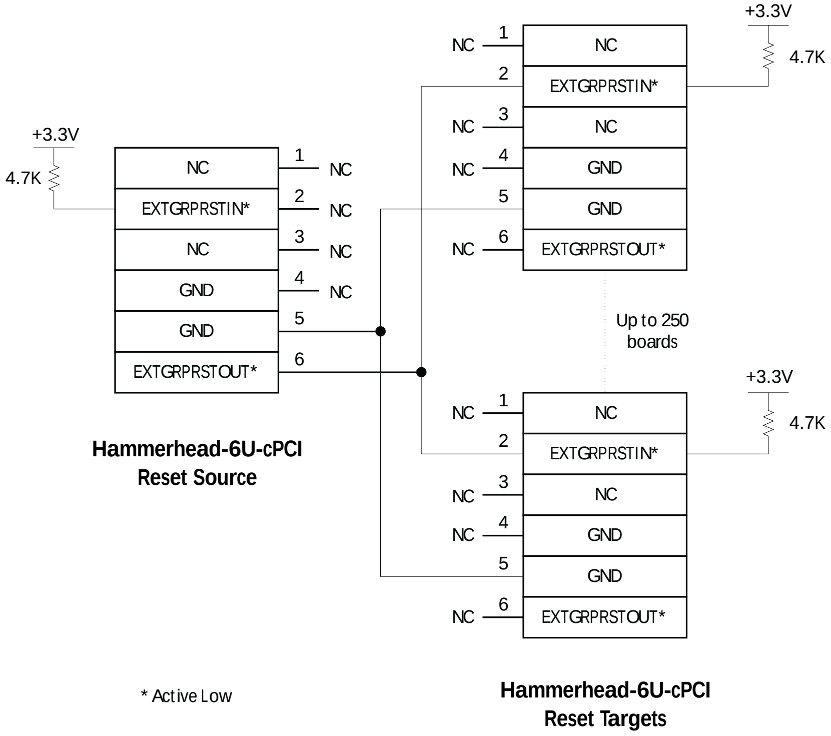
Connecting an External Reset Signal to the Reset Connectors

The external reset connectors (J12 and J14) allow the Hammerhead-6U-cPCI board to reset or be reset by other system boards. Connector J14 is the reset connector for DSP cluster A, and J12 is the reset connector for DSP cluster B. Each connector supports an input reset line to allow the Hammerhead-6U-cPCI to receive reset signals from other boards; each also supports an output reset line to allow the Hammerhead-6U-cPCI to reset other boards.

To reset the Hammerhead-6U-cPCI with an external reset connector,

1. Connect a cable (see Figure 2–3) from one of the external reset connectors (J12 or J14) on the Hammerhead-6U-cPCI board to another system board.
2. Any reset that occurs on the Hammerhead-6U-cPCI reset source causes a reset on all Hammerhead-6U-cPCI reset targets.

Figure 2–3 *Cable Details for the Hammerhead-6U-cPCI's External Reset Connectors*



Connecting an External Power Supply to the Power Connector

The Hammerhead-6U-cPCI requires a +3.3 and +5V power supply for normal operation. When operating with a PMC module, it requires +12V and –12V. The external power connector (J13) supplies +3.3V, +5V, –12V, and +12V to the Hammerhead-6U-cPCI. Section 3.2.9 gives the pinout of the connector and shows where it is located.

To connect an external power source to the Hammerhead-6U-cPCI,

1. Plug a power adapter cable into the Hammerhead-6U-cPCI's external power connector (J13). Be sure to align pin 1 (+12V) on J13 with the +12V pin on the cable.
2. Connect the remaining end of the cable to an external power source, such as a switching standalone power supply or the PC's power supply.
3. Apply power to the system.
4. Reset the Hammerhead-6U-cPCI. Section 2.5 explains in more detail how to reset the board.

Connecting an ICE Emulator to the JTAG Header

The Hammerhead-6U-cPCI is compatible with Analog Devices' ICE emulators, which are separate ISA bus, PCI bus, ethernet, or USB cards that connect to the Hammerhead-6U-cPCI's JTAG connector. The emulator provides a controlled environment for observing, debugging, and testing real-time activities in a target hardware environment by connecting directly to the target processor through its JTAG interface. Below is an overview of the steps required to connect an ICE emulator to the Hammerhead-6U-cPCI. For detailed instructions, refer to Appendix A and the user's guide for the emulator.

1. Connect the probe on the ICE card to the Hammerhead-6U-cPCI's JTAG connector.
2. Depending on the form-factor of your ICE card, either connect it to or install it in your PC.
3. Install the Hammerhead-6U-cPCI in a 6U slot in your CompactPCI chassis (see section 2.3.3) or set it up to operate in standalone mode (see Appendix B).
4. Apply power to the Hammerhead-6U-cPCI.
5. Start the emulator software on the PC.

Connecting the Hammerhead-6U-cPCI to an RS-232 Source

The Hammerhead-6U-cPCI's RS-232 ports allow you to connect the Hammerhead-6U-cPCI to an RS-232 source. The board has two RS-232 ports, one per DSP cluster. Connector J2 is the RS-232 port for DSP cluster A, and J3 is the RS-232 port for DSP cluster B.

To connect the Hammerhead-6U-cPCI to your PC via the RS-232 interface, use a ribbon cable connected to a mass-terminated DB-9 connector. The cable provides a straight-through connection from the Hammerhead-6U-cPCI's dual UART to the PC. Since the connector's pinout (see Table 3–7) is data communication equipment (DCE), you can connect it directly to equipment configured as data terminal equipment (DTE), such as a PC without a null-modem cable.

BittWare offers a host serial interface cable that connects the RS-232 connectors directly to a standard PC's DB-9 RS-232 COM port. To connect the Hammerhead-6U-cPCI to a PC with a host serial interface cable, follow the steps below.

1. Plug a serial port adapter into a 9-pin host serial interface cable, such as the cable described above.
2. Connect the other end of the serial port adapter to the RS-232 port (J2 or J3). One side of the adapter is marked with a red line. Be sure to line up the marked side with pin 1 on the RS-232 connector (Figure 3–5 shows where pin 1 is located).
3. Making sure that the PC's power is off, connect the serial interface cable to the PC.

Attaching a PMC or PMC+ Module

The Hammerhead-6U-cPCI board features two PMC+ interfaces, which allow you to attach standard PMC modules or BittWare's PMC+ I/O modules to the Hammerhead-6U-cPCI. The PMC+ interfaces feature three standard PMC connectors, which provide the 64-bit, 66 MHz PCI interface, and an additional connector that provides a TDM serial connection, four link ports, and flags and interrupts directly to the DSPs on the host board.

Warning!

BittWare uses Jn4[*] of the PMC connectors (see section 3.2.10) for our PMC+ extensions. If you are mounting a PMC card that uses the Jn4 connector and is not from BittWare, the PMC card may have incompatibilities with the PMC+ (Jn4) connector on the Hammerhead-6U-cPCI. Call BittWare technical support for assistance.

[*] Jn4 is the connector number assigned to the fourth (user-definable) PMC connector in the *IEEE P1386.1 Standard Physical and Environmental Layers for PCI Mezzanine Cards: PMC* (PMC Specification).

To attach a PMC or PMC+ module to the Hammerhead-6U-cPCI,

1. Plug the PMC(+) module onto either of the Hammerhead-6U-cPCI's PMC+ interfaces.
2. Secure the PMC(+) module to the mounting holes on the Hammerhead-6U-cPCI.
3. Check the required PCI signalling and width.
4. Set the secondary PCI interface speed jumpers (JP4 and JP7) and the secondary PCI bus signal level jumpers (JP11 and JP12) appropriately (see section 2.2.1).

2.3 Installing the Board and its Software

This section explains how to install the Hammerhead-6U-cPCI board and its software. It explains where to find installation information for the Analog Devices code development tools and the BittWare DSP21k-SFToolkit. It also explains how to install the Hammerhead-6U-cPCI board in a CompactPCI chassis.

2.3.1 Installing the Code DevelopmentTools

Installing Analog Devices' VisualDSP++

The Hammerhead-6U-cPCI is compatible with the VisualDSP++® software development toolset from Analog Devices. VisualDSP++ is an easy-to-use project management environment comprised of an integrated development environment (IDE) and debugger. The VisualDSP++ IDE includes access to Analog Devices' SHARC C compiler, C runtime library, assembler, linker, loader, simulator, and splitter.

To install the Analog Devices development tools, refer to the *VisualDSP++ IDE User's Manual* (Analog Devices, Inc.).

Installing BittWare's VisualDSP Target

BittWare's VisualDSP Target is a plug-in for VisualDSP++ that allows you to use the VisualDSP++ debugger with your BittWare board. The Target works with the VisualDSP++ debugger to allow direct communication with the DSPs on the Hammerhead-6U-cPCI. This section describes where to find installation instructions for the VisualDSP++ IDE and the BittWare VisualDSP Target.

If you will be using the VisualDSP Target debugger with the Hammerhead-6U-cPCI, you will need to install BittWare's VisualDSP Target after installing the VisualDSP++ IDE. The *VisualDSP Target User's Guide* gives detailed installation instructions.

2.3.2 Installing the Host-to-DSP InterfaceTools

This section gives a basic overview of installing the BittWare DSP21k-SF Toolkit. For detailed installation instructions, refer to the *DSP21k-SF Toolkit User's Guide*.

Overview of the DSP21k-SF Toolkit

BittWare's DSP21k-SF Toolkit is a set of libraries and utilities that enable you to develop DSP applications for the Hammerhead-6U-cPCI more quickly and easily. It contains a host interface library of C-callable functions for PC-based programs, diagnostic utilities, demo programs, and a DSP library that provides standard I/O extensions to DSP programs.

Libraries. The primary component of the DSP21k-SF Toolkit is the *Host Interface Library* (HIL). The HIL is a library of C-callable functions for DSP programs that allows you to download and start programs on the DSP, read from and write to the DSP's memory, and control other board functions.

The DSP21k-SF Toolkit also contains the *DspHost Library*, which gives your DSP programs standard I/O routines such as screen display, keyboard input, and disk file access. It consists of a library of standard I/O routines that you link into your DSP program and a program that runs on the PC to act as an I/O server. DspHost is an excellent tool for porting existing C applications to the DSP.

Diagnostic Utilities. *Diag21k* is a character-based diagnostic utility that lets you interactively download DSP programs, start and stop their operation, and access DSP memory.

The *DSP Board Automated Diagnostic* (DspBad) is a command-line operated utility that verifies the ability to communicate with the DSP from the host, tests the memory of the board, and confirms the DSPs' ability to load and run a program.

Installing the DSP21k-SF Toolkit Libraries and Utilities

Run the DSP21k-SF Toolkit setup program to install the DSP21k-SF Toolkit libraries and utilities. The *DSP21k-SF Toolkit User's Guide* explains the procedure in more detail.

2.3.3 Installing the Hammerhead-6U-cPCI in a CompactPCI chassis

After installing the DSP21k-SF Toolkit, you can install your Hammerhead-6U-cPCI board in a CompactPCI chassis. The Hammerhead-6U-cPCI plugs into a 6U slot in a CompactPCI chassis. To install the Hammerhead-6U-cPCI in a CompactPCI chassis, follow the instructions below.

Note

After *installing the Hammerhead-6U-cPCI in a CompactPCI chassis, you must run the BittWare Configuration Manager (see section 2.3.4).*

Warning!

When installing the Hammerhead-6U-cPCI in a cPCI chassis, make sure that the chassis is compatible with the Hammerhead-6U-cPCI's backplane connectors (P2, P3, and P5). P2 provides the 64-bit cPCI extensions and must be connected to a 64-bit backplane to function properly. P3 and P5 provide BittWare rear panel I/O and must be properly connected to P3 and P5 on the backplane to function properly.

1. Remove the Hammerhead-6U-cPCI from its anti-static packaging.
2. Power down the CompactPCI chassis.
3. Find a 6U bus-mastering peripheral slot in your CompactPCI chassis.
4. Line up the top and bottom edges of your Hammerhead-6U-cPCI board with the slot guides in the chassis.
5. Making sure that the ejectors are in the unlocked position (the top ejector should be "up," and the bottom ejector should be "down"), slide the board into the slot.
6. Push the board into the chassis until it stops.
7. Lock the board in place by pushing the top ejector "down" and the bottom ejector "up".
8. Power up the system.

2.3.4 Verifying Board Configuration

After installing the board and powering up, run the BittWare Configuration Manager to ensure that the board is properly configured. The BittWare Configuration Manager is a utility included with the DSP21k-SF Toolkit that allows you to install, uninstall, or get and set properties for the Hammerhead-6U-cPCI board. The *DSP21k-SF Toolkit User's Guide* explains how to run the BittWare Configuration Manager.

Note

Because the Hammerhead-6U-cPCI has two SharcFIN devices, when you are installing the Hammerhead-6U-cPCI, the DSP21k-SF Toolkit will recognize it as two independent PCI boards.

2.4 Testing the Installation

This section explains several options for testing the board to make sure it is working properly after installing it. It discusses two DSP21k-SF Toolkit utilities, Diag21k and DspBad, that allow you to test communication between the DSPs and the host. It also discusses the example files included with the board, which allow you to test various components of the board.

2.4.1 Testing the Installation with the Hammerhead Example Files

The example software provided with the Hammerhead-6U-cPCI contains examples that demonstrate how to use the various features of your board and software. The examples are located in the `examples` directory of the Hammerhead-6U-cPCI CD-ROM.

2.4.2 Testing the Installation With DSP21k-SF Diagnostic Utilities

The DSP21k-SF Toolkit contains two utilities for testing a DSP board to make sure it is operating properly: the DSP Board Automated Diagnostic (DspBad) and Diag21k.

- *DspBad* is a command-line-operated utility that verifies the ability to communicate with the DSP board from the PC, tests the memory of the board, and confirms the DSP's ability to load and run a program.
- *Diag21k* is a character-based diagnostic utility that you start from the MS-DOS command prompt. It lets you interactively download DSP programs, start and stop their operation, and access DSP memory.

Testing the Board with DspBad

To test a processor with DspBad, enter the following command at a command prompt:

```
C:>dspbad -b<N> <enter>
```

or

```
C:>dspbad -d<N> -i<N> <enter>
```

The **`<N>`** in **`-b<N>`** represents the processor number[1]. The **`<N>`** in **`-d<N>`** represents the device number. The **`<N>`** in **`-i<N>`** represents the processor ID number of the processor you want to open on the specified device. The *DSP21k-SF Toolkit User's Guide* explains DspBad commands in more detail.

Testing the Board with Diag21k

The example below shows you basic Diag21k commands you can use to test the Hammerhead-6U-cPCI's memory and load and run a DSP program. Be sure to follow the example steps below in the order in which they appear. The *DSP21k-SF Toolkit User's Guide* describes the Diag21k commands in more detail.

Step 1: Start Diag21k

a. The Diag21k program is located in the `dsp21ksf\bin` directory. Start the program from the DOS prompt. The **`-b`** switch tells Diag21k which processor you will access. If you do not specify a processor number with **`-b`** (or both **`-d`** and **`-i`**), Diag21k will use all processors that are installed in your PC.

```
C:\DSP21KSF\BIN>diag21k -b1

C:\DSP21KSF\BIN>diag21k -d0 -i1
```

Both of the command line options above tell Diag21k to open the first processor on device 0.

b. Diag21k will start and display a copyright banner. The command prompt shows the active board number in square brackets.

```
DSP21K Interactive Diagnostic Utility
32-bit version for SharcFIN boards under Windows 95/98 and
Windows NT/2000. Release 6.12 [ DSP21K-SF, December 2000
15:54:36 ], Version 3.93 Copyright (c) 1992-2000 BittWare,
Inc. All rights reserved.

Type "?" for a list of commands.
        Available DSP numbers: 1 2 3 4

Opened 4 DSPs.

Current DSP:         #1, processor 1 on Hammerhead (device 0)
```

[1] The processor number is the *device number * 10 + id number*. See the *DSP21k-SF User's Manual* for further explanation.

Step 2: Display Board Information

a. Use the board information command to display information about the Hammerhead-6U-cPCI's DSPs.

diag21k[1]>bi

```
Board/Processor Information for DSP #1  (Not Started)

  Board Type: (38) Hammerhead             DSP Type: (7) ADSP-21160
  Multi-proc ID: 1                 Interrupt Number: 11

  BAR0: 0x0c800000  Size: 0x00000200    BAR3: 0x0c800200  Size: 0x00000100
  BAR1: 0x0c400000  Size: 0x00400000    BAR4: 0x0a000000  Size: 0x01000000
  BAR2: 0x80000000  Size: 0x02000000    BAR5:             Size: 0x0

Int. Mem: 4 Mbit    IMDW0: 32-bit data     IMDW1: 32-bit data
  MMS WS: 0   Ext Bank Size: 32768 KW (MSIZE = 12)  DRAM PgSz: 256 W
Bank 0: Start = 0x00800000  Width = 32 bits  Depth = 32768 KW WS/WM = 1/2
Bank 1: Start = 0x02800000  Width =  8 bits  Depth =  2048 KW WS/WM = 7/0
Bank 2: Start = 0x04800000  WS/WM = 1/2
Bank 3: Start = 0x06800000  WS/WM = 7/0
Unbnkd: Start = 0x08800000  WS/WM = 7/0

  Program loaded: (none)
  Labels: *not defined*
```

b. Notice the memory size information for the external memory banks 0 and 1. The memory test command (**`mt`**) uses these values when it performs various tests on different regions of the ADSP-21160's memory.

Step 3: Test the Hammerhead-6U-cPCI's Memory

Now that you have found the memory bank settings, you can test all of the Hammerhead-6U-cPCI's memory with the following commands.

a. To ensure that neither of the processors is executing programs that might change memory while you are testing it, use the following command to reset the board:

diag21k[1]>br

```
        Board reset
```

b. Next, use the following command to configure the processor you selected to access external memory (MSIZE and WAIT settings from the environment variable):

diag21k[1]>pc

```
        processor configured
```

c. Now use the following command to test all memory banks:

```
diag21k[1]>mt aa
```

```
Program Memory Test at 0x040000, Size: 0xa000 48-bit Words
        Self-Address.............. ok
        Self-Address Complement... ok
        Checkerboard A............ ok
        Checkerboard 5............ ok
        All Bits Clear............ ok
        All Bits Set.............. ok
        Random Numbers............ ok
Data Memory Test at 0x050000, Size: 0x10000 32-bit Words
        Self-Address.............. ok
        Self-Address Complement... ok
        Checkerboard A............ ok
        Checkerboard 5............ ok
        All Bits Clear............ ok
        All Bits Set.............. ok
        Random Numbers............ ok
External Bank 0 Test at 0x800000, Size: 0x2000000 32-bit
Words
        Self-Address.............. ok
        Self-Address Complement... ok
        Checkerboard A............ ok
        Checkerboard 5............ ok
        All Bits Clear............ ok
        All Bits Set.............. ok
        Random Numbers............ ok
```

Step 4: Load and Execute a Program

Now that you have tested the memory, you know that Diag21k can successfully communicate with the Hammerhead-6U-cPCI board. Next, load a program and execute it.

a. The `dsp21ksf\etc` directory contains an example program that calculates the first twenty prime numbers. The source code is in the `examples\21160\prime160` directory. Load the pre-compiled executable file with the file load (**`fl`**) command.

```
diag21k[1]>fl\dsp21ksf\etc\prm21160
```

```
        “\dsp21ksf\etc\prm21160.dxe” loaded
```

b. Now that Diag21k has downloaded the executable file into the ADSP-21160’s memory and holds the processor in reset, start the processor with the processor start command.

```
diag21k[1]>ps
```

```
        processor running
```

c. To see the results of the primes program, examine the variable that contains the calculated prime numbers. The C program `primes.c` defines a global array called primes, which is stored in data memory. The memory read command can use global labels to locate variables and functions. Notice that the C compiler adds an underscore to global labels.

```
diag21k[0]>mr li _primes 20
```

```
     DATA_SRAM [00050040] =      2
     DATA_SRAM [00050041] =      3
     DATA_SRAM [00050042] =      5
     DATA_SRAM [00050043] =      7
     DATA_SRAM [00050044] =     11
     DATA_SRAM [00050045] =     13
     DATA_SRAM [00050046] =     17
     DATA_SRAM [00050047] =     19
     DATA_SRAM [00050048] =     23
     DATA_SRAM [00050049] =     29
     DATA_SRAM [0005004A] =     31
     DATA_SRAM [0005004B] =     37
     DATA_SRAM [0005004C] =     41
     DATA_SRAM [0005004D] =     43
     DATA_SRAM [0005004E] =     47
     DATA_SRAM [0005004F] =     53
     DATA_SRAM [00050050] =     59
     DATA_SRAM [00050051] =     61
     DATA_SRAM [00050052] =     67
     DATA_SRAM [00050053] =     71
```

Step 5: Test the Remaining Processors

a. To test the remaining ADSP-21160 processors, select one of them with the board select command.

```
diag21k[1]>ds 2
```

```
   Current DSP:     #2, processor 2 on Hammerhead (device 0)
```

b. With another processor selected, you can use the same commands as before to load a program and start the processor.

```
diag21k[2]>fl ..\etc\prm21160
```

```
    "..\etc\prm21160.dxe" loaded
```

```
diag21k[2]>ps
```

```
    processor running
```

```
diag21k[2]>mr li _primes 20
        DATA_SRAM [00050040] =      2
        DATA_SRAM [00050041] =      3
        DATA_SRAM [00050042] =      5
        DATA_SRAM [00050043] =      7
        DATA_SRAM [00050044] =     11
        DATA_SRAM [00050045] =     13
        DATA_SRAM [00050046] =     17
        DATA_SRAM [00050047] =     19
        DATA_SRAM [00050048] =     23
        DATA_SRAM [00050049] =     29
        DATA_SRAM [0005004A] =     31
        DATA_SRAM [0005004B] =     37
        DATA_SRAM [0005004C] =     41
        DATA_SRAM [0005004D] =     43
        DATA_SRAM [0005004E] =     47
        DATA_SRAM [0005004F] =     53
        DATA_SRAM [00050050] =     59
        DATA_SRAM [00050051] =     61
        DATA_SRAM [00050052] =     67
        DATA_SRAM [00050053] =     71
```

Step 6: Exit Diag21k

To exit Diag21k and reset the processor you have selected, use the quit command.

```
diag21k[0]>q
          exiting...resetting processor(s)
          C:\DSP21KSF\BIN>
```

2.5 Resetting the Board

This section explains three methods of resetting the Hammerhead-6U-cPCI, including:

- with the watchdog timer
- with an external reset switch
- via the PCI interface

2.5.1 Resetting the Board with the Watchdog Timer

The Hammerhead-6U-cPCI features two watchdog timers, one for each DSP cluster. The watchdog timers help to ensure that the Hammerhead-6U-cPCI is operating properly. They are also useful for standalone applications that need to restart when certain errors occur or a program crashes.

The Watchdog Configuration register, which is located in the SharcFIN ASIC, enables and disables the watchdog timer (see section 6.5.4). The register is located at offset 0x0000 0043 from the base of the ADSP-21160s' memory select line MS2.

How the Watchdog Timer Functions When Disabled

The watchdog is disabled after a reset occurs. When the watchdog is disabled, the SharcFIN chip constantly strobes the timer to keep it from elapsing. Since it is constantly being strobed, the watchdog timer will not time-out regardless of whether the program fails.

How the Watchdog Timer Functions When Enabled

When enabled, the watchdog timer must be reset before it expires to prevent a board reset from occurring. The watchdog timer is reset every time FLAG1 from a configured processor toggles from 0 to 1 or from 1 to 0. The FLAG1 signals are flags that are under program control and can strobe the watchdog timer to prevent it from elapsing.

Six bits in the Watchdog Configuration register control the watchdog timer. The first two bits enable it and select its time-out time, and the next four bits determine which flag the watchdog will respond to (see section 6.5.4). The Watchdog Configuration register is a write-once register; therefore, once the watchdog is enabled it cannot be disabled except by a board reset.

If the watchdog timer is enabled, the DSP program must toggle FLAG1 within the given time frame. The Watchdog Configuration register allows you to select the watchdog’s time-out time (see 6.5.4). If the watchdog timer elapses, it will generate a system reset and the normal boot process will begin.

2.5.2 Resetting the Board via the External Reset Connector

The external reset connectors (J12 and J14) allow the Hammerhead-6U-cPCI board to reset or be reset by other system boards. Connector J14 is the reset connector for DSP cluster A, and J12 is the reset connector for DSP cluster B. The connectors support an input reset line to allow the Hammerhead-6U-cPCI to receive reset signals from other boards. They also support an output reset line to allow the Hammerhead-6U-cPCI to reset other boards.

To reset the Hammerhead-6U-cPCI with the external reset connector, follow the instructions in “Connecting an External Reset Signal to the Reset Connectors” on page 22.

2.5.3 Resetting the Board via the PCI Interface

A register bit in the SharcFIN ASIC allows the board to be reset from the host PC. This bit is B0 of the register located at Byte offset 0x58 from the base of Base Address Register 0 (BAR0). When the register is written, all components on the board will be reset. Complete this reset procedure regardless of other reset methods to ensure hardware and software initialization. Refer to the *SharcFIN ASIC User’s Manual* for complete details on this register.

2.5.4 Resetting Any Attached PMC+ Cards

The PMC+ interfaces on the Hammerhead-6U-cPCI feature a reset line that allows the Hammerhead-6U-cPCI to reset a PMC+ board that is attached to it.

Chapter 3

Overview of the Hardware Components

This chapter shows where the Hammerhead-6U-cPCI's major components and connectors are located and briefly describes their function. Section 3.1 describes the layout and function of the major components, section 3.2 describes the external connectors, and section 3.3 describes the configuration jumpers. This chapter covers the following components and connectors:

- PCI-to-PCI bridges
- SharcFIN ASICs
- ADSP-21160 DSPs
- Flash memory
- SDRAM
- On-board oscillators
- Dual RS-232 UARTs
- Watchdog timers
- LEDs
- CompactPCI interface
- Rear panel I/O
- JTAG header
- External power connector
- External reset connectors
- Flag I/O connectors
- External interrupt inputs
- PMC+ interfaces
- Configuration jumpers and switches

3.1 Layout and Function of the Major Components

This section briefly describes the function of each major component on the board and shows where each is located. Figure 3–1 shows the components on the top side of the board.

Figure 3–1 *Location of the Hammerhead-6U-cPCI's Major Components (Top)*

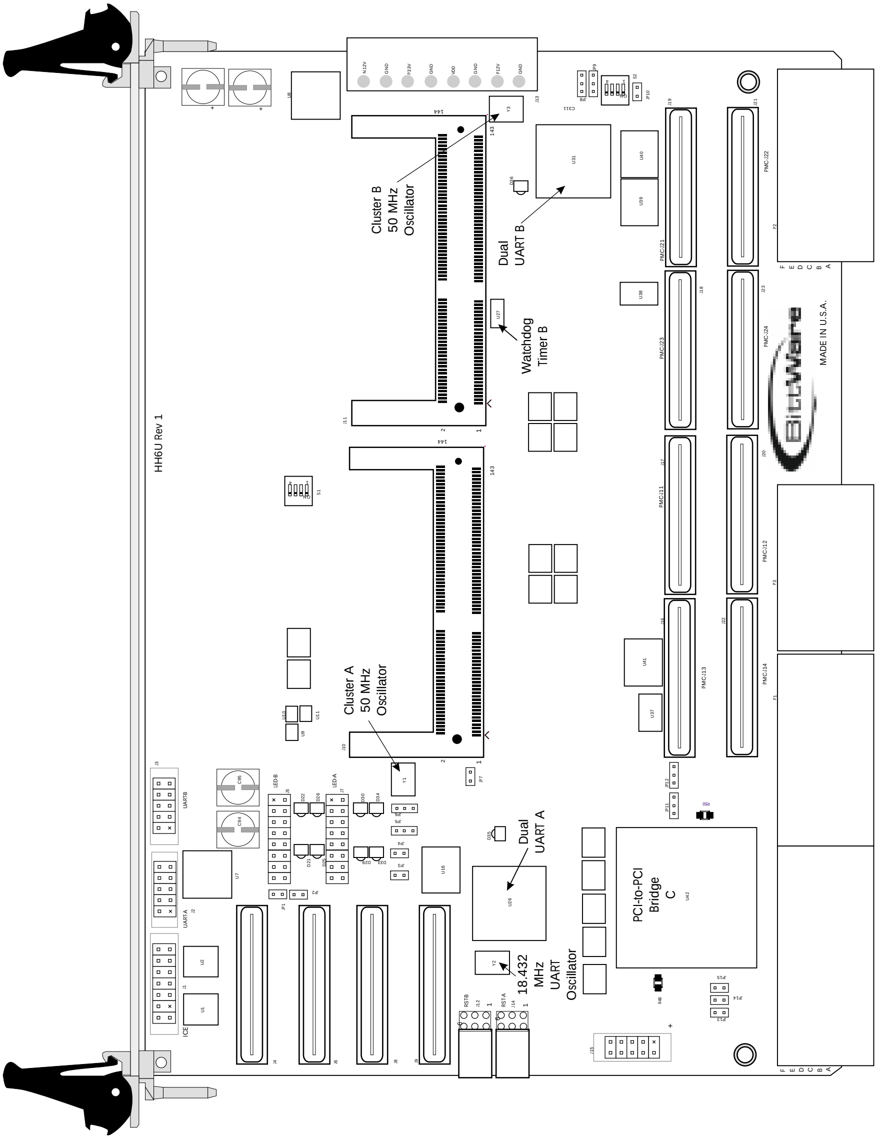
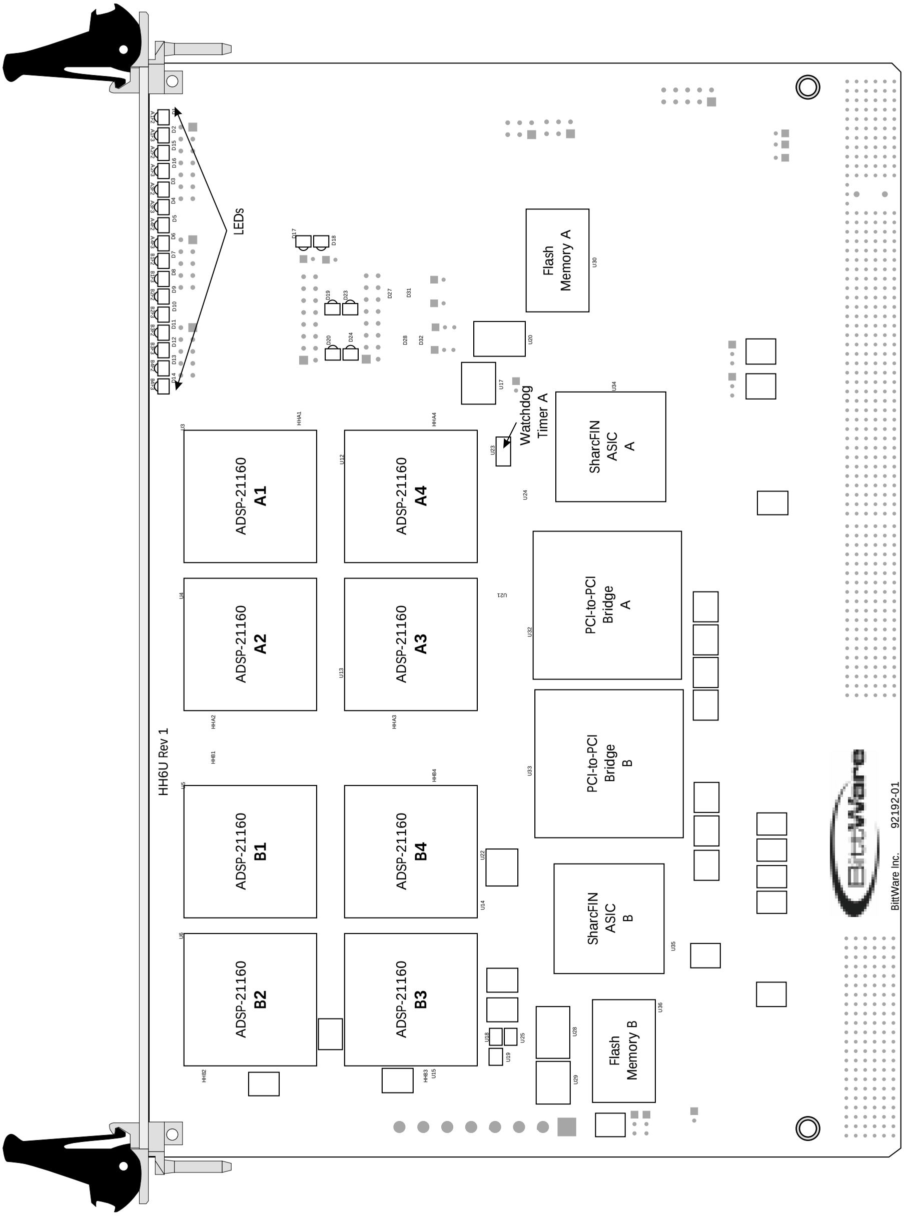


Figure 3–2 shows the components on the bottom side of the Hammerhead-6U-cPCI.

Figure 3–2 *Location of the Hammerhead-6U-cPCI’s Major Components (Bottom)*



3.1.1 PCI-to-PCI Bridges

The Hammerhead-6U-cPCI has three PCI-to-PCI bridge chips (Intel 21154-BC from Intel Corporation), which provide the bridges between the primary PCI bus and the three secondary PCI buses. One provides the bridge for cluster A, one provides the bridge for cluster B, and the third provides the bridge between the two clusters and the host interface. Section 4.3.2 explains the function of each bridge in more detail.

3.1.2 SharcFIN ASICs

The Hammerhead-6U-cPCI features two SharcFIN ASICs, one per DSP cluster. The SharcFIN ASIC flexibly interfaces the ADSP-21160 DSPs to a wide range of the Hammerhead-6U-cPCI's interfaces, including 64/66 MHz PCI bus (rev. 2.2 compliant), SDRAM, UART, I^2C™ serial ports, Flash, and a general-purpose expansion bus (the 8-bit peripheral bus). The SharcFIN also provides a feature-rich set of DMA functions and interrupt options to support very high-speed, real-time data flow with a minimum of processor overhead. The following is a list of the SharcFIN's features:

- 64-bit, 66 MHz PCI rev. 2.2 compliant interface (528 MB/s burst)
- Connected to 64-bit, 40 MHz ADSP-21160 cluster bus
- Connected to the Hammerhead-6U-cPCI's peripheral bus
  - 8 bits wide @ 20 MHz
  - Accessible from the ADSP-21160 cluster bus and the PCI bus
  - Flash interface for SHARC boot and non-volatile data storage
- Six independent FIFOs (2.4 KB total)
  - Four DMA buffers, 64×64 each (two transmit, two receive)
  - Two target buffers, 32×64 write, 16×64 read
- Direct, single PCI access from the ADSP-21160 cluster bus
- 16-byte configurable PCI mailbox registers
- I_2O™ V1.5 compliant
- Programmable interrupt multiplexer: 10 inputs, 7 outputs (one of each dedicated to PCI)
- SDRAM controller on ADSP-21160 cluster bus; supports up to 512 MB
- Standard UART and I^2C interface

3.1.3 Analog Devices ADSP-21160 DSPs

The Hammerhead-6U-cPCI features eight ADSP-21160 SHARC processors from Analog Devices, arranged in two clusters of four processors. The Hammerhead-6U-cPCI's processors have a total of 4800 MFLOPs of processing power and operate at 80 MHz. Each processor supports two I^2C serial ports, 14 DMA channels, four flags, three interrupts, and six link ports. Each processor also features 4 Mbits of dual-ported on-chip SRAM.

3.1.4 Memory

Flash Memory

Two 2 MB banks of Flash memory, one for each DSP cluster, store boot programs that the processor can load, enabling the Hammerhead-6U-cPCI to boot without a host computer (see section 5.2). The ADSP-21160s can also read, write, and erase the Flash, which allows them to use it as non-volatile storage space.

SDRAM

The Hammerhead-6U-cPCI has up to 512 MB of SDRAM for banked external memory. For each DSP cluster, it features a standard 144-pin SODIMM that supports 64, 128, 256, or 512 MB SDRAM modules. The SDRAM is available to the ADSP-21160 DSPs at 40 MHz via the ADSP-21160 cluster buses.

3.1.5 On-Board Oscillators

The Hammerhead-6U-cPCI has three on-board oscillators: one for each DSP cluster, and one for the RS-232 UARTs.

SHARC Oscillators

A 40 MHz system oscillator chip (Y1 and Y3) for each DSP cluster provides the 1× clock for the four ADSP-21160 DSPs in the cluster. Figure 3–1 shows where the oscillators are located.

UART Oscillator

An 18.432 MHz oscillator chip (Y2) provides the clock for the RS-232 UARTs. Figure 3–1 shows where it is located.

3.1.6 Dual UARTs

The Hammerhead-6U-cPCI features a dual RS-232 UART for each DSP cluster. The UARTs interface serial data from the RS-232 ports to the ADSP-21160 DSPs. Figure 3–1 shows where the UARTs are located, and section 3.2.4 describes the RS-232 ports.

3.1.7 Watchdog Timers

The Hammerhead-6U-cPCI has a watchdog timer for each DSP cluster. The watchdog timers help to ensure that the Hammerhead-6U-cPCI is operating properly. They are also useful for standalone applications that need to restart when certain errors occur or a program crashes.

3.1.8 LEDs

The Hammerhead-6U-cPCI has sixteen user LEDs, which are available to indicate certain conditions in the software or to provide feedback. Two LEDs are connected to each ADSP-21160, each LED corresponding to a different ADSP-21160 flag. Section 4.2.5 shows the LEDs' connections to the ADSP-21160 flags.

3.2 Layout and Function of the External Connectors

This section briefly describes the function of the external connectors on the board and shows where they are located (see Figure 3–3 below). It also provides the pinouts for the connectors.

Figure 3–3 *Layout of the External Connectors*

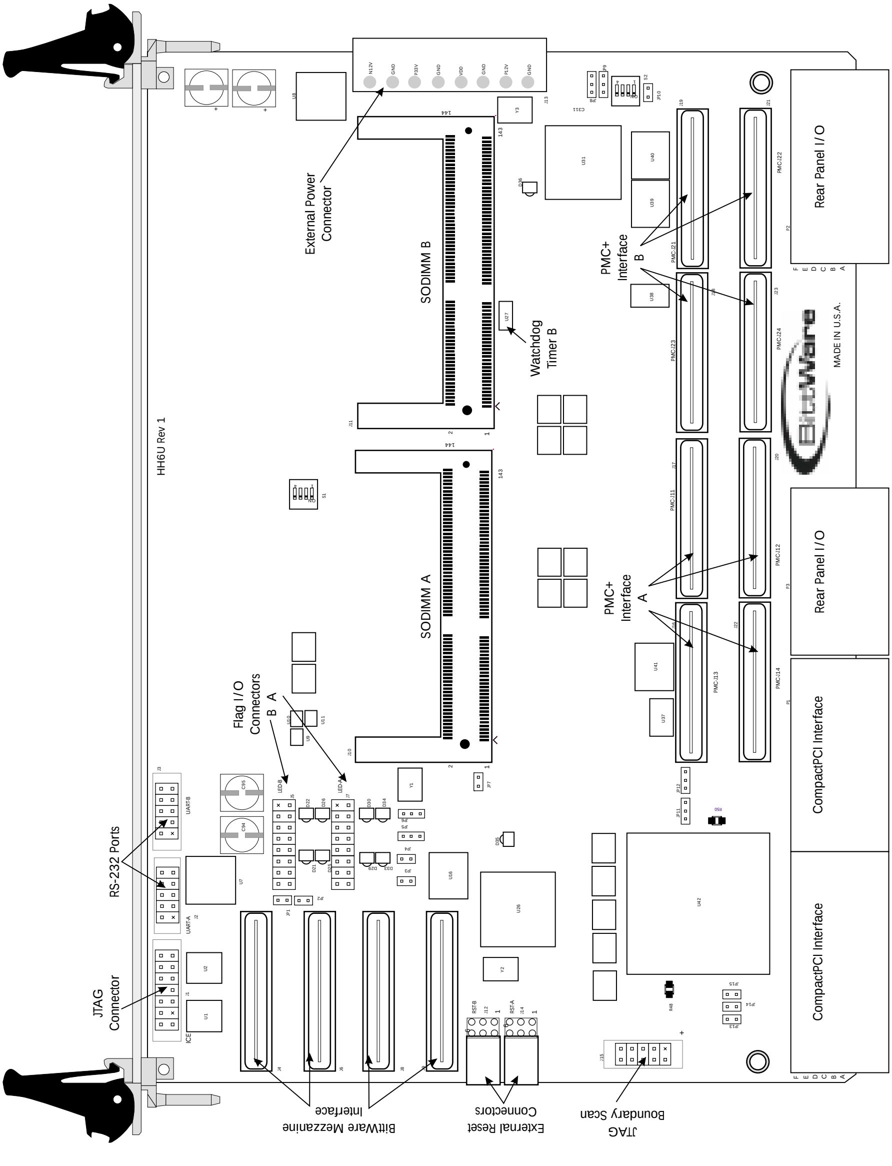


Table 3–1 below gives an overview of the connectors and their functions.

Table 3–1 *Overview of the External Connectors*

Connector	Ref Des	Type	Description
CompactPCI	P1, P2	cPCI	64-bit, 66 MHz CompactPCI interface
Rear Panel I/O A	P3	cPCI	Rear panel I/O with external link and serial ports for DSP cluster A
Rear Panel I/O B	P5	cPCI	Rear panel I/O with external link and serial ports for DSP cluster B
JTAG Header	J1	14-pin	Connection for ICE in-circuit emulator
RS-232 Port A	J2	10-pin	External RS-232 serial interface to DSP cluster A via UART A
RS-232 Port B	J3	10-pin	External RS-232 serial interface to DSP cluster B via UART B
BittWare Mezzanine	J4, J6, J8, J9	64-pin	BittWare mezzanine interface for additional SDRAM or I/O modules
Flag I/O B	J5	16-pin	Access to FLAG2 and FLAG3 on each processor in DSP cluster B
Flag I/O A	J7	16-pin	Access to FLAG2 and FLAG3 on each processor in DSP cluster A
SODIMM A	J10	144-pin	Connection for standard 144-pin SODIMM SDRAM modules for DSP cluster A
SODIMM B	J11	144-pin	Connection for standard 144-pin SODIMM SDRAM modules for DSP cluster B
External Reset B	J12	6-pin	Connection for external reset signals for DSP cluster B
External Power	J13	8-pin	Connection for +3.3V, +5V, and ±12V external power supply
External Reset A	J14	6-pin	Connection for external reset signals for DSP cluster A
JTAG Boundary Scan	J15	10-pin	Manufacturer use only
PMC+ Interface A	J16, J17, J20, J22	64-pin	Connection for BittWare PMC+ I/O module or for standard PMC module
PMC+ Interface B	J18, J19, J21, J23	64-pin	Connection for BittWare PMC+ I/O module or for standard PMC module

3.2.1 CompactPCI Connector

Rear panel connectors P1 and P2 provide the 64-bit, 66 MHz CompactPCI interface. Each connector consists of six rows of 22 pins. Table 3–2 and Table 3–3 give the connector pinouts.

Table 3–2 *CompactPCI Interface Pinout (P1: A–F)*

		P1:A	P1:B	P1:C	P1:D	P1:E	P1:F
22	GND	VDD	REQ64#	BRSV	P33V	VDD	GND
21	GND	AD1	VDD	V(I/O)	AD0	ACK64#	GND
20	GND	P33V	AD4	AD3	VDD	AD2	GND
19	GND	AD7	GND	P33V	AD6	AD5	GND
18	GND	P33V	AD9	AD8	M66EN#	C/BE0#	GND
17	GND	AD12	GND	V(I/O)	AD11	AD10	GND
16	GND	P33V	AD15	AD14	GND	AD13	GND
15	GND	SERR#	GND	P33V	PAR	C/BE1#	GND
14	GND	P33V	SDONE	SBO#	GND	PERR#	GND
13	GND	DEVSEL#	GND	V(I/O)	STOP#	LOCK#	GND
12	GND	P33V	FRAME#	IRDY#	GND	TRDY#	GND
11	GND	AD18	AD17	AD16	GND	C/BE2#	GND
10	GND	AD21	GND	P33V	AD20	AD19	GND
9	GND	C/BE3#	IDSEL	AD23	GND	AD22	GND
8	GND	AD26	GND	V(I/O)	AD25	AD24	GND
7	GND	AD30	AD29	AD28	GND	AD27	GND
6	GND	REQ#	GND	P33V	CLK	AD31	GND
5	GND	BRSV	BRSV	RST#	GND	GNT#	GND
4	GND	BRSV	GND	V(I/O)	INTP	INTS	GND
3	GND	INTA#	INTB#	INTC#	VDD	INTD#	GND
2	GND	TCK	VDD	TMS	TD0	TD1	GND
1	GND	VDD	N12V	TRST#	P12V	VDD	GND
Row	Z	A	B	C	D	E	F

Table 3–3 *CompactPCI Interface Pinout (P2: A–F)*

		P2:A	**P2:B**	**P2:C**	**P2:D**	**P2:E**	**P2:F**
22	GND	USR	USR	USR	USR	USR	GND
21	GND	USR	USR	USR	USR	USR	GND
20	GND	USR	USR	USR	USR	USR	GND
19	GND	USR	USR	USR	USR	USR	GND
18	GND	USR	USR	USR	USR	USR	GND
17	GND	BRSV	GND	PRST#	REQ6#	GNT6#	GND
16	GND	BRSV	BRSV	DEG#	GND	BRSV	GND
15	GND	BRSV	GND	FAL#	REQ5#	GNT5#	GND
14	GND	AD35	AD34	AD33	GND	AD32	GND
13	GND	AD38	GND	V(I/O)	AD37	AD36	GND
12	GND	AD42	AD41	AD40	GND	AD39	GND
11	GND	AD45	GND	V(I/O)	AD44	AD43	GND
10	GND	AD49	AD48	AD47	GND	AD46	GND
9	GND	AD52	GND	V(I/O)	AD51	AD50	GND
8	GND	AD56	AD55	AD54	GND	AD53	GND
7	GND	AD59	GND	V(I/O)	AD58	AD57	GND
6	GND	AD63	AD62	AD61	GND	AD60	GND
5	GND	C/BE5#	GND	V(I/O)	C/BE4#	PAR64	GND
4	GND	V(I/O)	BRSV	C/BE7#	GND	C/BE6#	GND
3	GND	CLK4	GND	GNT3#	REQ4#	GNT4#	GND
2	GND	CLK2	CLK3	SYSEN#	GNT2#	REQ3#	GND
1	GND	CLK1	GND	REQ1#	GNT1#	REQ2#	GND
Row	Z	A	B	C	D	E	F

3.2.2 Rear Panel I/O

Connectors P3 and P5 on the rear panel provide rear panel I/O for the DSP clusters. P3 provides rear panel I/O for DSP cluster A, and P5 provides rear panel I/O for cluster B. Each connector consists of six rows of 22 pins and provides four link ports, a TDM serial port, flags, and interrupts. Table 3–4 and Table 3–5 give the connector pinouts.

Table 3–4 *Rear Panel I/O Pinout (P3: A–F)*

		P3:A	P3:B	P3:C	P3:D	P3:E	P3:F	
19	GND	HA1_F2	HA2_F2	GND	HA3_F2	HA4_F2	GND	General purpose*
18	GND	P33V	VDD	GND	$\overline{\text{HA\_EXTGRPRSTOUT}}$	$\overline{\text{HA\_EXTGRPRSTIN}}$	GND	
17	GND	$\overline{\text{HA1\_I2}}$ HA2_F3	$\overline{\text{HA2\_I2}}$ HA1_F3	GND	$\overline{\text{HA3\_I2}}$ HA4_F3	$\overline{\text{HA4\_I2}}$ HA3_F3	GND	
16	GND	HA_EXTTXD1	HA_EXTCTS1	GND	$\overline{\text{HA\_EXTRTS1}}$	HA_EXTRXD1	GND	
15	GND	HA_TDMRD1	HA_TDMTRC1	GND	NC	HA_TDMTD1	GND	External TDM serial port
14	GND	NC	HA_TDMRFS1	GND	XHA_TDMTFS1	NC	GND	
13	GND	NC	NC	GND	NC	NC	GND	
12	GND	EA_L4ACK	EA_L4CLK	GND	NC	NC	GND	External link port 4
11	GND	EA_L4DAT4	EA_L4DAT5	GND	EA_L4DAT6	EA_L4DAT7	GND	
10	GND	EA_L4DAT0	EA_L4DAT1	GND	EA_L4DAT2	EA_L4DAT3	GND	
9	GND	EA_L3ACK	EA_L3CLK	GND	NC	NC	GND	External link port 3
8	GND	EA_L3DAT4	EA_L3DAT5	GND	EA_L3DAT6	EA_L3DAT7	GND	
7	GND	EA_L3DAT0	EA_L3DAT1	GND	EA_L3DAT2	EA_L3DAT3	GND	
6	GND	EA_L2ACK	EA_L2CLK	GND	NC	NC	GND	External link port 2
5	GND	EA_L2DAT4	EA_L2DAT5	GND	EA_L2DAT6	EA_L2DAT7	GND	
4	GND	EA_L2DAT0	EA_L2DAT1	GND	EA_L2DAT2	EA_L2DAT3	GND	
3	GND	EA_L1ACK	EA_L1CLK	GND	NC	NC	GND	External link port 1
2	GND	EA_L1DAT4	EA_L1DAT5	GND	EA_L1DAT6	EA_L1DAT7	GND	
1	GND	EA_L1DAT0	EA_L1DAT1	GND	EA_L1DAT2	EA_L1DAT3	GND	
Row	Z	A	B	C	D	E	F	

* Power, flags, interrupts, reset, RS-232

Table 3–5 *Rear Panel I/O Pinout (P5: A–F)*

		P5:A	P5:B	P5:C	P5:D	P5:E	P5:F	
22	GND	P12V	N12V	GND	N12V	P12V	GND	General purpose[*]
21	GND	VDD	VDD	GND	VDD	VDD	GND	
20	GND	P33V	P33V	GND	P33V	P33V	GND	
19	GND	HB1_F2	HB2_F2	GND	HB3_F2	HB4_F2	GND	
18	GND	P33V	VDD	GND	$\overline{\text{HB\_EXTGRPRSTOUT}}$	$\overline{\text{HB\_EXTGRPRSTIN}}$	GND	
17	GND	$\overline{\text{HB1\_I2}}$ HB2_F3	$\overline{\text{HB2\_I2}}$ HB1_F3	GND	$\overline{\text{HB3\_I2}}$ HB4_F3	$\overline{\text{HB4\_I2}}$ HB3_F3	GND	
16	GND	HB_EXTTXD1	$\overline{\text{HB\_EXTCTS1}}$	GND	$\overline{\text{HB\_EXTRTS1}}$	HB_EXTRXD1	GND	
15	GND	HB_TDMRD1	HB_TDMTRC1	GND	NC	HB_TDMTD1	GND	External TDM serial port
14	GND	NC	HB_TDMRFS1	GND	XHB_TDMTFS1	NC	GND	
13	GND	NC	NC	GND	NC	NC	GND	
12	GND	EB_L4ACK	EB_L4CLK	GND	NC	NC	GND	External link port 4
11	GND	EB_L4DAT4	EB_L4DAT5	GND	EB_L4DAT6	EB_L4DAT7	GND	
10	GND	EB_L4DAT0	EB_L4DAT1	GND	EB_L4DAT2	EB_L4DAT3	GND	
9	GND	EB_L3ACK	EB_L3CLK	GND	NC	NC	GND	External link port 3
8	GND	EB_L3DAT4	EB_L3DAT5	GND	EB_L3DAT6	EB_L3DAT7	GND	
7	GND	EB_L3DAT0	EB_L3DAT1	GND	EB_L3DAT2	EB_L3DAT3	GND	
6	GND	EB_L2ACK	EB_L2CLK	GND	NC	NC	GND	External link port 2
5	GND	EB_L2DAT4	EB_L2DAT5	GND	EB_L2DAT6	EB_L2DAT7	GND	
4	GND	EB_L2DAT0	EB_L2DAT1	GND	EB_L2DAT2	EB_L2DAT3	GND	
3	GND	EB_L1ACK	EB_L1CLK	GND	NC	NC	GND	External link port 1
2	GND	EB_L1DAT4	EB_L1DAT5	GND	EB_L1DAT6	EB_L1DAT7	GND	
1	GND	EB_L1DAT0	EB_L1DAT1	GND	EB_L1DAT2	EB_L1DAT3	GND	
Row	Z	A	B	C	D	E	F	

[*] Power, flags, interrupts, reset, RS-232

3.2.3 JTAG header

The JTAG header (J1) allows in-circuit emulation with an optional ICE emulator (available from Analog Devices). All eight ADSP-21160 DSPs are connected to the JTAG connector. Figure 3–4 shows where pin 1 is located on the JTAG header, and Table 3–6 gives the connector pinout. Appendix A explains how to connect an emulator to the Hammerhead-6U-cPCI.

Figure 3–4 *Location of the JTAG Header Pins*

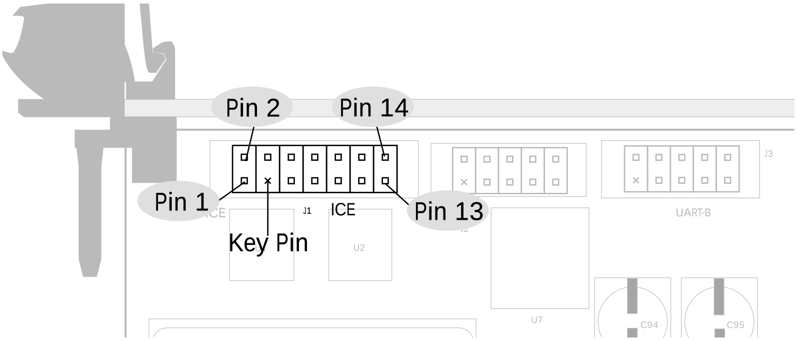


Table 3–6 *JTAG Header Pinout (J1)*

Pin	Signal	Pin	Signal
1	GND	2	$\overline{\text{EMU}}$
3	KEY	4	CLK
5	BTMS	6	TMS
7	BTCK	8	TCK
9	$\overline{\text{BTRST}}$	10	$\overline{\text{TRST}}$
11	BTDI	12	TDI
13	GND	14	TDO

3.2.4 RS-232 Connectors

The Hammerhead-6U-cPCI is configured with two 10-pin RS-232 connectors (J2, J3). J2 is the RS-232 connector for DSP cluster A, and J3 is the RS-232 connector for DSP cluster B. The connectors transport serial data between the host and the dual UARTs, which interface the data between the RS-232 connectors and the ADSP-21160 processors. The SharcFIN ASIC provides a memory-mapped register set that allows each DSP cluster to interface to a PC via its RS-232 port. Figure 3–5 shows where the connector pins are located, and Table 3–7 gives the connector pinout.

Figure 3–5 *Location of the RS-232 Connector Pins*

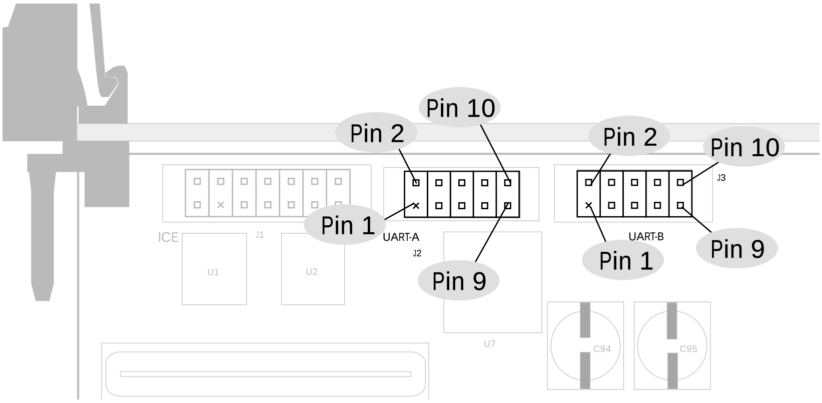


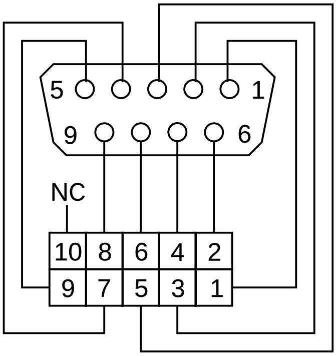
Table 3–7 *RS-232 Connector Pinout (J2, J3)*

Pin*	Signal	Pin	Signal
1		2	
3	TXD	4	$\overline{\text{CTS}}$
5	RXD	6	$\overline{\text{RTS}}$
7		8	NC
9	GND	10	NC

* Pins 1, 2, and 7 (Carrier Detect, Data Set Ready, and Data Terminal Ready) are jumpered.

Figure 3–6 shows the connections between the DB9 cable and the RS-232 connectors (J2, J3). For instructions on connecting a DB9 cable to the RS-232 connectors, refer to the section entitled “Connecting the Hammerhead-6U-cPCI to an RS-232 Source” on page 24.

Figure 3–6 *DB9-to-RS-232 Cable Connections*



DB9	J2, J3	
1 Carrier Detect	1	
2 Transmit Data	3	Transmit Data
3 Receive Data	5	Receive Data
4 Data Terminal Ready	7	
5 Ground	9	Ground
6 Data Set Ready	2	
7 NC	4	Clear to Send
8 Clear to Send	6	Request to Send
9 NC	8	NC
	10	NC

3.2.5 BittWare Mezzanine Interface

The BittWare mezzanine interface (J4, J6, J8, J9) allows you to attach additional mezzanines containing shared memory to the Hammerhead-6U-cPCI. For further information on this interface, contact BittWare.

3.2.6 Flag I/O Connectors

The 16-pin flag I/O connectors (J5, J7) allow access to the FLAG2 and FLAG3 signals on each ADSP-21160 DSP. This access to the ADSP-21160s' flags allows you to input and output signals directly to the processors. Because the connector is directly routed to the ADSP-21160s, the circuit is diode protected to GND and 3.3V to shield the processors from voltage overload. Connector J7 is the flag I/O for cluster A, and J5 is the flag I/O for cluster B. Figure 3–7 shows where the pins are located on the connectors, and Table 3–8 gives the connector pinout.

Figure 3–7 *Location of the Flag I/O Connector Pins*

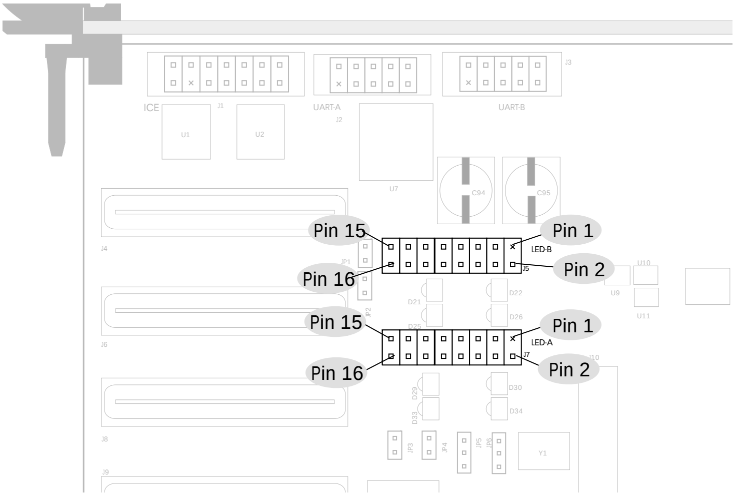


Table 3–8 *Flag I/O Connector Pinouts (J5, J7)*

J7

Pin	Signal	Description	Pin	Signal
1	HA1_F2	21160-A1 FLAG2	2	GND
3	HA1_F3 HA2_I2	21160-A1 FLAG3 21160-A2 IRQ2	4	GND
5	HA2_F2	21160-A2 FLAG2	6	GND
7	HA2_F3 HA1_I2	21160-A2 FLAG3 21160-A1 IRQ2	8	GND
9	HA3_F2	21160-A3 FLAG2	10	GND
11	HA3_F3 HA4_I2	21160-A3 FLAG3 21160-A4 IRQ2	12	GND
13	HA4_F2	21160-A4 FLAG2	14	GND
15	HA4_F3 HA3_I2	21160-A4 FLAG3 21160-A3 IRQ2	16	GND

J5

Pin	Signal	Description	Pin	Signal
1	HB1_F2	21160-B1 FLAG2	2	GND
3	HB1_F3 HB2_I2	21160-B1 FLAG3 21160-B2 IRQ2	4	GND
5	HB2_F2	21160-B2 FLAG2	6	GND
7	HB2_F3 HB1_I2	21160-B2 FLAG3 21160-B1 IRQ2	8	GND
9	HB3_F2	21160-B3 FLAG2	10	GND
11	HB3_F3 HB4_I2	21160-B3 FLAG3 21160-B4 IRQ2	12	GND
13	HB4_F2	21160-B4 FLAG2	14	GND
15	HB4_F3 HB3_I2	21160-B4 FLAG3 21160-B3 IRQ2	16	GND

3.2.7 SODIMM Connectors

For each DSP cluster, the Hammerhead-6U-cPCI has an industry-standard 144-pin connection for a standard SODIMM module. J10 is the SODIMM connector for cluster A, and J11 is the SODIMM connector for cluster B. The SODIMMs are available in 64, 128, 256, and 512 MB SDRAM modules.

3.2.8 Optional External Reset Connectors

The Hammerhead-6U-cPCI has two optional 6-pin external reset connectors (J12, J14) to allow one Hammerhead-6U-cPCI board to reset other Hammerhead-6U-cPCI boards in the same system. J14 is the external reset connector for DSP cluster A, and J12 is the connector for cluster B. These connectors are not populated, except by special order.

Each connector supports an output reset line, which allows the board to reset other boards. Each also supports an input reset line, which allows it to accept a reset signal from another board. If the input signal is driven low, the board will perform a hardware reset on all four SHARCs in the cluster. The input signal is pulled up with a 10K resistor.

If the output signal is driven low, the board will output a reset signal to other boards. When the output is connected to group reset, it can drive a reset signal to up to 250 boards. If the output signal is tied to the board's hardware reset line, it is driven low by either a host board reset or by a watchdog reset.

The section entitled "Connecting an External Reset Signal to the Reset Connectors" on page 22 explains how to use the external reset signals, and section 2.5 explains the board's reset events. Figure 3–8 shows where the pins are located on the connectors, and Table 3–9 gives the connector pinout.

Figure 3–8 *Location of the External Reset Connector Pins*

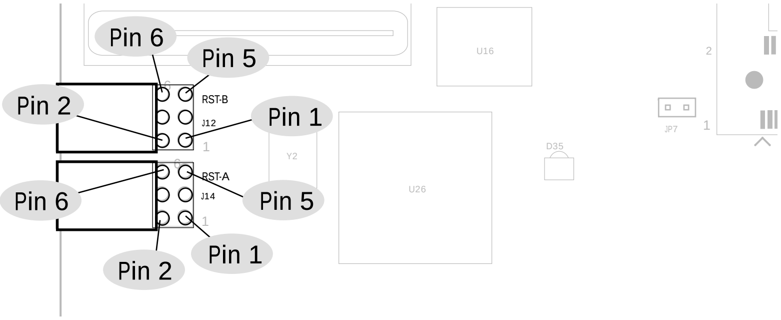


Table 3–9 *External Reset Connector Pinout (J12, J14)*

J14				**J12**			
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
1	NC	2	HA_EXTGRPRSTIN	1	NC	2	HB_EXTGRPRSTIN
3	NC	4	GND	3	NC	4	GND
5	GND	6	HA_EXTGRPRSTOUT	5	GND	6	HB_EXTGRPRSTOUT

3.2.9 External Power Connector

The Hammerhead-6U-cPCI has an external power connector (J13) to provide power to the board when it is operating in standalone mode. The external power connector is an 8-pin connector that supplies +3.3V, +5V, +12V, and –12V to the Hammerhead-6U-cPCI. Figure 3–9 shows the location of the pins on the external power connector (J13), andTable 3–10 gives the connector pinout. Section 2.2.3 explains how to connect an external power supply to the connector. This connector is not populated, except by special order, because it interferes with CompactPCI chassis slot guides.

Figure 3–9 *Location of the External Power Connector Pins*

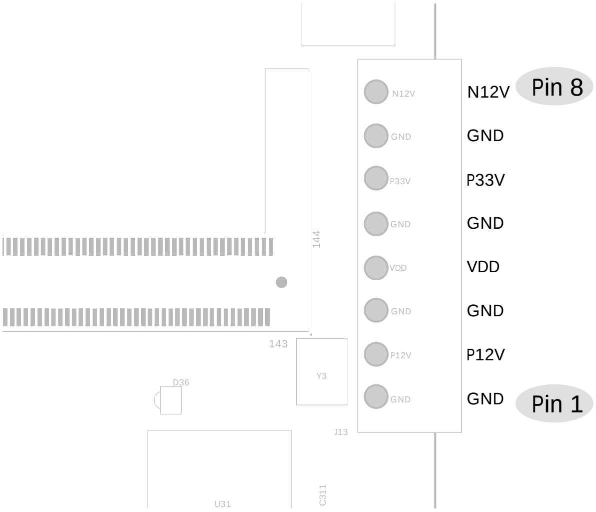


Table 3–10 *External Power Connector Pinout (J13)*

Pin	Signal
1	GND
2	+12V
3	GND
4	VDD
5	GND
6	+3.3V
7	GND
8	–12V

3.2.10 PMC+ Sites

The Hammerhead-6U-cPCI features a PMC+ site for each DSP cluster. You can attach either standard PMC modules or BittWare PMC+ I/O modules to the PMC+ sites.

Each PMC+ site consists of four 64-pin connectors. Three connectors are standard 64-pin PMC connectors (J17, J20, J22 for cluster A; J18, J19, J21 for cluster B) that provide the 64-bit, 66 MHz PCI interface. The fourth connector (J16 for cluster A, J23 for cluster B) is a 64-pin PMC+ connector that connects BittWare's PMC+ I/O modules directly to the ADSP-21160 processors via four link ports, a serial TDM bus, two PMC-to-host interrupts, two host-to-PMC interrupts, and a reset line. Table 3–11 gives the connector pinout for PMC+ site A, and Table 3–12 gives the connector pinout for PMC+ site B.

Warning!

BittWare uses Jn4 of the PMC connectors (see Table 3–11 and Table 3–12) for our PMC+ extensions. If you are mounting a PMC card that uses the Jn4 connector and is not from BittWare, the PMC card may have incompatibilities with the PMC+ (Jn4) connector on the Hammerhead-6U-cPCI. Call BittWare technical support for assistance.

Table 3–11 *PMC+ Site A Connector Pinout*

J17 (PMC-A Jn1)				J20 (PMC-A Jn2)			
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
1	TCK	2	N12V	1	P12V	2	$\overline{\text{TRST}}$
3	GND	4	$\overline{\text{INTA}}$	3	TMS	4	TDO
5	$\overline{\text{INTB}}$	6	$\overline{\text{INTC}}$	5	TDI	6	GND
7	$\overline{\text{BUSMODE1}}$	8	VDD	7	GND	8	$\overline{\text{PCI-RSVD}}$
9	$\overline{\text{INTD}}$	10	$\overline{\text{PCI-RSVD}}$	9	$\overline{\text{PCI-RSVD}}$	10	$\overline{\text{PCI-RSVD}}$
11	GND	12	$\overline{\text{PCI-RSVD}}$	11	$\overline{\text{BUSMODE2}}$	12	P33V
13	CLK	14	GND	13	$\overline{\text{RST}}$	14	$\overline{\text{BUSMODE3}}$
15	GND	16	$\overline{\text{GNT}}$	15	P33V	16	$\overline{\text{BUSMODE4}}$
17	$\overline{\text{REQ}}$	18	VDD	17	$\overline{\text{PCI-RSVD}}$	18	GND
19	V(I/O)	20	AD[31]	19	AD[30]	20	AD[29]
21	AD[28]	22	AD[27]	21	GND	22	AD[26]
23	AD[25]	24	GND	23	AD[24]	24	P33V
25	GND	26	$\overline{\text{C/BE[03]}}$	25	IDSEL	26	AD[23]
27	AD[22]	28	AD[21]	27	P33V	28	AD[20]
29	AD[19]	30	VDD	29	AD[18]	30	GND
31	V(I/O)	32	AD[17]	31	AD[16]	32	$\overline{\text{C/BE[2]}}$
33	$\overline{\text{FRAME}}$	34	GND	33	GND	34	PMC-RSVD
35	GND	36	$\overline{\text{IRDY}}$	35	$\overline{\text{TRDY}}$	36	P33V
37	$\overline{\text{DEVSEL}}$	38	VDD	37	GND	38	$\overline{\text{STOP}}$
39	GND	40	$\overline{\text{LOCK}}$	39	$\overline{\text{PERR}}$	40	GND
41	$\overline{\text{SDONE}}$	42	$\overline{\text{SBO}}$	41	P33V	42	$\overline{\text{SERR}}$
43	PAR	44	GND	43	$\overline{\text{C/BE[1]}}$	44	GND
45	V(I/O)	46	AD[15]	45	AD[14]	46	AD[13]
47	AD[12]	48	AD[11]	47	GND	48	AD[10]
49	AD[09]	50	VDD	49	AD[08]	50	P33V
51	GND	52	$\overline{\text{C/BE[0]}}$	51	AD[07]	52	PMC-RSVD
53	AD[06]	54	AD[05]	53	P33V	54	PMC-RSVD
55	AD[04]	56	GND	55	PMC-RSVD	56	GND
57	V(I/O)	58	AD[03]	57	PMC-RSVD	58	PMC-RSVD
59	AD[02]	60	AD[01]	59	GND	60	PMC-RSVD
61	AD[00]	62	VDD	61	$\overline{\text{ACK64}}$	62	P33V
63	GND	64	$\overline{\text{REQ64}}$	63	GND	64	PMC-RSVD

J22 (PMC-A Jn3)

Pin	Signal	Pin	Signal
1	PCI-RSVD	2	GND
3	GND	4	$\overline{\text{C/BE[7]}}$
5	$\overline{\text{C/BE[6]}}$	6	$\overline{\text{C/BE[5]}}$
7	$\overline{\text{C/BE[4]}}$	8	GND
9	V(I/O)	10	PAR64
11	AD[63]	12	AD[62]
13	AD[61]	14	GND
15	GND	16	AD[60]
17	AD[59]	18	AD[58]
19	AD[57]	20	GND
21	V(I/O)	22	AD[56]
23	AD[55]	24	AD[54]
25	AD[53]	26	GND
27	GND	28	AD[52]
29	AD[51]	30	AD[50]
31	AD[49]	32	GND
33	GND	34	AD[48]
35	AD[47]	36	AD[46]
37	AD[45]	38	GND
39	V(I/O)	40	AD[44]
41	AD[43]	42	AD[42]
43	AD[41]	44	GND
45	GND	46	AD[40]
47	AD[39]	48	AD[38]
49	AD[37]	50	GND
51	GND	52	AD[36]
53	AD[35]	54	AD[34]
55	AD[33]	56	GND
57	V(I/O)	58	AD[32]
59	PCI-RSVD	60	PCI-RSVD
61	PCI-RSVD	62	GND
63	GND	64	PCI-RSVD

J16 (PMC-A Jn4)

Pin	Signal	Pin	Signal
1	TDMTD	2	TDMRFS
3	TDMRD	4	TDMTRC
5	GND	6	GND
7	L1CLK/L1TXCLK	8	L1ACK/L1RXCLK
9	GND	10	GND/L1FSYNC
11	L1DAT0	12	L1DAT1
13	L1DAT2	14	L1DAT3
15	L1DAT4	16	L1DAT5
17	L1DAT6	18	L1DAT7
19	GND	20	GND
21	L2CLK/L2TXCLK	22	L2ACK/L2RXCLK
23	GND	24	GND/L2FSYNC
25	L2DAT0	26	L2DAT1
27	L2DAT2	28	L2DAT3
29	L2DAT4	30	L2DAT5
31	L2DAT6	32	L2DAT7
33	GND	34	GND
35	L3CLK/L3TXCLK	36	L3ACK/L3RXCLK
37	GND	38	GND/L3FSYNC
39	L3DAT0	40	L3DAT1
41	L3DAT2	42	L3DAT3
43	L3DAT4	44	L3DAT5
45	L3DAT6	46	L3DAT7
47	GND	48	GND
49	L4CLK/L4TXCLK	50	L4ACK/L4RXCLK
51	GND	52	GND/L4FSYNC
53	L4DAT0	54	L4DAT1
55	L4DAT2	56	L4DAT3
57	L4DAT4	58	L4DAT5
59	L4DAT6	60	L4DAT7
61	GND	62	$\overline{\text{RST}}$
63	SCL	64	SDA

Table 3–12 *PMC+ Site B Connector Pinout*

J18 (PMC-B Jn1)				J21 (PMC-B Jn2)			
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
1	TCK	2	N12V	1	P12V	2	TRST
3	GND	4	INTA	3	TMS	4	TDO
5	INTB	6	INTC	5	TDI	6	GND
7	BUSMODE1	8	VDD	7	GND	8	PCI-RSVD
9	INTD	10	PCI-RSVD	9	PCI-RSVD	10	PCI-RSVD
11	GND	12	PCI-RSVD	11	BUSMODE2	12	P33V
13	CLK	14	GND	13	RST	14	BUSMODE3
15	GND	16	GNT	15	P33V	16	BUSMODE4
17	REQ	18	VDD	17	PCI-RSVD	18	GND
19	V(I/O)	20	AD[31]	19	AD[30]	20	AD[29]
21	AD[28]	22	AD[27]	21	GND	22	AD[26]
23	AD[25]	24	GND	23	AD[24]	24	P33V
25	GND	26	C/BE[03]	25	IDSEL	26	AD[23]
27	AD[22]	28	AD[21]	27	P33V	28	AD[20]
29	AD[19]	30	VDD	29	AD[18]	30	GND
31	V(I/O)	32	AD[17]	31	AD[16]	32	C/BE[2]
33	FRAME	34	GND	33	GND	34	PMC-RSVD
35	GND	36	IRDY	35	TRDY	36	P33V
37	DEVSEL	38	VDD	37	GND	38	STOP
39	GND	40	LOCK	39	PERR	40	GND
41	SDONE	42	SBO	41	P33V	42	SERR
43	PAR	44	GND	43	C/BE[1]	44	GND
45	V(I/O)	46	AD[15]	45	AD[14]	46	AD[13]
47	AD[12]	48	AD[11]	47	GND	48	AD[10]
49	AD[09]	50	VDD	49	AD[08]	50	P33V
51	GND	52	C/BE[0]	51	AD[07]	52	PMC-RSVD
53	AD[06]	54	AD[05]	53	P33V	54	PMC-RSVD
55	AD[04]	56	GND	55	PMC-RSVD	56	GND
57	V(I/O)	58	AD[03]	57	PMC-RSVD	58	PMC-RSVD
59	AD[02]	60	AD[01]	59	GND	60	PMC-RSVD
61	AD[00]	62	VDD	61	ACK64	62	P33V
63	GND	64	REQ64	63	GND	64	PMC-RSVD

J19 (PMC-B Jn3)

Pin	Signal	Pin	Signal
1	PCI-RSVD	2	GND
3	GND	4	$\overline{\text{C/BE[7]}}$
5	$\overline{\text{C/BE[6]}}$	6	$\overline{\text{C/BE[5]}}$
7	$\overline{\text{C/BE[4]}}$	8	GND
9	V(I/O)	10	PAR64
11	AD[63]	12	AD[62]
13	AD[61]	14	GND
15	GND	16	AD[60]
17	AD[59]	18	AD[58]
19	AD[57]	20	GND
21	V(I/O)	22	AD[56]
23	AD[55]	24	AD[54]
25	AD[53]	26	GND
27	GND	28	AD[52]
29	AD[51]	30	AD[50]
31	AD[49]	32	GND
33	GND	34	AD[48]
35	AD[47]	36	AD[46]
37	AD[45]	38	GND
39	V(I/O)	40	AD[44]
41	AD[43]	42	AD[42]
43	AD[41]	44	GND
45	GND	46	AD[40]
47	AD[39]	48	AD[38]
49	AD[37]	50	GND
51	GND	52	AD[36]
53	AD[35]	54	AD[34]
55	AD[33]	56	GND
57	V(I/O)	58	AD[32]
59	PCI-RSVD	60	PCI-RSVD
61	PCI-RSVD	62	GND
63	GND	64	PCI-RSVD

J23 (PMC-B Jn4)

Pin	Signal	Pin	Signal
1	TDMTD	2	TDMRFS
3	TDMRD	4	TDMTRC
5	GND	6	GND
7	L1CLK/L1TXCLK	8	L1ACK/L1RXCLK
9	GND	10	GND/L1FSYNC
11	L1DAT0	12	L1DAT1
13	L1DAT2	14	L1DAT3
15	L1DAT4	16	L1DAT5
17	L1DAT6	18	L1DAT7
19	GND	20	GND
21	L2CLK/L2TXCLK	22	L2ACK/L2RXCLK
23	GND	24	GND/L2FSYNC
25	L2DAT0	26	L2DAT1
27	L2DAT2	28	L2DAT3
29	L2DAT4	30	L2DAT5
31	L2DAT6	32	L2DAT7
33	GND	34	GND
35	L3CLK/L3TXCLK	36	L3ACK/L3RXCLK
37	GND	38	GND/L3FSYNC
39	L3DAT0	40	L3DAT1
41	L3DAT2	42	L3DAT3
43	L3DAT4	44	L3DAT5
45	L3DAT6	46	L3DAT7
47	GND	48	GND
49	L4CLK/L4TXCLK	50	L4ACK/L4RXCLK
51	GND	52	GND/L4FSYNC
53	L4DAT0	54	L4DAT1
55	L4DAT2	56	L4DAT3
57	L4DAT4	58	L4DAT5
59	L4DAT6	60	L4DAT7
61	GND	62	$\overline{\text{RST}}$
63	SCL	64	SDA

3.2.11 External Interrupt Inputs

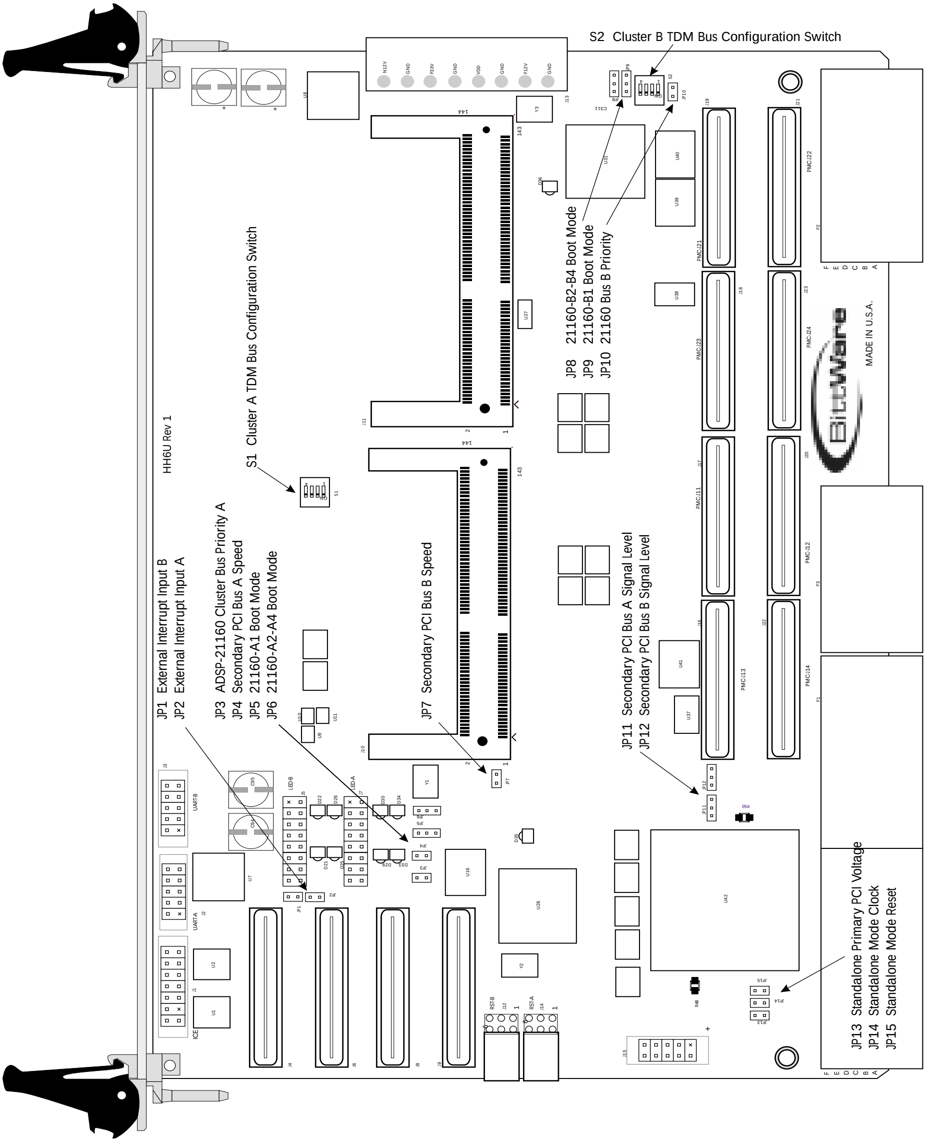
JP1 and JP2 provide a direct path for external interrupts to the SharcFIN ASICs via the SharcFIN's PR_INT signal. JP1 is the connector for SharcFIN B, and JP2 is the connector for SharcFIN A. The PR_INT signal allows an interrupt to the SharcFIN from the peripheral bus. However, the jumpers and the PR_INT lines are wired together; therefore, only one of them should be used. JP2 and the PR_INT line from SharcFIN A are wired together, and JP1 and the PR_INT line from SharcFIN B are wired together.

To input an external interrupt, attach a ground wire to pin 2 of JP1 or JP2 and a signal wire to pin 1. Since the connector is directly routed to the SharcFIN, the circuit is diode protected to GND and 3.3V to shield the ASIC from voltage overload.

3.3 Layout and Function of the Configuration Jumpers and Switches

This section shows where each of the Hammerhead-6U-cPCI's configuration jumpers and switches is located (see Figure 3–10 below) and gives a short description of each (see Table 3–13 and Table 3–14).

Figure 3–10 *Layout of the Configuration Jumpers and Switches*


S2 Cluster B TDM Bus Configuration Switch
JP8 21160-B2-B4 Boot Mode
JP9 21160-B1 Boot Mode
JP10 21160 Bus B Priority
S1 Cluster A TDM Bus Configuration Switch
HH6U Rev 1
JP1 External Interrupt Input B
JP2 External Interrupt Input A
JP3 ADSP-21160 Cluster Bus Priority A
JP4 Secondary PCI Bus A Speed
JP5 21160-A1 Boot Mode
JP6 21160-A2-A4 Boot Mode
JP7 Secondary PCI Bus B Speed
JP11 Secondary PCI Bus A Signal Level
JP12 Secondary PCI Bus B Signal Level
JP13 Standalone Primary PCI Voltage
JP14 Standalone Mode Clock
JP15 Standalone Mode Reset
N12V
GND
P33V
GND
VDD
GND
P12V
GND
UART B
UART A
ICE
LED B
LED A
RST B
RST A
PMC J21
PMC J22
PMC J23
PMC J24
PMC J11
PMC J12
PMC J13
PMC J14
MADE IN U.S.A.

The Hammerhead-6U-cPCI has fifteen configuration jumpers, which allow you to configure and control certain features of the board. Table 3–13 gives an overview of the jumpers, and section 2.2.1 describes their settings in detail.

Table 3–13 *Overview of the Configuration Jumpers*

Jumper	Name	Description
JP1, JP2	External Interrupt Inputs	Connectors for external interrupt to SharcFIN
JP3, JP10	21160 Bus Priority	JP3 selects “fixed” or “rotating” bus priority scheme for ADSP-21160 cluster A DSPs JP10 selects “fixed” or “rotating” bus priority scheme for ADSP-21160 cluster B DSPs
JP4, JP7	Secondary PCI Bus Speed	JP4 sets the speed of secondary PCI bus A to either 33 MHz or 66 MHz JP7 sets the speed of secondary PCI bus B to either 33 MHz or 66 MHz
JP5	21160-A1 Boot Mode	Sets 21160-A1 to boot from host computer, on-board Flash, or remote processor via link port
JP6	21160-A2–A4 Boot Mode	Sets 21160-A2–A4 to boot from host computer, on-board Flash, or remote processor via link port
JP8	21160-B2–B4 Boot Mode	Sets 21160-B2–B4 to boot from host computer, on-board Flash, or remote processor via link port
JP9	21160-B1 Boot Mode	Sets 21160-B1 to boot from host computer, on-board Flash, or remote processor via link port
JP11, JP12	Secondary PCI Bus Signal Level	JP11 sets the signal level of secondary PCI bus A to 3.3 or 5 Volts JP12 sets the signal level of secondary PCI bus B to 3.3 or 5 Volts
JP13	Standalone Primary PCI Voltage	Jumper must be on to operate board in standalone mode
JP14	Standalone Mode Clock	Jumper must be on to operate board in standalone mode
JP15	Standalone Mode Reset	Jumper must be on to operate board in standalone mode

In addition to its configuration jumpers, the Hammerhead-6U-cPCI also has two dip switches, which allow you to configure the connections of the external TDM serial ports. Table 3–14 below gives an overview of the switches; section 2.2.2 explains how to use them to configure the TDM serial connections.

Table 3–14 *Overview of the Serial Port Configuration Switches*

Switch	Name	Description
S1	TDM SPORT A	Configures the external TDM serial connections for cluster A TDM SPORT0 and TDM SPORT1
S2	TDM SPORT B	Configures the external TDM serial connections for cluster B TDM SPORT0 and TDM SPORT1

Chapter 4
Hammerhead-6U-cPCI Board Architecture

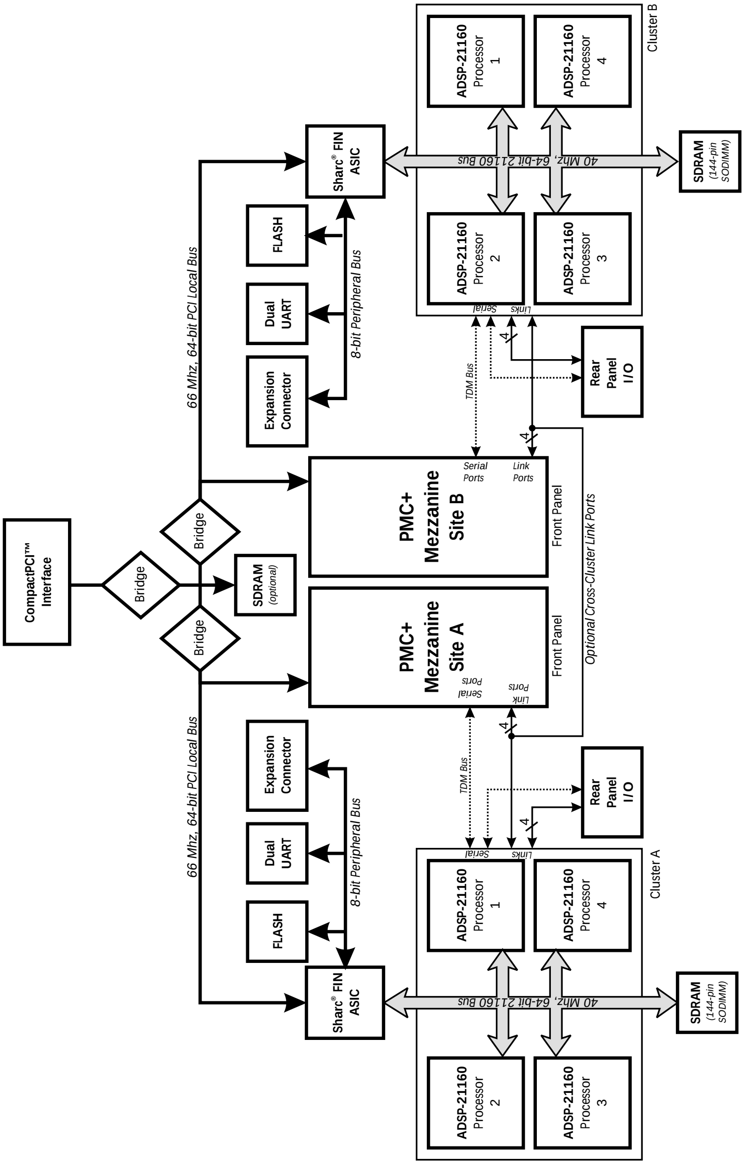
This chapter discusses the architecture of the board, describing how the ADSP-21160 DSPs communicate with other DSPs, with the host, and with other I/O peripherals on and off the board. This chapter covers the following topics:

- The DSPs' memory structure
- The connections of the DSPs' serial ports
- The connections of the DSPs' link ports
- The connections of the DSPs' flags and interrupts
- The connections to the DSPs' 64-bit cluster buses
- The structure of the PCI interface (including the SharcFIN ASIC, primary and secondary PCI buses, and peripheral buses)
- The connections available via the PMC+ interfaces

4.1 Overview of the Board Architecture

This section briefly describes how data flows through the Hammerhead-6U-cPCI board. The sections that follow discuss the board's architecture in more detail. Figure 4–1 is a block diagram of the system.

Figure 4–1 *Architecture of the Hammerhead-6U-cPCI System*



The Hammerhead-6U-cPCI features eight ADSP-21160 DSPs arranged in two clusters of four DSPs. The clusters operate independently of each other, and each cluster features a 64-bit 66 MHz PCI bus, a 64-bit 40 MHz ADSP-21160 cluster bus, a bank of up to 512 MB of SDRAM, a 2 MB bank of Flash memory, a dual UART, a PMC+ interface, a SharcFIN ASIC, and a PCI-to-PCI bridge.

The 64-bit, 40 MHz cluster bus interconnects the four DSPs in the cluster and provides access to the bank of SDRAM. The ADSP-21160 cluster bus connects to that cluster's SharcFIN ASIC, which provides a bridge between the DSPs and the 64-bit, 66 MHz PCI bus. A peripheral bus also extends off of each SharcFIN, providing access to the UART and the Flash memory for each cluster.

The Hammerhead-6U-cPCI features three PCI buses: one bus for each cluster, and one common bus to interface the other two to the CompactPCI interface. The SharcFINs interface the PCI bus for each cluster to PCI-to-PCI bridges, which interface them to a common PCI-to-PCI bridge. The common PCI-to-PCI bridge interfaces the two clusters to the CompactPCI interface.

For I/O options, the board features two rear panel I/O connectors, two RS-232 ports, and two PMC+ interfaces. The two rear panel I/O connectors provide external link and serial ports. The RS-232 ports connect to the UARTs to allow serial communication with the ADSP-21160 DSPs. A PMC+ interface extends off of the PCI bus for each cluster, providing access to standard PMC cards or to BittWare's PMC+ I/O cards. When a BittWare PMC+ card is attached, the PMC+ interfaces provide two link ports, a TDM serial bus, an I^2C interface, interrupts, and a reset line directly to the DSPs.

4.2 ADSP-21160 Architecture

This section gives a short description of the architecture of the ADSP-21160 DSPs. For additional information, refer to the *ADSP-21160 User's Manual* (Analog Devices, Inc.).

4.2.1 Resources Available to the ADSP-21160s

This section discusses the resources available to each processor; resources include memory banks, flags and interrupts, serial ports, and link ports. The following tables summarize how the DSPs' resources are used on the Hammerhead-6U-cPCI. The rows labeled "MS" refer to the DSPs' external memory select lines (MS0–MS3).

Table 4–1 *Resources for 21160-A1*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM (PMC A)	SharcFIN A[*]	SharcFIN A	PMC A L1
1	Flash, UART	TDM (P3)	21160-A3 F1[†] 21160-A4 F1	SharcFIN A	21160-A4 L2
2	SharcFIN		21160-A2 F3 Flag I/O J7 Rear panel P3	LED D1 Flag I/O J7 Rear panel P3	21160-A2 L4
3				21160-A2 I2 LED D2 Flag I/O J7 Rear panel P3	21160-A2 L5
4					RP I/O[‡] L1
5					21160-A4 L3

[*] IRQ0, FLAG0, and FLAG1 on each processor in cluster A connect to the SharcFIN ASIC A interrupt multiplexer. From the SharcFIN, you can route them to different locations on the board.

[†] IRQ1 on each DSP in cluster A is hard-wired within SharcFIN A.

[‡] RP I/O = rear panel I/O connector P3

Table 4–2 *Resources for 21160-A2*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM (PMC A)	SharcFIN A	SharcFIN A	PMC A L2
1	Flash, UART	TDM (P3)	21160-A3 F1 21160-A4 F1	SharcFIN A	RP I/O* L2
2	SharcFIN		21160-A1 F3 Flag I/O J7 Rear panel P3	LED D3 Flag I/O J7 Rear panel P3	21160-A3 L4
3				21160-A1 I2 LED D4 Flag I/O J7 Rear panel P3	21160-A3 L5
4					21160-A1 L2
5					21160-A1 L3

* RP I/O = rear panel I/O connector P3

Table 4–3 *Resources for 21160-A3*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM (PMC A)	SharcFIN A	SharcFIN A	PMC A L3
1	Flash, UART	TDM (P3)	21160-A1 F1 21160-A2 F1	SharcFIN A	RP I/O* L3
2	SharcFIN		21160-A4 F3 Flag I/O J7 Rear panel P3	LED D5 Flag I/O J7 Rear panel P3	21160-A4 L4
3				21160-A4 I2 LED D6 Flag I/O J7 Rear panel P3	21160-A4 L5
4					21160-A2 L2
5					21160-A2 L3

* RP I/O = rear panel I/O connector P3

Table 4–4 *Resources for 21160-A4*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM (PMC A)	SharcFIN A	SharcFIN A	PMC A L4
1	Flash, UART	TDM (P3)	21160-A1 F1 21160-A2 F1	SharcFIN A	RP I/O[*] L4
2	SharcFIN		21160-A3 F3 Flag I/O J7 Rear panel P3	LED D15 Flag I/O J7 Rear panel P3	21160-A1 L1
3				21160-3 I2 LED D16 Flag I/O J7 Rear panel P3	21160-A1 L5
4					21160-A3 L2
5					21160-A3 L3

[*] RP I/O = rear panel I/O connector P3

Table 4–5 *Resources for 21160-B1*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM (PMC B)	SharcFIN B[*]	SharcFIN B	PMC B L1
1	Flash, UART	TDM (P5)	21160-B3 F1[†] 21160-B4 F1	SharcFIN B	21160-B4 L2
2	SharcFIN		21160-B2 F3 Flag I/O J5 Rear panel P5	LED D7 Flag I/O J5 Rear panel P5	21160-B2 L4
3				21160-B2 I2 LED D8 Flag I/O J5 Rear panel P5	21160-B2 L5
4					RP I/O[‡] L1
5					21160-B4 L3

[*] IRQ0, FLAG0, and FLAG1 on each processor in cluster B connect to the SharcFIN ASIC B interrupt multiplexer. From the SharcFIN, you can route them to different locations on the board.

[†] IRQ1 on each DSP in cluster B is hard-wired within SharcFIN B.

[‡] RP I/O = rear panel I/O connector P5

Table 4–6 *Resources for 21160-B2*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM (PMC B)	SharcFIN B	SharcFIN B	PMC B L2
1	Flash, UART	TDM (P5)	21160-B3 F1 21160-B4 F1	SharcFIN B	RP I/O* L2
2	SharcFIN		21160-B1 F3 Flag I/O J5 Rear panel P5	LED D9 Flag I/O J5 Rear panel P5	21160-B3 L4
3				21160-B1 I2 LED D10 Flag I/O J5 Rear panel P5	21160-B3 L5
4					21160-B1 L2
5					21160-B1 L3

* RP I/O = rear panel I/O connector P5

Table 4–7 *Resources for 21160-B3*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM (PMC B)	SharcFIN B	SharcFIN B	PMC B L3
1	Flash, UART	TDM (P5)	21160-B1 F1 21160-B2 F1	SharcFIN B	RP I/O* L3
2	SharcFIN		21160-B4 F3 Flag I/O J5 Rear panel P5	LED D11 Flag I/O J5 Rear panel P5	21160-B4 L4
3				21160-B4 I2 LED D12 Flag I/O J5 Rear panel P5	21160-B4 L5
4					21160-B2 L2
5					21160-B2 L3

* RP I/O = rear panel I/O connector P5

Table 4–8 *Resources for 21160-B4*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM (PMC B)	SharcFIN B	SharcFIN B	PMC B L4
1	Flash, UART	TDM (P5)	21160-B1 F1 21160-B2 F1	SharcFIN B	RP I/O* L4
2	SharcFIN		21160-B3 F3 Flag I/O J5 Rear Panel P5	LED D13 Flag I/O J5 Rear panel P5	21160-B1 L4
3				21160-B3 I2 LED D14 Flag I/O J5 Rear panel P5	21160-B1 L5
4					21160-B3 L2
5					21160-B3 L3

* RP I/O = rear panel I/O connector P5

4.2.2 ADSP-21160 Memory Structure

This section describes the memory structure of the ADSP-21160 DSPs. The processors can access their own internal memory, the internal memory of other processors in the same cluster, and external memory devices. Section 5.1 explains how to access the DSPs' memory.

Internal Memory

Internal memory addresses an ADSP-21160 DSP's on-chip, dual-ported SRAM and its memory-mapped registers. Each ADSP-21160 DSP has 4 Mbits of on-chip SRAM. The *ADSP-21160 SHARC User's Manual* gives details about the on-chip SRAM's limitations and how to configure it.

Multiprocessor Memory

Multiprocessor memory space (MMS) is the on-chip SRAM of other ADSP-21160 DSPs in the same cluster. A cluster is up to six ADSP-21160 DSPs that share a common processor bus, and any DSP that is connected to the processor bus shares the MMS. The Hammerhead-6U-cPCI has two clusters of four DSPs. All four DSPs in the cluster share a common bus (the ADSP-21160 cluster buses), and each can view the on-chip SRAM of the other three DSPs in its cluster.

External Memory

External memory space consists of other devices that share the ADSP-21160's 64-bit cluster bus. The external memory space is divided into four banked sections of memory and an unbanked section of memory. The four banked sections contain the SDRAM, Flash, UART, and SharcFIN ASIC configuration registers. The unbanked memory allows access to the peripheral bus. Section 5.1 discusses the external memory in more detail.

4.2.3 Serial Port Connections

Each ADSP-21160 DSP has two independent, synchronous serial ports, SPORT0 and SPORT1, that provide an I/O interface to a wide variety of peripheral devices. Each serial port has its own set of control registers and data buffers. With a range of clock and frame synchronization options, the SPORTs allow a variety of serial communications protocols and provide a glueless hardware interface to many industry-standard peripherals.

The serial ports can operate at 1/2 the full clock rate of the processor. Since the Hammerhead-6U-cPCI board is populated with 80 MHz ADSP-21160s, the serial ports will operate at 40 MHz. The serial ports support independent transmit and receive functions and can automatically transfer serial port data to and from on-chip memory using DMA block transfers. All serial ports on the Hammerhead-6U-cPCI operate in TDM (time division multiplexed) multichannel mode.

SPORT0 and SPORT1 from each DSP connect to external TDM buses; each cluster supports two TDM buses. TDM bus 0 for each cluster interconnects SPORT0 from each DSP in the cluster with the PMC interface and the SharcFIN ASIC for that cluster. TDM bus 1 for each cluster interconnects SPORT1 from each DSP in the cluster with the rear panel I/O connector and the SharcFIN ASIC for that cluster. Figure 4–2 illustrates the serial port connections for DSP cluster A, and Figure 4–3 illustrates the serial port connections for DSP cluster B.

Figure 4–2 *Block Diagram of Serial Port Connections: Cluster A*

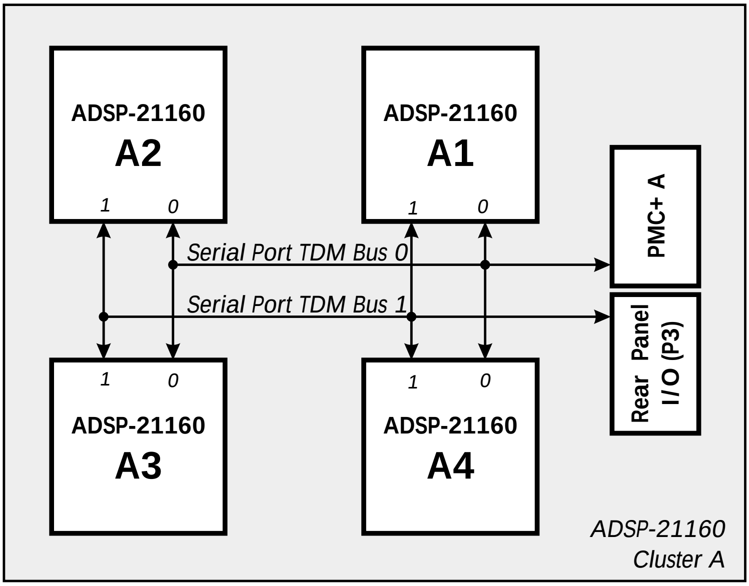
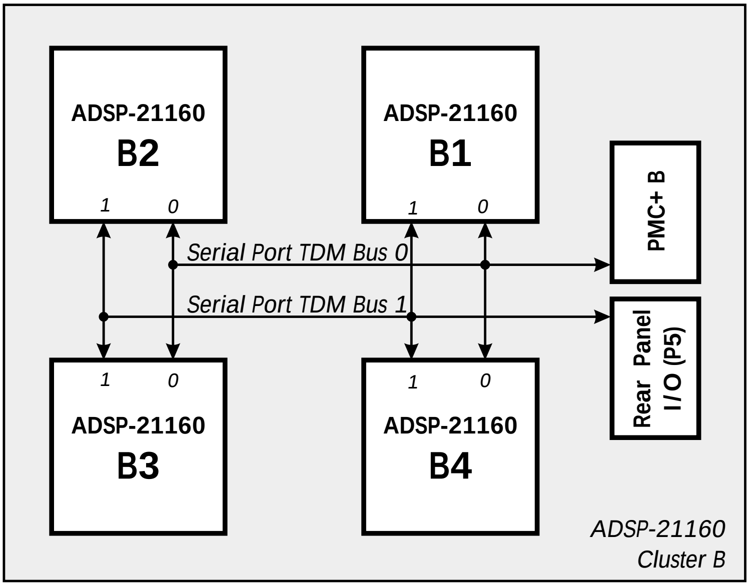


Figure 4–3 *Block Diagram of Serial Port Connections: Cluster B*



4.2.4 Link Port Connections

Each ADSP-21160 DSP has six 8-bit, 80 Mbyte/s link ports, which can connect to link ports from other DSPs or peripherals. The link ports are also available for link booting. They are bidirectional and can operate at frequencies up to the same speed as the DSP's internal clock, letting each port transfer up to 8 bits of data per internal clock cycle. One link per DSP connects to rear panel I/O, one connects to the PMC+ interface, and four are connected to other DSPs for interprocessor communication. Figure 4–4 shows the link port connections for DSP cluster A, and Figure 4–5 shows the link port connections for DSP cluster B.

Figure 4–4 *Block Diagram of Link Port Connections: Cluster A*

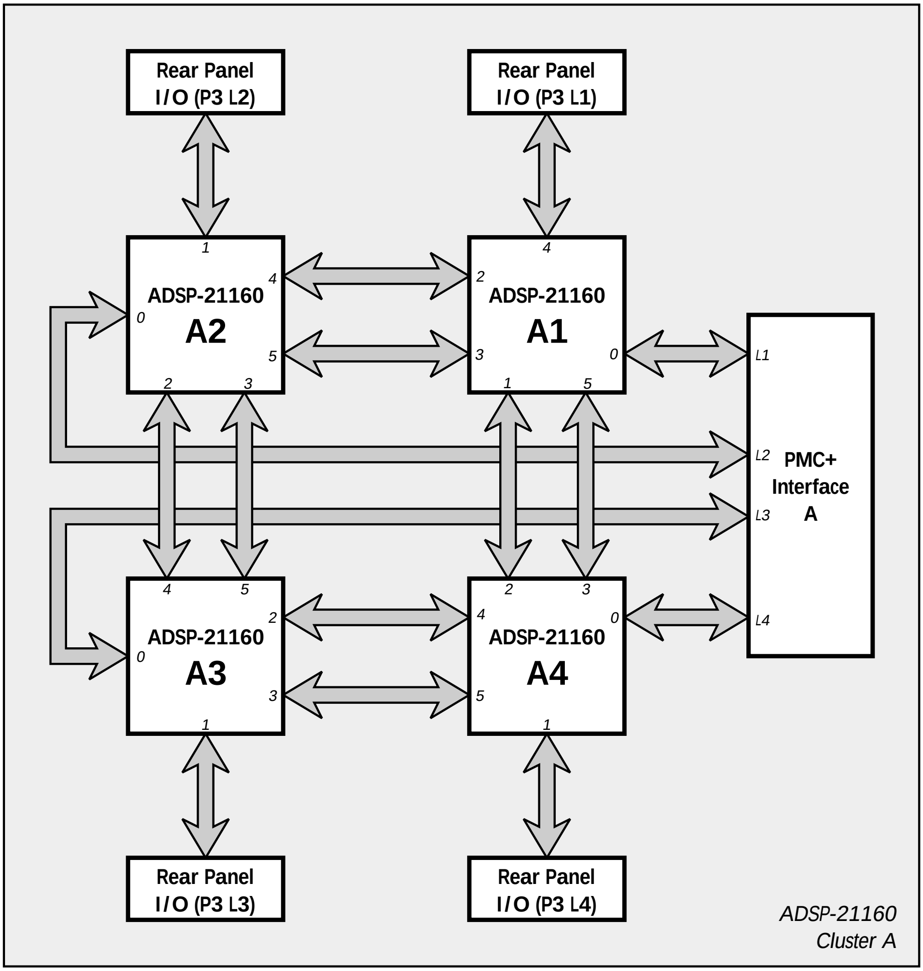
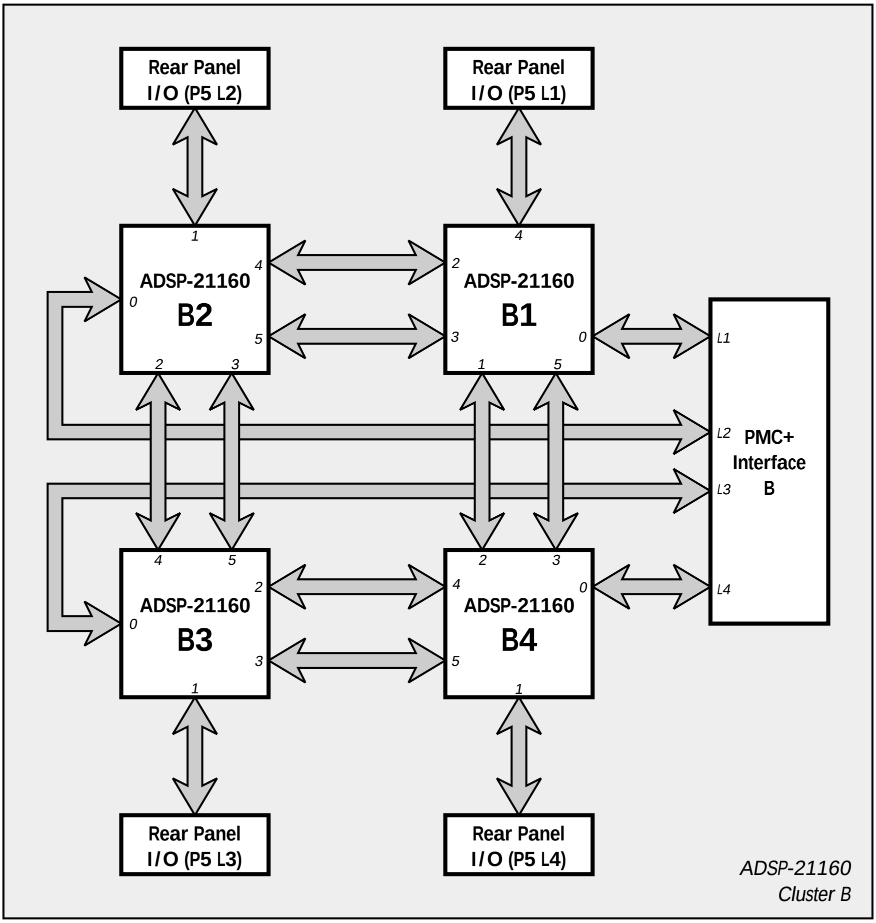


Figure 4–5 *Block Diagram of Link Port Connections: Cluster B*



4.2.5 Flag and Interrupt Connections

Each ADSP-21160 DSP has four flags and three interrupts, which you can use to send and receive control signals to and from other devices in the system. Interrupts can come from devices that require the DSP to perform some task on demand or they can alert the DSP that data is available. The flags allow single-bit signaling between the DSP and other devices. The flags are bidirectional, and each flag can be programmed to be either an input or output. Many DSP instructions can be conditioned on a flags's input value, enabling efficient communication and synchronization between multiple processors or other interfaces.

Two flags and two interrupts from each DSP connect to the SharcFIN ASIC; using registers in the SharcFIN, you can configure the routing of those flags and interrupts. The remaining flags from each DSP connect to the other DSPs in the cluster, to LEDs, to rear panel I/O, and to a buffered inverted flag output connector. The remaining interrupts connect to the other DSPs in the cluster. Figure 4–6 illustrates the flag and interrupt connections for DSP cluster A, and Figure 4–7 illustrates the flag and interrupt connections for DSP cluster B.

Figure 4–6 *Block Diagram of Flag and Interrupt Connections: Cluster A*

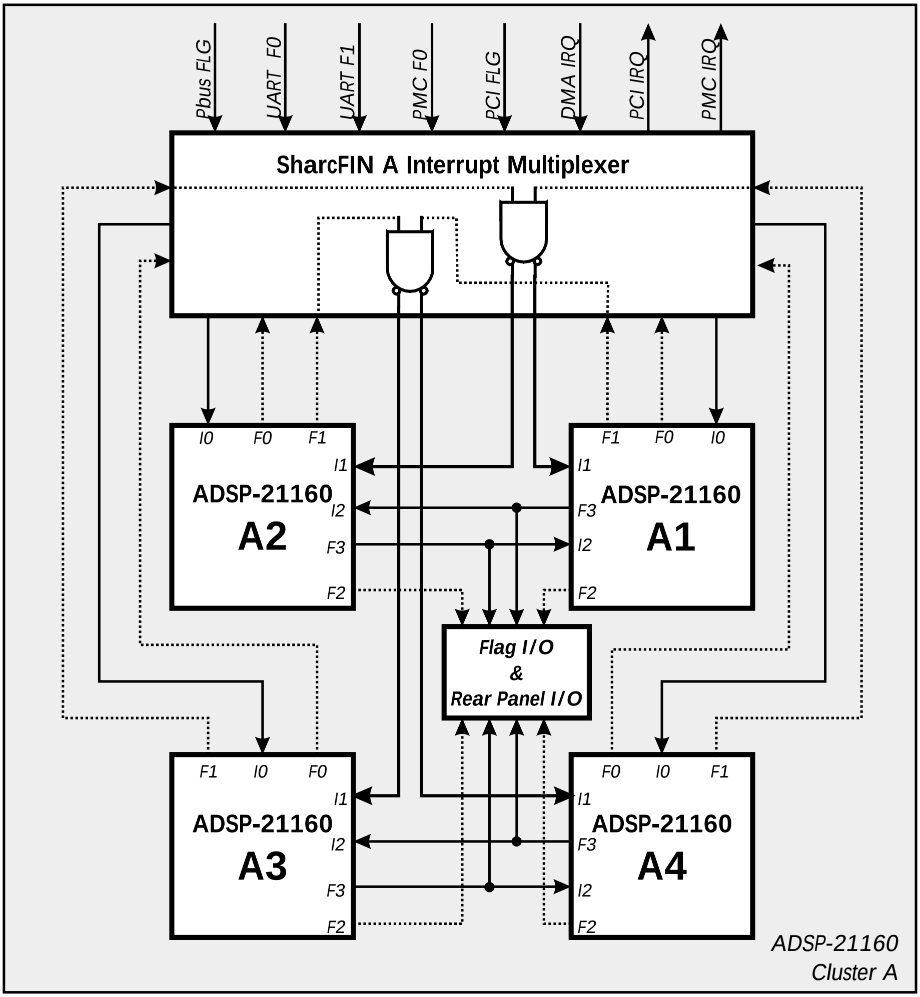


Figure 4–7 *Block Diagram of Flag and Interrupt Connections: Cluster B*

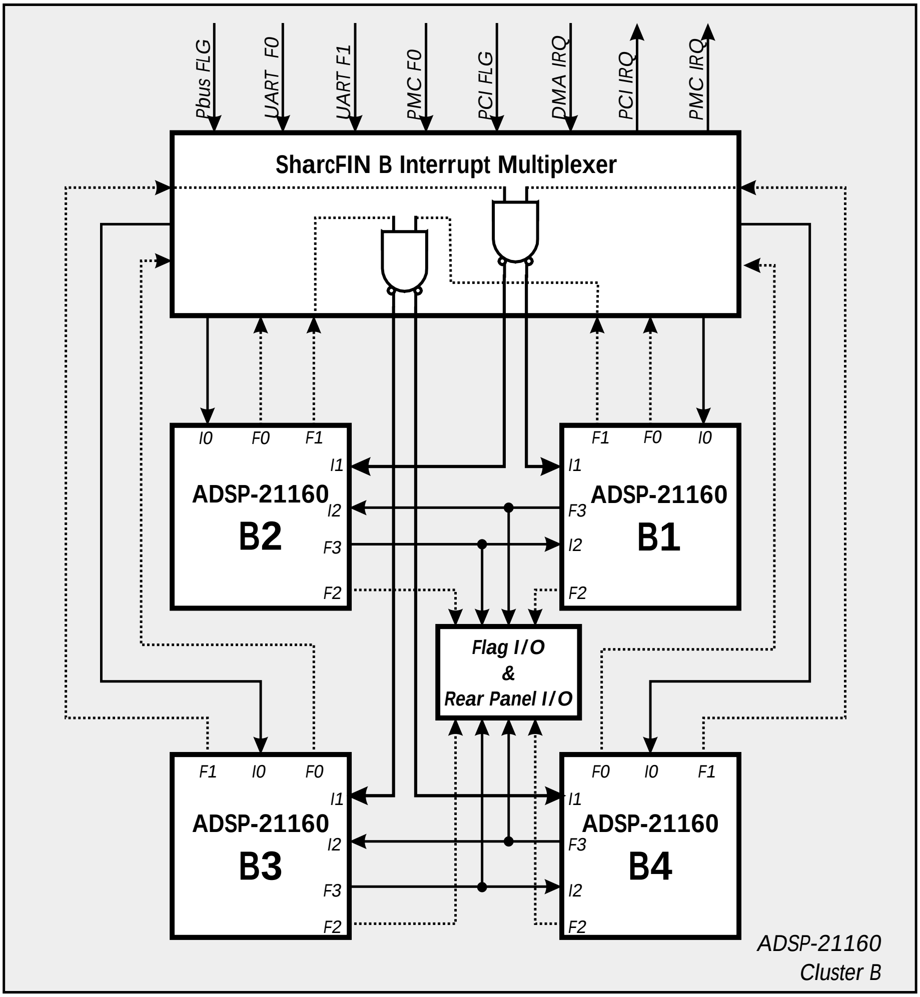


Table 4–9 *Flag and Interrupt Connections for DSP Cluster A*

	21160-A1	**21160-A2**	**21160-A3**	**21160-A4**
Flag0	SharcFIN A	SharcFIN A	SharcFIN A	SharcFIN A
Flag1	SharcFIN A	SharcFIN A	SharcFIN A	SharcFIN A
Flag2	LED D1 Flag I/O J7 Rear panel P3	LED D3 Flag I/O J7 Rear panel P3	LED D5 Flag I/O J7 Rear panel P3	LED D15 Flag I/O J7 Rear panel P3
Flag3	21160-A2 I2 LED D2 Flag I/O J7 Rear panel P3	21160-A1 I2 LED D4 Flag I/O J7 Rear panel P3	21160-A4 I2 LED D6 Flag I/O J7 Rear panel P3	21160-A3 I2 LED D16 Flag I/O J7 Rear panel P3
IRQ0	SharcFIN A	SharcFIN A	SharcFIN A	SharcFIN A
IRQ1	21160-A3 F1 21160-A4 F1	21160-A3 F1 21160-A4 F1	21160-A1 F1 21160-A2 F1	21160-A1 F1 21160-A2 F1
IRQ2	21160-A2 F3 Flag I/O J7 Rear panel P3	21160-A1 F3 Flag I/O J7 Rear panel P3	21160-A4 F3 Flag I/O J7 Rear panel P3	21160-A3 F3 Flag I/O J7 Rear panel P3

Table 4–10 *Flag and Interrupt Connections for DSP Cluster B*

	21160-B1	**21160-B2**	**21160-B3**	**21160-B4**
Flag0	SharcFIN B	SharcFIN B	SharcFIN B	SharcFIN B
Flag1	SharcFIN B	SharcFIN B	SharcFIN B	SharcFIN B
Flag2	LED D7 Flag I/O J5 Rear panel P5	LED D9 Flag I/O J5 Rear panel P5	LED D11 Flag I/O J5 Rear panel P5	LED D13 Flag I/O J5 Rear panel P5
Flag3	21160-B2 I2 LED D8 Flag I/O J5 Rear panel P5	21160-B1 I2 LED D10 Flag I/O J5 Rear panel P5	21160-B4 I2 LED D12 Flag I/O J5 Rear panel P5	21160-B3 I2 LED D14 Flag I/O J5 Rear panel P5
IRQ0	SharcFIN B	SharcFIN B	SharcFIN B	SharcFIN B
IRQ1	21160-B3 F1 21160-B4 F1	21160-B3 F1 21160-B4 F1	21160-B1 F1 21160-B2 F1	21160-B1 F1 21160-B2 F1
IRQ2	21160-B2 F3 Flag I/O J5 Rear panel P5	21160-B1 F3 Flag I/O J5 Rear panel P5	21160-B4 F3 Flag I/O J5 Rear panel P5	21160-B3 F3 Flag I/O J5 Rear panel P5

SharcFIN Flags and Interrupts

Each ADSP-21160 SHARC DSP has two flags and two interrupts that connect to the SharcFIN. IRQ1 from each DSP is hardwired (non-configurable) in the SharcFIN with the following connections:

- 21160-1 IRQ1 is hardwired to 21160-3 FLAG1 and 21160-4 FLAG1.
- 21160-2 IRQ1 is hardwired to 21160-3 FLAG1 and 21160-4 FLAG1.
- 21160-3 IRQ1 is hardwired to 21160-1 FLAG1 and 21160-2 FLAG1.
- 21160-4 IRQ1 is hardwired to 21160-1 FLAG1 and 21160-2 FLAG1.

FLAG0, FLAG1, and IRQ0 from each DSP connect to the SharcFIN ASIC interrupt multiplexer (see section 6.5.11). Using registers in the SharcFIN's configuration space, you can configure the routing of those flags and interrupts. FLAG0 and FLAG1 from each DSP in cluster A are inputs to the SharcFIN A interrupt multiplexer, and FLAG0 and FLAG1 from each DSP in cluster B are inputs to the SharcFIN B interrupt multiplexer. IRQ0 from each DSP in cluster A is an output from the SharcFIN A interrupt multiplexer, and IRQ0 from each DSP in cluster B is an output from the SharcFIN B interrupt multiplexer.

Using the SharcFIN interrupt multiplexer, you can configure the DSPs to receive interrupts from the following sources:

- peripheral bus
- PCI bus
- UART
- other DSPs in the same cluster
- DMA interrupt
- PMC interface

Table 4–9 and Table 4–10 list the flag and interrupt connections for each DSP. Section 6.5.11 explains the interrupt multiplexer in more detail.

Interprocessor Flags and Interrupts

Each DSP can generate and receive interrupts to and from the other three DSPs in its cluster. One interrupt (IRQ2) on each processor is interconnected to a flag (FLAG3) on another processor. IRQ2 and FLAG3 on 21160-1 and 21160-2 are interconnected, and IRQ2 and FLAG3 on 21160-3 and 21160-4 are interconnected. Table 4–9 and Table 4–10 show the flag and interrupt connections for each DSP (see also Figure 4–6 and Figure 4–7).

Flag Connections to LEDs and Test Points

Two flags from each DSP (FLAG2 and FLAG3) connect to LEDs and a flag I/O connector. These flags are useful for testing. DSPs in cluster A connect to flag I/O connector J7 and to LEDs D1–D6 and D15–16; DSPs in cluster B connect to flag I/O connector J5 and to LEDs D7–D14.

One interrupt from each DSP also connects to the flag I/O connector. IRQ2 from each DSP in cluster A connects to flag I/O connector J7; IRQ2 from each DSP in cluster B connects to flag I/O connector J5.

Flag and Interrupt Connections to the Rear Panel

Two flags (FLAG2 and FLAG3) and one interrupt (IRQ2) from each DSP connect to rear panel I/O. Flags and interrupts from cluster A connect to rear panel I/O connector P3, and flags and interrupts from cluster B connect to rear panel I/O connector P5. The flags are outputs only, and the interrupts are bi-directional.

4.2.6 ADSP-21160 Cluster Buses

The Hammerhead-6U-cPCI has two ADSP-21160 cluster buses – one for each cluster. The ADSP-21160 cluster buses are 40 MHz, 64-bit buses that interconnect the four ADSP-21160 processors in the cluster and a bank of up to 512 MB SDRAM. They are connected to the PCI interface through the SharcFIN ASICs. The ADSP-21160 cluster bus is a 64-bit data, 32-bit address bus and uses 3.3 volt signaling. It allows transactions between the ADSP-21160s, the SDRAM, and the PCI-to-DSP bridge.

The ADSP-21160 cluster bus has access to the secondary PCI bus via a single PCI access channel capable of reading or writing single words from the PCI bus. The reads or writes may be memory mapped, I/O mapped, or configuration operations.

The ADSP-21160 cluster bus has access to the peripheral bus via the following:

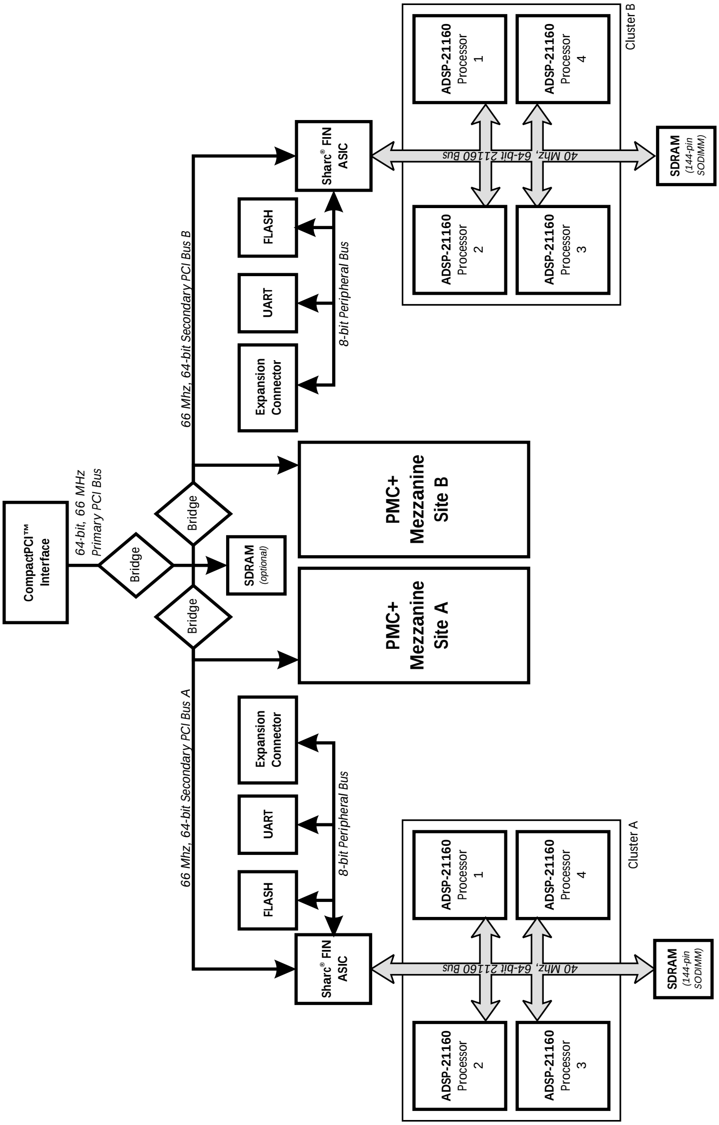
- $\overline{\text{BMS}}$[1]
- MS1

[1] $\overline{\text{BMS}}$ is the ADSP-21160s' Boot Memory Select pin. This pin allows access to a separate external memory space for booting.

4.3 PCI Interface Architecture

The Hammerhead-6U-cPCI PCI interface consists of three 64-bit, 66 MHz PCI buses, which are interconnected by a series of three PCI-to-PCI bridge chips. Two SharcFIN ASICs interface the PCI buses to the ADSP-21160 DSPs. Figure 4–8 below is a block diagram of the PCI interface.

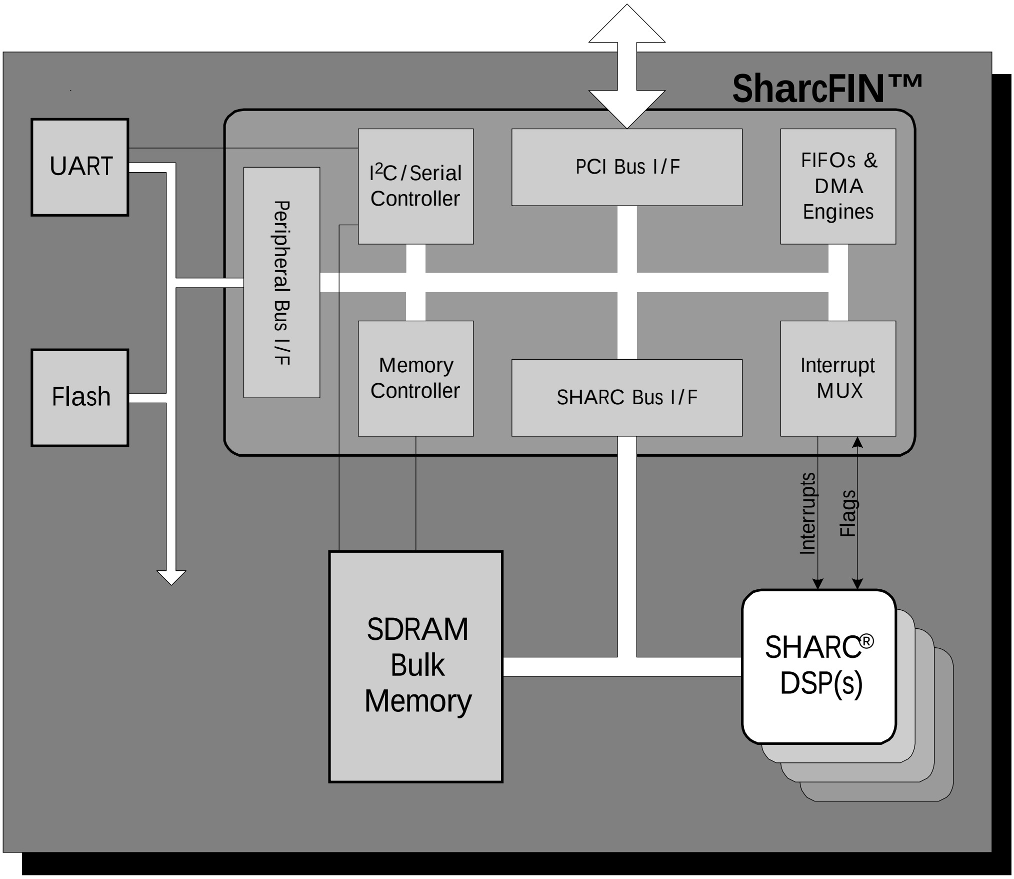
Figure 4–8 *Block Diagram of the PCI Interface Architecture*



4.3.1 Overview of the SharcFIN Architecture

The Hammerhead-6U-cPCI features two BittWare SharcFIN ASICs. The SharcFINs interface the ADSP-21160 DSPs to the PCI bus, SDRAM, and devices on the peripheral bus. They also provide an interrupt multiplexer to allow you to configure the interrupt connections on the board. This section provides an overview of the SharcFIN architecture. Figure 4–9 below is a simplified block diagram of the SharcFIN architecture as it is implemented on the Hammerhead-6U-cPCI board.

Figure 4–9 *Simplified Block Diagram of the SharcFIN Architecture*



Interface to ADSP-21160 Cluster Bus

The first function of the SharcFIN is to interface to the ADSP-21160 cluster bus. The SharcFIN provides a 64-bit interface to the ADSP-21160 cluster bus. It also integrates a full-featured SDRAM controller, which allows the ADSP-21160s to access SDRAM using burst mode access.

Interface to PCI

The second function of the SharcFIN is to interface to PCI. The SharcFIN implements a full 64-bit/66MHz master PCI interface. The PCI interface is PCI rev 2.2 compliant and provides 16 Bytes of configurable PCI mailbox registers.

Interface to Peripheral Bus

A third bus interface is provided by the SharcFIN's peripheral bus. The peripheral bus is a general-purpose utility bus that allows easy interface to standard microprocessor peripherals such as UARTs and Flash memory. It provides a simple, glueless way to add additional functionality to the Hammerhead-6U-cPCI.

I^{2}C Interface

The SharcFIN's I^{2}C/Serial controller integrates some of the most common peripheral requirements right into the SharcFIN. Uses include data communications, SharcFIN interconnection, and hardware configuration and identification.

Interrupt Multiplexer

The SharcFIN integrates an extensive interrupt and flag multiplexer to facilitate system-level control and coordination of multiprocessors. This programmable resource allows each ADSP-21160 to select the sources of its hardware interrupts; sources include other processors, PCI, peripherals, and the internal DMA engines.

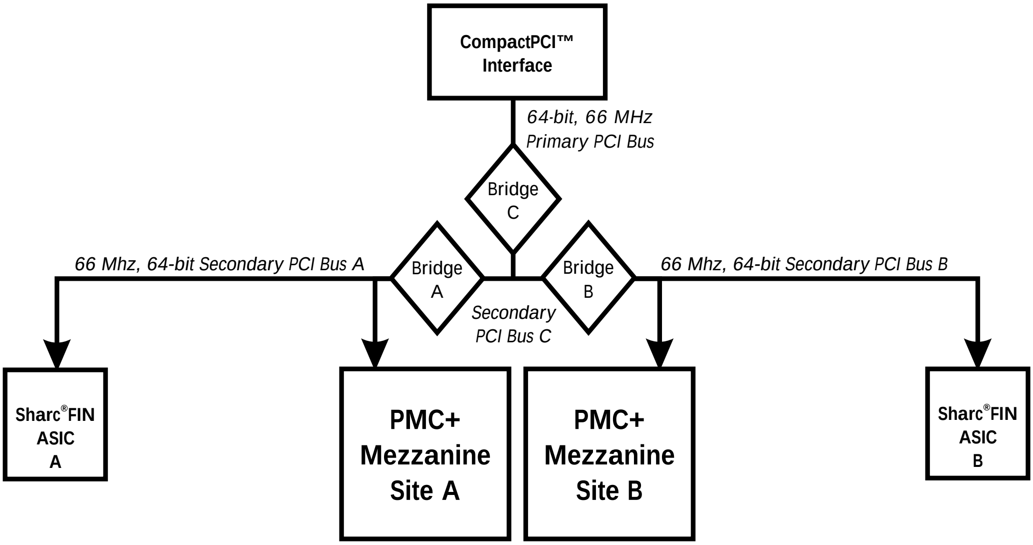
4.3.2 PCI Bus Interface

The Hammerhead-6U-cPCI's 64-bit, 66 MHz PCI bus interface consists of a primary PCI bus, three PCI-to-PCI bridges (A–C), and three secondary PCI buses (A–C). The SharcFIN ASICs provide a bridge between the PCI interface and the ADSP-21160 DSPs.

PCI-to-PCI Interface

The PCI-to-PCI interface consists of the primary PCI bus, secondary PCI bus C, and the three PCI-to-PCI bridge chips (Intel 21154). The first PCI-to-PCI bridge chip (PCI-to-PCI bridge C) interfaces the primary PCI bus to secondary PCI bus C. Secondary PCI bus C branches off to two additional PCI-to-PCI bridge chips – one for each DSP cluster. Secondary PCI bus A extends off PCI-to-PCI bridge A and secondary PCI bus B extends off PCI-to-PCI bridge B (see “PCI-to-DSP Interface” on page 87). Figure 4–10 illustrates the connections of the PCI-to-PCI interface.

Figure 4–10 *Block Diagram of the PCI-to-PCI Interface*

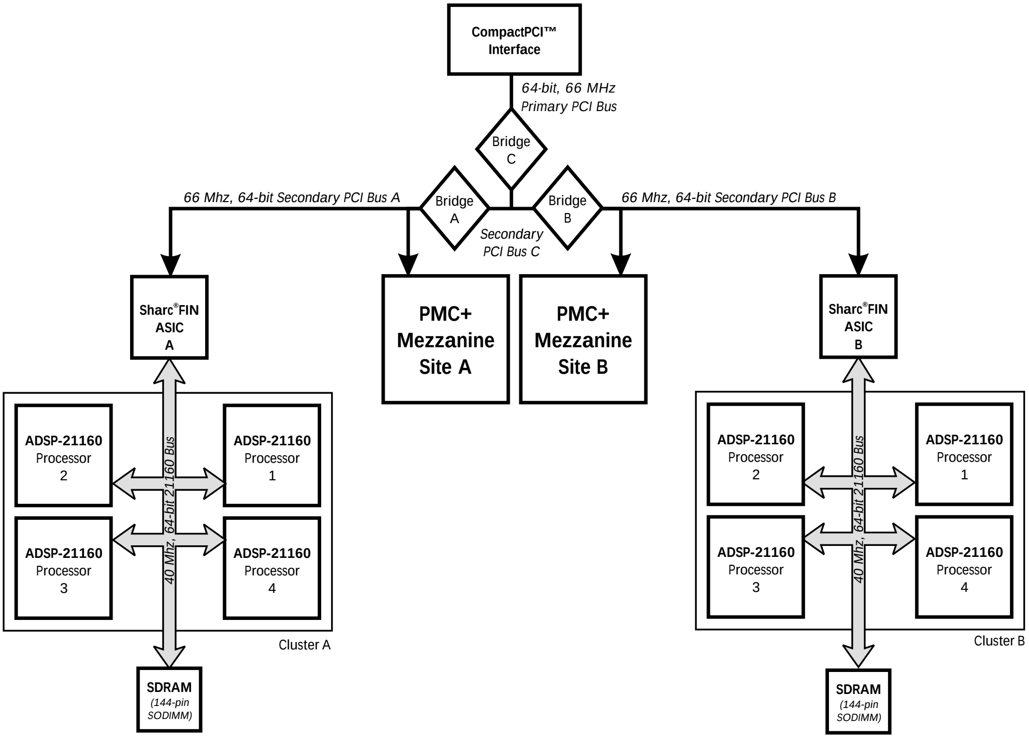


The primary PCI bus is a 66 MHz, 64-bit bus but will operate as a 32-bit bus when it is communicating with 32-bit peripherals. It can run from 0 to 75 MHz, and its maximum data rate is 600 MB/s at 75 MHz and 528 MB/s at 66 MHz.

PCI-to-DSP Interface

The PCI-to-DSP interface allows the ADSP-21160 processors to communicate with the PCI bus, and vice-versa. It consists of the SharcFIN ASICs (PCI-to-DSP bridges) and two buses: secondary PCI bus A and secondary PCI bus B. Figure 4–11 below illustrates the PCI-to-DSP interface.

Figure 4–11 *Block Diagram of the PCI-to-DSP Interface*



Secondary PCI bus A connects PCI-to-PCI bridge A to SharcFIN A, and secondary PCI bus B connects PCI-to-PCI bridge B to SharcFIN B. The SharcFIN ASICs connect the secondary PCI buses to the ADSP-21160 cluster buses. The 66 MHz, 64-bit secondary PCI buses operate at 5 volts but are jumper-configurable to run at 3.3 volts. By default, these buses operate at the same speed as the primary PCI bus, but a jumper setting will allow you to use the buses at half the speed of the primary PCI bus. However, these buses cannot run at 66 MHz when the primary bus is running at 33 MHz.

Note

The secondary PCI buses will operate as 64-bit buses even when the primary bus is connected to a 32-bit bus. Refer to the Intel 21154 (Intel Corporation) manual for more details.

4.3.3 Peripheral Bus

A 20 MHz, 8-bit peripheral bus extends off of each SharcFIN ASIC. The peripheral buses connect to low-speed peripherals, which include the Flash memory, the dual UARTs, and the expansion connectors. The purpose of the peripheral bus is to allow additional components to communicate with the ADSP-21160s without affecting the signal quality of the ADSP-21160 cluster bus. It interfaces to the SharcFIN (see section 4.3.1), which connects it to the ADSP-21160 DSPs and the PCI interface. The peripheral bus operates at either 3.3 volts or 5 volts.

4.4 PMC Interface Architecture

The Hammerhead-6U-cPCI features two PMC interfaces, one for each DSP cluster. This section explains the connections available via the PMC interface, which include a 64-bit 66 MHz PCI interface, link ports, a TDM serial bus, a reset line, and an I^{2}C interface.

Each PMC interface features four connectors. The first three connectors (Jn1–Jn3, according to the PMC standard) provide the 64-bit, 66 MHz PCI interface. The fourth connector (Jn4) provides BittWare's PMC+ extensions.

4.4.1 PMC-to-PCI Interface

The first three connectors from each PMC interface provide the 64-bit, 66 MHz PCI interface. PMC interface A connects to secondary PCI bus A, and PMC interface B connects to secondary PCI bus B.

4.4.2 PMC+ Extensions

The fourth connector from each PMC interface provides the PMC+ extensions, which include link ports, a TDM serial bus, a reset line, and an I^{2}C interface. Please note that the PMC+ extensions are only available when a BittWare PMC+ module is attached to the Hammerhead-6U-cPCI.

Table 4–11 *ADSP-21160 Link and Serial Port Connections to PMC Interfaces*

Cluster A

	21160-A1	21160-A2	21160-A3	21160-A4
Link 0	PMC A Link 1	PMC A Link 2	PMC A Link 3	PMC A Link 4
Sport 0	PMC A TDM	PMC A TDM	PMC A TDM	PMC A TDM

Cluster B

	21160-B1	21160-B2	21160-B3	21160-B4
Link 0	PMC B Link 1	PMC B Link 2	PMC A Link 3	PMC A Link 4
Sport 0	PMC B TDM	PMC B TDM	PMC B TDM	PMC B TDM

Chapter 5
ADSP-21160 Programming Details

This chapter provides programming details for the ADSP-21160 DSPs, discussing how to access the DSPs' memory and how to boot the DSPs. In addition to the information in this chapter, you will also need to refer to the *ADSP-21160 User's Manual* (Analog Devices, Inc.).

5.1 Accessing the DSPs' Memory

5.1.1 Accessing Internal Memory

The DSPs' internal memory space ranges from address 0x0000 0000 through 0x0007 FFFF. Internal memory space refers to the DSPs' on-chip SRAM and memory-mapped registers.

5.1.2 Accessing the DSPs' Multiprocessor Memory

Multiprocessor memory space (MMS) is the on-chip SRAM of other ADSP-21160 DSPs in the same cluster. A cluster is up to six ADSP-21160 DSPs that share a common processor bus, and any DSP that is connected to the processor bus shares the MMS. The Hammerhead-6U-cPCI has four DSPs on board that share a common bus (the ADSP-21160 cluster bus); each processor can view the other three DSPs' on-chip SRAM. Table 5–1 below provides the address region and processor ID variable for each DSP.

Table 5–1 *Multiprocessor Memory Allocation for DSPs*

DSP*	Address Region	Processor ID
21160-1	0x0010 0000 – 0x0017 FFFF	001
21160-2	0x0020 0000 – 0x0027 FFFF	010
21160-3	0x0030 0000 – 0x0037 FFFF	011
21160-4	0x0040 0000 – 0x0047 FFFF	100

* Addresses in this table apply to DSPs in both clusters.

5.1.3 Accessing External Memory Banks

External memory space refers to the off-chip memory or memory-mapped peripherals that are attached to the ADSP-21160 cluster buses. On the Hammerhead-6U-cPCI, these devices include the SDRAM, the Flash, the SharcFIN ASICs, and the UARTs.

The external memory space for each ADSP-21160 DSP has five regions: four banks (bank 0–3) and an unbanked region. The four external memory banks are of equal, programmable size. The remaining area of memory that is not assigned to a bank is the unbanked memory. Mapping peripherals into different banks lets systems accommodate I/O devices with different timing requirements because the banked and unbanked regions have associated waitstate and access mode settings.

The address range for the external memory spans from 0x0080 0000 through 0xFFFF FFFF. The DSP controls access to the four banked regions both with memory select lines (MS0–MS3) and with the memory address; it controls access to the unbanked region with only the memory address. Whenever the DSP generates an address that is located within one of the four banks, the DSP asserts the corresponding memory select line (MS0–MS3).

Table 5–2 *External Memory Bank Allocation*

External Memory Bank	Memory Select Line	Description	Wait State	Wait Mode
0	MS0	SDRAM	1	2
1	MS1	Flash, UART, peripheral bus	7	0
2	MS2	SharcFIN chip control registers	1	2
3	MS3	Unused	7	0

Setting the Size of the External Memory Banks

The MSIZE (Memory Bank Size) bits of the ADSP-21160’s SYSCON (System Configuration) register define the size of the four external memory banks (bank 0–3). Bank 0 starts at 0x0080 0000, and banks 1, 2, 3 and unbanked follow. The size of bank 0 determines the starting address of each of the other banks. (Refer to the *ADSP-21160 SHARC User’s Manual* for more details.)

You can use the BittWare Configuration Manager (see section 2.3.2) to set the MSIZE bits. The default setting for the MSIZE should be equal to the size of the largest external memory device, which is the SDRAM. Table 5–3 lists the recommended settings for MSIZE and shows how MSIZE affects the bank addresses. Note that programming the MSIZE bits may affect where other resources available to the ADSP-21160 processor are located.

Table 5–3 *Recommended MSIZE Settings for the Hammerhead-6U-cPCI*

MSIZE	Size	SDRAM Size
0	8 KWords	
1	16 KWords	
2	32 KWords	
3	64 KWords	
4	128 KWords	
5	256 KWords	
6	512 KWords	
7	1024 KWords	4 MBytes
8	2048 KWords	8 MBytes
9	4096 KWords	16 MBytes
A	8 MWords	32 MBytes
B	16 MWords	64 MBytes
C	32 MWords	128 MBytes
D	64 MWords	256 MBytes
E	128 MWords	512 MBytes
F	256 MWords	

Accessing the SDRAM

The Hammerhead-6U-cPCI supports a bank of up to 512 MB of SDRAM for each DSP cluster. SDRAM bank A is located on ADSP-21160 cluster bus A, and SDRAM bank B is located on ADSP-21160 cluster bus B. The DSPs can access the SDRAM via MS0.

Accessing the Flash, UART, and Peripheral Bus

MS1 provides access to the peripheral bus and all devices located on it, including the Flash memory and the RS-232 dual UART. MS1 from the DSPs in cluster A accesses peripheral bus A, and MS1 from the DSPs in cluster B accesses peripheral bus B.

Accessing the SharcFIN ASIC Chip Control Registers

The DSPs access the SharcFIN ASICs' chip control registers via MS2. The SharcFIN's chip control registers begin at offset 0x00 from the base address of MS2 and control both the PCI interface and the SHARC interface of the SharcFIN (see also section 6.4.1). The SharcFIN's SHARC interface control registers begin at offset 0x40 from the base address of MS2. For additional information on accessing these registers, refer to Chapter 6 of this user's manual and to the *SharcFIN ASIC User's Manual*.

5.1.4 Accessing Unbanked Memory Space

The region of memory above banks 0–3 is called unbanked external memory space. The unbanked memory space begins after external memory bank 3 and covers the remainder of the external memory space up to 0xFFFF FFFF. No MSx memory select line is asserted for accesses in this address space. On the Hammerhead-6U-cPCI board, the unbanked memory space is unused.

5.2 Booting the DSPs

This section explains the three booting options for the Hammerhead-6U-cPCI: link port, Flash, and PCI.

5.2.1 Booting the Board via Link Port

In link port booting, the DSP gets boot data from another DSP's link port or from a four- or 8-bit wide external device[1] after system powerup. Both 21160-A1 and 21160-B1 are connected to external link ports from which they can boot. P3 L1 is the external link port for 21160-A1, and P5 L1 is the link port for 21160-B1 (refer to section 3.2.2 for more information on the external connectors). After booting via link port, 21160-A1 and 21160-B1 can be used to boot the remaining DSPs in their clusters.

To boot the Hammerhead-6U-cPCI via link port,

1. Develop a boot program using Analog Devices VisualDSP++.
2. Using the external link ports, load the boot program onto the DSPs. Analog Devices supplies loading routines (loader kernels) that load an entire program through the selected port. These routines come with the Analog Devices development tools. For more information on loader kernels, refer to the development tools documentation.
3. Link port booting uses DMA channel 8 of the I/O processor to transfer the instructions to internal memory. In this boot mode, the DSP receives 8-bit wide data.
4. After the boot process loads 256 words into memory locations 0x40000 through 0x400FF, the DSP begins executing instructions. Because most DSP programs require more than 256 words of instructions and initialization data, the 256 words typically serve as a loading routine for the application.

For additional information on link booting, refer to the *ADSP-21160 SHARC DSP Hardware Reference Manual* (Analog Devices).

[1] The external device must provide a clock signal to the link port. The clock can be any frequency, up to a maximum of the DSP clock frequency. The clock's falling edges strobe the data into the link port. The most significant 4-bit nibble of the 48-bit instruction must be downloaded first.

5.2.2 Booting the Board From the Host

For host booting, the DSP accepts data from a host processor via the PCI interface. If you are using the DSP21k-SF Toolkit with the Hammerhead-6U-cPCI, the Host Interface Library (HIL) and Diag21k contain functions that will perform the boot process.

To boot the Hammerhead-6U-cPCI from the host using HIL functions or Diag21k commands,

1. Develop a DSP executable program using Analog Devices VisualDSP++.
2. Use HIL functions or Diag21k commands to reset the board and load the program onto the DSPs.
3. Use the HIL's *dsp21k_start* function or Diag21k's Processor Start (**ps**) command to start executing the program.

For additional information on these DSP21k-SF Toolkit functions and commands, refer to the DSP21k-SF Toolkit documentation (BittWare).

To boot the DSPs from the host without using functions from the HIL,

1. Develop the boot program using Analog Devices VisualDSP++.
2. Load the boot program onto the DSPs via the DSPs' external port. Analog Devices supplies loading routines (loader kernels) that load an entire program through the selected port. These routines come with the Analog Devices development tools. For more information on loader kernels, refer to the development tools documentation.
3. The host boot mode uses DMA channel 10 of the DSPs' I/O processor to transfer the instructions to internal memory. In this boot mode, the DSP receives the boot data in 48-bit instructions.
4. After the boot process loads 256 words into memory locations 0x40000 through 0x400FF, the DSP begins executing instructions. Because most DSP programs require more than 256 words of instructions and initialization data, the 256 words typically serve as a loading routine for the application.

For additional information on booting the DSPs from the host without the HIL, refer to the *ADSP-21160 SHARC DSP Hardware Reference Manual* (Analog Devices).

5.2.3 Booting the Board via the Flash

The Flash memory allows you to boot the Hammerhead-6U-cPCI in standalone mode, without a host PC.

Developing the Boot Program

Use the Analog Devices software development tools and BittWare's host interface tools (DSP21k-SF Toolkit) to develop a boot program for the ADSP-21160 processors. The example software included with the Hammerhead-6U-cPCI includes example programs that demonstrate how to create a DSP program in the proper form and then load it into Flash memory so that it automatically boots and begins executing after the board is powered on or the DSPs are reset.

Loading a Boot Program into the Flash Memory

Use programs provided with the Hammerhead-6U-cPCI to load the Flash memory with a boot program.

Chapter 6
SharcFIN Programming Details

This chapter provides a brief functional description of the SharcFIN and describes its SHARC-related control registers. For complete details on the SharcFIN, refer to the *SharcFIN ASIC User's Manual* (BittWare).

6.1 Overview of the SharcFIN

The SharcFIN is the glue that holds the Hammerhead-6U-cPCI board together. It flexibly interfaces the ADSP-21160 cluster to the PCI bus, SDRAM, Flash, dual UART, and I^2C; provides interrupt multiplexers for the ADSP-21160s and PCI; controls the SDRAM; and provides DMA engines for moving data between interfaces.

6.1.1 The Two Sections of the SharcFIN

The SharcFIN consists of two main sections: the PCI interface and the SHARC interface. The PCI interface consists of a full 64-bit, 66 MHz bus mastering PCI interface and includes two DMA transmit channels, two DMA receive channels, and a single PCI read/write channel. Also included in the PCI interface is full I_2O support with the associated mailboxes.

The SHARC interface provides the SHARC specific functionality, which includes the SDRAM interface and control, the peripheral bus, the Flash and dual UART, the interrupt multiplexers, and the I^2C interface.

6.1.2 How the SharcFIN Maps to the PCI and ADSP-21160 Buses

The SharcFIN maps into PCI and the ADSP-21160 cluster bus. It uses PCI Base Address Registers (BARs) to map its various parts onto the PCI bus. BAR0 maps to the SharcFIN's control registers, BAR1 maps to the peripheral bus (Flash and dual UART), BAR2 maps to the ADSP-21160's MMS space, BAR3 is unused in a 32-bit environment and used for the upper 32 bits of address in a 64-bit addressable system, and BAR4 maps to the SDRAM.

The SharcFIN maps into the ADSP-21160 cluster bus space using the MS (memory select) lines of the ADSP-21160s. MS0 maps to the SDRAM, MS1 maps to the Flash, dual UART, and peripheral bus, and MS2 maps to the SharcFIN control registers.

6.2 Function of the SharcFIN PCI Interface

The PCI side of the SharcFIN provides the complete PCI interface. It interfaces the PCI bus and the SHARC interface of the SharcFIN and moves data between them. The SHARC side of the SharcFIN completes the interface, whether it be to SDRAM or to the ADSP-21160s.

The PCI side provides four DMA channels for performing PCI bus mastering DMAs: two are receive (for reads), and two are transmit (for writes). These channels can be run independently and will self-arbitrate for bus access. Along with the DMA channels, the PCI side provides a single PCI access channel for doing single PCI reads and writes and supports interrupts both to and from the PCI bus.

All control registers for the PCI interface are in Base Address Register 0 (BAR0) and occupy byte addresses from 0x00 to 0x100 off of BAR0. For complete details on these registers, refer to the *SharcFIN ASIC User's Manual* (BittWare). From the ADSP-21160, these control registers are at MS2 and are 32-bit addressable, so that they occupy word addresses 0x00 to 0x40 off of MS2. Section 6.4.1 gives an overview of how to access these registers.

6.2.1 Performing PCI Side DMAs

To perform a PCI bus mastering DMA, program a DMA channel in the PCI side of the SharcFIN. Next, program a DMA in the SHARC side to work in conjunction with the PCI side DMA. The PCI side DMA will move the data between the PCI bus and an internal FIFO, and the SHARC side DMA will move data between the internal FIFO and the actual source or destination on the board. If the source or destination is the ADSP-21160's internal memory, the SHARC side DMA used is actually an ADSP-21160 IOP DMA. If the source or destination is the SDRAM, use the internal SharcFIN DMA engine to move the data to/from the SDRAM.

The PCI side DMAs are designed for 64-bit based transfers and expect 64-bit aligned data, regardless of the actual width of the PCI bus. The PCI address used in the transfer is a standard PCI byte based address. For complete details on how to perform PCI side and SHARC side DMAs, refer to the section on "Bus Mastering and DMAs" in the S*harcFIN ASIC User's Manual*.

6.2.2 Performing a PCI Side Single Access

The SharcFIN supports a single PCI access channel for performing single PCI reads and writes. To perform PCI reads and writes, tell the SharcFIN which address to read or write, provide the data (for a write), and then request the transfer. On a read completion, the data is available in a buffer to be read. As with the DMAs, the SharcFIN is designed for 64-bit transfers and alignment. You can make it perform any number of byte width transfers by specifying which of the 8 bytes of the 64-bit access are to be enabled. However, you will need to align the data in the 64-bit word and use the 64-bit aligned address. For complete details on single PCI accesses, refer to the section on “Bus Mastering and DMAs” in the S*harcFIN ASIC User’s Manual*.

6.2.3 Performing PCI Side Interrupts

The SharcFIN provides full I_2O support with the associated mailboxes. To generate PCI side interrupts, either write to PCI outgoing mailboxes or use the SHARC side PCI interrupt multiplexer, which generates the PCI side user interrupt bit.

The PCI side can interrupt the ADSP-21160s by writing into mailboxes. Writing into mailboxes will cause the PCI interrupt bit to be set in the SHARC side interrupt multiplexers, which will generate an ADSP-21160 interrupt if the mask is open. For details on the SharcFIN’s interrupt capabilities, refer to the *SharcFIN ASIC User’s Manual*.

Note

When reading the PCI side documentation of these registers, take careful note of whether you are looking at them from a PCI side or the “user” side. Phrases such as “PCI outgoing” have different meaning depending on your viewpoint, and several mailboxes and registers are duplicated – one for each direction.

6.3 Function of the SharcFIN SHARC Interface

The SHARC interface of the SharcFIN consists of the ADSP-21160 bus interface, the SDRAM controller, the peripheral bus (with Flash and dual UART), the I^{2}C interface, and the interrupt multiplexers. The SharcFIN control registers for the SHARC interface are mapped into PCI in BAR0, starting at byte offset 0x100. On the ADSP-21160 side, they are mapped into MS2, starting at word offset 0x40.

6.3.1 ADSP-21160 Bus Interface

The SharcFIN interfaces to the ADSP-21160 cluster bus as a synchronous host. It sits on the ADSP-21160 bus and will request the bus to complete a PCI side initiated transfer. It also monitors the bus for any accesses to memory spaces it controls, including SDRAM, Flash, dual UART, and the SharcFIN registers.

6.3.2 SDRAM Interface and Control

The SharcFIN's SDRAM controller supports up to 512 Mbytes of SDRAM. It refreshes the SDRAM and controls all of the interfacing from the ADSP-21160s to the SDRAM. In the ADSP-21160 memory space, the SDRAM is mapped into MS0, and the ADSP-21160s have full access to all of the SDRAM.

Accessing SDRAM from the PCI Side

From the PCI side, the SDRAM is mapped into BAR4 with a 16 Mbyte window viewable at a time. Because the SDRAM is so large, this window exists to keep the entire SDRAM from being mapped into PCI memory. A SharcFIN control register (the SD Size Config register at word offset 0x45), which provides the upper address bits for a PCI initiated SDRAM access, sets the window location.

The SDRAM window has the following two limitations:

1. Window boundaries must be crossed carefully.
2. The window register is a shared resource.

The Host Interface Library (in BittWare's DSP21k-SF Toolkit) takes care of the first limitation. The second limitation is a system issue that you must consider. Because the SharcFIN uses the window register for every PCI access to SDRAM, be careful to coordinate SDRAM accesses from PCI if you have multiple threads on the host or multiple PCI bus masters accessing the SDRAM.

SDRAM Timing from the ADSP-21160

SDRAM timing from the ADSP-21160 is synchronous, 1 wait state. A single write access takes two bus cycles. Since each additional write is single cycle, using the ADSP-21160's burst mode, you can achieve a four word burst write in five bus cycles. Reads require additional setup in the SDRAM, resulting in four bus cycles for the first access and a four word burst read in seven cycles. Because the SDRAM is page based, you will encounter additional latencies when page boundaries are crossed.

Using DMA-Based SDRAM Accesses

To achieve optimal system performance, use the power of the ADSP-21160's IOP DMA engines and its dual ported internal RAM. Using these features, you can perform DMA-based SDRAM accesses at the same time that the ADSP-21160 core is performing processing on its internal data space, which is full core speed, 0 wait state memory.

6.3.3 Peripheral Bus Interface (Flash and Dual UART)

The peripheral bus is an 8-bit wide bus containing the Flash and dual UART. On the Hammerhead-6U-cPCI board, optional headers for custom applications are also located on the peripheral bus. The peripheral bus is mapped into the ADSP-21160 space as MS1 and into PCI space as BAR1. From the ADSP-21160, the Flash and UART are also at MS1. From PCI, the Flash occupies the first 2 Mbytes of BAR1, and the UART sits at a 2 Mbyte offset from BAR1.

6.3.4 I^{2}C Interface

The SDRAM and configuration EEPROM sit on an I^{2}C bus that is connected to the SharcFIN. The SDRAM is interrogated over the I^{2}C to determine its size and type so that the SDRAM configuration registers can be written. The Host Interface Library (included with BittWare's DSP21k-SF Toolkit) sets up the SDRAM on a board reset command.

The EEPROM contains factory programmed board information, including a serial number and factory build date.You can use the BittWare Configuration Manger (bwcfg) to view this information. Space for the user is also reserved in the EEPROM. The I^2C interface in the SharcFIN is the low level clock and data lines for the I^2C available in a control register. Perform all bit manipulation through software. Along with the on-board I^2C, the SharcFIN supports a second I^2C bus called the PMC I^2C, which is pinned out to the PMC+ connector.

6.3.5 Interrupt Multiplexer

The SharcFIN contains a flexible interrupt multiplexer that you can use to create complex interrupt schemes on the Hammerhead-6U-cPCI board. The interrupt multiplexer contains an interrupt multiplexer for each ADSP-21160, the PCI, and the PMC. Inputs to the multiplexer include a flag from each ADSP-21160, a PCI side flag, a PMC flag, two UART flags, and a DMA interrupt. A second flag from each ADSP-21160 also functions as an input to the multiplexer but can only be used to generate interrupts to other ADSP-21160s. Outputs from the multiplexer are an interrupt line to each ADSP-21160, the PCI side, and the PMC site.

How the Interrupt Multiplexer Functions

The interrupt multiplexer for each output is completely independent and can handle multiple input sources. Each interrupt multiplexer consists of a 32-bit configuration register that selects the desired interrupt sources and then masks the results (see section 6.5.11 for a description of the registers). To generate an interrupt, both the flag input from the desired source and its corresponding bit in the configuration register must be high. Any other flag input and its corresponding bit in the register can also be high to generate an interrupt. The interrupt multiplexer ANDs each flag input and its corresponding bit; it then ORs the results of the inputs together to create the output.

Note

The interrupt multiplexer is level sensitive and does not latch interrupt sources. Therefore, the interrupt is active as long as the source is driven.

Creating PCI Side Interrupts

To create PCI side interrupts, configure the multiplexer, which will generate the “user side” flag into the PCI side interrupt mechanism. The PCI side must then “open” the interrupt.

PCI side interrupts into the SharcFIN via the I^{2}O mailbox registers show up as PCI flags into the SHARC side interrupt multiplexer. Therefore, you can program the ADSP-21160s to respond to PCI interrupts as desired.

6.4 Overview of the SharcFIN Memory Map

The SharcFIN maps into PCI and the ADSP-21160 cluster bus. The following section provides an overview of the PCI and ADSP-21160 memory mapping of the SharcFIN. All addresses are shown as offsets from the appropriate BAR or MS. Table 6–1 gives an overview of how the SharcFIN maps to the PCI and ADSP-21160 cluster buses.

Note

The SharcFIN ASIC User's Manual contains descriptions of the registers listed in this section. Refer to it for specifics. If you cannot find sufficient information, contact BittWare for more detail.

Table 6–1 *Overview of How the SharcFIN Maps to the PCI and ADSP-21160 Buses*

PCI Base Address Register	Description	21160 Memory Select	Description
BAR0	SharcFIN control registers	MS0	SDRAM
BAR1	Peripheral bus (Flash, UART)	MS1	Peripheral Bus (Flash, UART)
BAR2	ADSP-21160 MMS	MS2	SharcFIN control registers
BAR4	SDRAM		

Even though the following sections list both 32-bit (word) and byte addresses, some BARs should be accessed in specific ways from the PCI side. Table 6–2 shows how to access those BARs.

Table 6–2 *Accessing BAR0–BAR4 From the PCI Side*

BAR	Access	Description
BAR0	Read / Write	Byte or 32-bit word accesses for all registers
BAR1	Read / Write	Byte accesses for all registers*
BAR2	Read Only Write Only	Byte or word accesses Word accesses only†
BAR4	Read Only Write Only	Byte or word accesses Word accesses only†

* Word accesses will produce erroneous data.

† Byte writes will corrupt the rest of the word.

6.4.1 Accessing System Settings and Configuration Registers

BAR0 = MS2 = system settings and configuration registers

BAR0 from the PCI interface and MS2 from the ADSP-21160 cluster bus map to system settings and configuration registers in the SharcFIN. Table 6–3 gives the PCI and ADSP-21160 offset addresses for accessing system settings and configuration registers.

Table 6–3 *PCI and ADSP-21160 Addresses for System Settings and Configuration Registers*

PCI 32-bit offset from BAR0	PCI byte offset from BAR0	ADSP-21160 offset from MS2	Description
0x00 – 0x5F	0x000 – 0x17F	0x00 – 0x5F	Chip control registers (PCI and ADSP-21160)

6.4.2 Accessing the Flash, UART, and Peripheral Bus

BAR1 = MS1 = Flash/UART/Peripheral bus

BAR1 from the PCI interface maps to the Flash, dual UART, and peripheral bus. MS1 from the ADSP-21160 cluster bus also maps to the Flash, the dual UART, and the peripheral bus. Table 6–4 gives the PCI and ADSP-21160 offset addresses for accessing them.

Warning

*BAR1 **must** be accessed a byte at a time from the PCI side. Word accesses will produce erroneous data.*

Table 6–4 *PCI and ADSP-21160 Addresses for Flash, UART, and Peripheral Bus*

PCI byte offset from BAR1	ADSP-21160 offset from MS1	Description
0x000000 – 0x1FFFFF	0x000000 – 0x1FFFFF	Flash
0x200000 – 0x20000F	0x200000 – 0x20000F	UART
0x200010 – 0x2FFFFF	0x200010 – 0x3FFFFF	Reserved
0x300000 – 0x3FFFFF	0x400000 – 0x5FFFFF	Peripheral Bus

6.4.3 Accessing Multiprocessor Memory Space

BAR 2 = MMS = flat map of Multiprocessor Memory Space

BAR2 from the PCI and MMS from the ADSP-21160 cluster bus allow access to the ADSP-21160 multiprocessor memory space. Table 6–5 gives the PCI and ADSP-21160 offset addresses for accessing the MMS.

Table 6–5 *PCI and ADSP-21160 Addresses for Multiprocessor Memory Space*

PCI 32-bit offset from BAR2	PCI byte offset from BAR2	ADSP-21160 address	Description
0x00000 – 0x0FFFFF	0x0000000 – 0x03FFFFF	0x000000 – 0x0FFFFF	Reserved
0x10000 – 0x1FFFFF	0x0400000 – 0x07FFFFF	0x100000 – 0x1FFFFF	21160-1 MMS space (ADSP-21160 ID 1)
0x20000 – 0x2FFFFF	0x0800000 – 0x0BFFFFF	0x200000 – 0x2FFFFF	21160-2 MMS space (ADSP-21160 ID 2)
0x30000 – 0x3FFFFF	0x0C00000 – 0x0FFFFFF	0x300000 – 0x3FFFFF	21160-3 MMS space (ADSP-21160 ID 3)
0x40000 – 0x4FFFFF	0x1000000 – 0x13FFFFF	0x400000 – 0x4FFFFF	21160-4 MMS space (ADSP-21160 ID 4)
0x50000 – 0x7FFFFF	0x1400000 – 0x1FFFFFF	0x500000 – 0x7FFFFF	Reserved

6.4.4 Accessing SDRAM

BAR 4 = MS0 = SDRAM[1]

You can see a window of 16 Mbytes of SDRAM from the PCI bus. The SD Window register allows you to select which 16 MB window is currently visible. The register is located at word/ADSP-21160 offset 0x4A in BAR0/MS2. Table 6–6 gives the PCI and ADSP-21160 offset addresses for accessing a 64 MB bank of SDRAM, and Table 6–7 gives addresses for a 128 MB bank.

Note

The addresses listed in Table 6–6 only apply to the given 64 MB and 128 MB SDRAM cases. Different memory sizes change the mapping.

Table 6–6 *PCI and ADSP-21160 Addresses for 64 MB SDRAM*

SD Window Register value*	PCI 32-bit offset from BAR4	PCI byte offset from BAR4	ADSP-21160 address	Description
0x02	0x00000 – 0x3FFFFF	0x000000 – 0xFFFFFF	0x800000 – 0xBFFFFF	First 16 MB block of SDRAM
0x03	0x00000 – 0x3FFFFF	0x000000 – 0xFFFFFF	0xC00000 – 0xFFFFFF	Second 16 MB block of SDRAM
0x00	0x00000 – 0x3FFFFF	0x000000 – 0xFFFFFF	0x1000000 – 0x13FFFFF	Third 16 MB block of SDRAM
0x01	0x00000 – 0x3FFFFF	0x000000 – 0xFFFFFF	0x1400000 – 0x17FFFFF	Fourth 16 MB block of SDRAM

* Window register values of 0x00 and 0x01 always access the last two 16 MB windows of SDRAM. Refer to section 6.3.2 for details.

[1] Some caveats apply.

Table 6–7 *PCI and ADSP-21160 Addresses for 128 MB SDRAM*

SD Window Register value[*]	PCI 32-bit offset from BAR4	PCI byte offset from BAR4	ADSP-21160 address	Description
0x02	0x00000 – 0x3FFFFF	0x000000 – 0xFFFFFF	0x800000 – 0xBFFFFF	First 16 MB block of SDRAM
0x03	0x00000 – 0x3FFFFF	0x000000 – 0xFFFFFF	0xC00000 – 0xFFFFFF	Second 16 MB block of SDRAM
0x04	0x00000 – 0x3FFFFF	0x000000 – 0xFFFFFF	0x1000000 – 0x13FFFFF	Third 16 MB block of SDRAM
0x05	0x00000 – 0x3FFFFF	0x000000 – 0xFFFFFF	0x1400000 – 0x17FFFFF	Fourth 16 MB block of SDRAM
0x06	0x00000 – 0x3FFFFF	0x000000 – 0xFFFFFF	0x1800000 – 0x1BFFFFF	Fifth 16 MB block of SDRAM
0x07	0x00000 – 0x3FFFFF	0x000000 – 0xFFFFFF	0x1C00000 – 0x1FFFFF	Sixth 16 MB block of SDRAM
0x00	0x00000 – 0x3FFFFF	0x000000 – 0xFFFFFF	0x2000000 – 0x23FFFFF	Seventh 16 MB block of SDRAM
0x01	0x00000 – 0x3FFFFF	0x000000 – 0xFFFFFF	0x2400000 – 0x27FFFFF	Eighth 16 MB block of SDRAM

[*] Window register values of 0x00 and 0x01 always access the last two 16 MB windows of SDRAM. Refer to section 6.3.2 for details.

6.5 Setting the SharcFIN’s SHARC Interface Control Registers

The SharcFIN has two sets of registers. One set, the PCI configuration registers, configures the PCI interface. The other set, the chip control registers, configures both the PCI and SHARC interfaces. The PCI configuration registers are only accessible by PCI (configuration access) and are documented in the *SharcFIN ASIC User’s Manual*. The chip control registers are broken into two groups: the PCI control registers (which are documented in the *SharcFIN ASIC User’s Manual*) and the SHARC interface control registers; both groups are accessible by PCI (BAR0) and the ADSP-21160 cluster bus (MS2).

This section describes the memory locations and settings for the SharcFIN’s SHARC interface control registers. All addresses described in this section are 32-bit addresses and are accessible from the ADSP-21160 DSPs (via MS2) and from the PCI interface (via BAR0). Table 6–8 gives the memory mapping for the SHARC interface control registers in the SharcFIN.

Note

Most of the SHARC interface control registers are already set and do not require you to program them. You will only need to set them if you are writing your own host interface programs.

Table 6–8 *Memory Map for the SHARC Interface Control Registers*

Address	Register	Type	Description
0x40	Address Override	R/W	Allows addressing of IOP registers when ADSP-21160 is using a host packing mode
0x41	Status	R/O	General set of status registers. Indicates ADSP-21160 cluster bus status, and last reset source
0x42	Peripheral Bus Configuration	R/W	Configures and shows status of wait cycles of the 8-bit peripheral bus
0x43	Watchdog Configuration	WORM*	Enables and disables the watchdog timer
0x44	PMC+ Configuration	R/W	Configures the PMC+ interface
0x45	SD Size Config	R/W	Resets and reinitializes the SDRAM controller
0x46	Onboard I^{2}C Control	R/W	Controls the I^{2}C interface
0x47	PMC I^{2}C Control	R/W	Controls the I^{2}C interface to the PMC+ interface
0x48	DMA Address	R/W	Sets the address of DMA to be performed
0x49	DMA Configuration	R/W	Controls various features of the SharcFIN DMA engine: size of DMA, increment size of DMA, other various configuration bits
0x4A	SD Window	R/W	Selects which 16 MB of SDRAM the PCI interface will view
0x4B–4F	Unused		
0x50,52, 54,56	H1I0, H2I0, H3I0, H4I0	R/W	Configure ADSP-21160 DSPs' interrupts; show status of interrupts
0x51,53, 55,57	Reserved		
0x58	PCInt	R/W	Configures PCI interrupts
0x5A	PMCI0	R/W	Configures PMC interrupt
0x59, 0x5B–5D	Unused		
0x5E	Flag Status	R/O	Shows state of all flags
0x5F	IRQ Status	R/O	Shows state of all interrupts

* Write Once Read Many

6.5.1 Address Override Register

The Address Override register (offset 0x40) configures how the ADSP-21160 DSPs access the least significant 32 bits on the ADSP-21160 cluster bus. It allows access to the DSPs' IOP space before the ADSP-21160's SYSCON register has been configured.

> **Note**
>
> *Only use this register if you are writing your own host interface programs.*

Table 6–9 *Address Override Register Description*

Bit	Name	Type	Reset Value	Function
0	A0 Override En*	R/W	0	Address override enable for booting across the PCI bus
1	Overridden A0	W/O	0	Overridden address
2	BusLockReq	W/O	0	Bus Lock Request. Requests the SharcFIN to acquire the ADSP-21160 cluster bus and locks access to the bus so that only the SharcFIN can access it. 0 = Disabled 1 = Requests that the SharcFIN acquire the ADSP-21160 cluster bus and not give it up
3	Destructive FIFO Read Enable	W/O	1	Determines whether a read to the DMA FIFOs will cause the FIFOs to advance 0 = A read to the DMA FIFOs does not cause the FIFOs to advance 1 = A read to the DMA FIFOs causes the FIFOs to advance
4	Host Clock Disable	R/W	0	Disables the clock to the ADSP-21160 DSPs. This bit addresses power up anomalies on current editions of ADSP-21160 processors. 1 = Clock disabled 0 = Clock enabled

* Do not use the Address Override bits (B0 and B1) under normal setup conditions. If you are running the Hammerhead-6U-cPCI in standalone mode and are booting across the PCI bus, you can change these bits. However, exercise extreme caution since data loss or corruption will occur if you set the bits improperly.

6.5.2 Status Register

The Status register (offset 0x41) is a 16-bit read only register that gives information about various features on the board.

Table 6–10 *Contents of the Status Register*

Bit	Name	Type	Reset Value	Function
0	PMCHostCfg	R/O	Set in hardware	Indicates whether SharcFIN is configured for a PMC host site or a PMC daughter card. 1 = On baseboard 0 = On PMC
1	StandAlone	R/O	Set by jumper	Determines whether PCI side resets are accepted by the SharcFIN 1 = PCI resets ignored 0 = PCI resets accepted
2	Bus Locked	R/O	0	Indicates whether the SharcFIN has locked and acquired the ADSP-21160 cluster bus 0 = Cluster bus is not locked 1 = Cluster bus is locked
3	Last Reset Source	R/O	0	Indicates whether the PCI interface or the watchdog was the source of the last board reset 0 = PCI reset 1 = Watchdog/external reset
4	SpareInput Pin	R/O	1	Spare external input signal

6.5.3 Peripheral Bus Configuration Register

The Peripheral Bus Configuration register (offset 0x42) allows you to configure the wait cycles of the peripheral bus.

Table 6–11 *Contents of the Peripheral Bus Configuration Register*

Bit	Name	Type	Reset Value	Function
3:0	PCI to Pbus Wait	R/W	0101 B0: 1 B1: 0 B2: 1 B3: 0	Select the number of wait cycles the SharcFIN will wait before completing a transaction on the peripheral bus. The actual value of wait cycles is one greater than the value in the register (for example, if the register value = 0, the number of wait cycles = 1). 0101 =Default setting (6 wait cycles)
4	Pbus Ack Enable	R/W	0	Selects whether the SharcFIN will monitor the peripheral bus Ack line after the peripheral bus wait time has expired. 0 = SharcFIN will wait the selected number of wait cycles and consider the transaction complete* 1 = SharcFIN will wait the selected number of wait cycles and then monitor the Ack line
5	Pbus Reset	R/W	0	Resets the peripheral bus reset line, the Flash, and the UART. The reset stays active until cleared by another write to the register.† 0 = No reset 1 = Resets Flash, UART, and all devices on the peripheral bus

* Five wait cycles is the minimum amount of wait cycles required to talk to the Flash memory.

† You can also reset the Flash, the UART, and all devices on the peripheral bus via a board reset.

6.5.4 Watchdog Configuration Register

The Watchdog Configuration register (offset 0x43) is a WORM (Write Once Read Many) register that allows you to enable or disable the watchdog timer, set its time-out time, and select which processor will reset its timer. Once the watchdog is enabled, it cannot be disabled except by a board reset (hence the WORM designation), which can be from the PCI interface, the watchdog, or an external source.

Table 6–12 *Contents of the Watchdog Configuration Register*

Bit	Name	Type	Reset Value	Function
1:0	WDEn1, WDEn0	WORM	00	Enable the watchdog timer and select its time-out time. 00 = Disabled 10 = Enabled; short time-out (200 ms) 01 = Enabled; medium time-out (600 ms) 11 = Enabled; long time-out (1.2 s)
3:2	Unused			
7:4	H4F1 En, H3F1 En, H2F1 En, H1F1 En	WORM	0000	Selects which processor will strobe the watchdog timer.* 0001 = 21160-1 FLAG1 will strobe the watchdog timer 0010 = 21160-2 FLAG1 will strobe the watchdog timer 0100 = 21160-3 FLAG1 will strobe the watchdog timer 1000 = 21160-4 FLAG1 will strobe the watchdog timer

* You can select more than one processor, but it is not recommended.

6.5.5 PMC+ Configuration Register

The PMC+ Configuration register (offset 0x44) is a read/write register that configures the bus mode lines of the PMC+ interface and allows you to read their status. Table 6–13 below shows the contents of the PMC+ Configuration register.

Table 6–13 *Contents of the PMC+ Configuration Register*

Bit	Name	Type	Reset Value	Function
0	PMC Flg/Int En	R/W	0	Configures the PMC+ interface's bus mode lines to be used as flag interrupts.[*] 1 = The PMC+ interface's bus mode lines will be used as flag interrupts 0 = The PMC+ interface's bus mode lines will be used as bus mode lines

3:1	BusMode2,
	BusMode3,
	BusMode4
		R/W	BusMode2: 1
			BusMode1: 0
			BusMode3: 0
				Tells the PMC site whether or not it should drive BusMode1 to indicate its presence.
				When the flag interrupts are disabled (by *PMC Flg/Int En* bit), the BusMode lines work according to the PMC specification. Bits 1–3 are Bus Mode lines 2–4.
				B1 = 1 Bus Mode line 2
				B2 = 0 Bus Mode line 3
				B3 = 0 Bus Mode line 4

4	BusMode 1	R/O		Bus Mode line 1 is an input[†]. It indicates that a PMC card is present on the board.[‡] 0 = PMC board is present
5	BusMode2	R/O		Current status of BusMode2 line
6	BusMode3	R/O		Current status of BusMode3 line
7	BusMode4	R/O		Current status of BusMode4 line

[*] The option of using the bus mode lines as flag interrupts is a feature of BittWare's PMC+ form factor; to work properly, it must be enabled on both the PMC+ card and the host board.

[†] Bits 3:1 are outputs. Bit 4 is an input.

[‡] Refer to the *IEEE P138.1 Standard Physical and Environmental Layers for PCI Mezzanine Cards: PMC* (PMC Specification) for details on the operation of these lines.

6.5.6 Onboard I^{2}C Control Register

The Onboard I^{2}C Control register (offset 0x46) controls the I^{2}C interface. The I^{2}C interface is a two-wire bus; one wire is a clock signal, and the other is a data signal. Both the clock and the data lines are pulled up. Table 6–14 shows the contents of the register.

As per standard I^{2}C, both the clock and data lines are pulled high. Devices on the I^{2}C bus either do not drive the bus, or they drive it low. Any device on the I^{2}C bus can drive either the clock or data line low when required. You can also read the actual status of the lines.

Table 6–14 *Contents of the I^{2}C Control Register*

Bit	Name	Type	Reset Value	Function
0	Clock	R/W	1	On write, drives the clock line. On read, shows the state of the clock line
1	Data	R/W	1	On write, drives the data line. On read, shows the state of the data line
2	Clock Drive	R/O	1	
3	Data Drive	R/O	1	

When you write a 1 to either the clock or data line in this register, the SharcFIN does not drive the corresponding line. When you write a 0 to either the clock or the data line, the SharcFIN drives the corresponding line to 0. When you read either line, you read the actual state of the line rather than what you have written to it. If you are not driving the line, it will be 0 if another device is driving it and 1 if nothing is driving it. Table 6–15 shows the effect of the values written to the Clock and Data bits.

Table 6–15 *Effects of Values Written to the Clock and Data Bits (B0, B1)*

Value Written	Description
0	Drives the line low; when read back, shows 0
1	When read back, shows the actual state of the I^{2}C line

6.5.7 PMC+ I^{2}C Control Register

The PMC+ I^{2}C Control register (offset 0x47) controls the I^{2}C interface to the PMC+ interface. The settings for this register are the same as the settings for the Onboard I^{2}C register, except that all settings apply to the PMC+ interface I^{2}C instead of the on-board I^{2}C.

6.5.8 SDRAM Configuration Registers

The SharcFIN contains two registers that configure the SDRAM:

- SDRAM Size Configuration Register (offset 0x45)
- SDRAM Window Register (offset 0x4A)

SDRAM Size Configuration Register

The SDRAM Size Configuration register sets the size of the SDRAM. The settings for this register depend on the type of SDRAM modules used on the DSP board. Table 6–16 below shows the contents of the register.

Table 6–16 *Contents of the SDRAM Size Configuration Register*

Bit	Name	Type	Reset Value	Function
1:0	SD Bank Size 1:0	R/W	0	Determine how the SDRAM controller uses the SharcFIN's CS0 and CS1 (chip select 0 and 1) pins. The chip select pins allow the SharcFIN to seamlessly connect to two banks of SDRAM. CS0 selects SDRAM bank 0, and CS1 selects SDRAM bank 1. **B0 B1 Result** 0 0 CS0 always active 0 1 CS1 active when 32-bit word address bit 24 is high; otherwise CS0 is active 1 0 CS1 active when 32-bit word address bit 25 is high; otherwise CS0 is active 1 1 CS1 active when 32-bit word address bit 26 is high; otherwise CS0 is active
2	SD RF Size	R/W	1	Sets the refresh rate of the SDRAM 0 = 4K refreshes every 64 milliseconds 1 = 8K refreshes every 64 milliseconds
3	SD Reset	W/O	0	Writing a 1 resets the SDRAM controller and reinitializes the SDRAM

SDRAM Window Register

The SDRAM Window register (offset 0x4A) is a 5-bit register that lets you select which 16 MB section of memory in the SDRAM to view from the host over the PCI interface. From the PCI side, the SDRAM is mapped into BAR4 with a 4 Mword (16 Mbyte) window viewable at a time. The offset into BAR4 provides bits 0 through 21 of the address of the SDRAM word to be accessed. The contents of the 5-bit SD Window register are appended to bits 22 through 26 of the address to complete the address and thereby select the 4 Mword window to be accessed. Table 6–17 lists the bits included in this register. For more information, refer to the *SharcFIN ASIC User's Manual*.

Note

You will not need to configure this register since the Diag21k utility, which is included with the DSP21k-SF Toolkit, will set these bits.

Table 6–17 *Contents of the SDRAM Window Register*

Bit	Name	Type	Reset Value	Description
0	Window A0	W/O	0	Selects window A0 of the SDRAM
1	Window A1	W/O	1	Selects window A1 of the SDRAM
2	Window A2	W/O	0	Selects window A2 of the SDRAM
3	Window A3	W/O	0	Selects window A3 of the SDRAM
4	Window A4	W/O	0	Selects window A4 of the SDRAM

6.5.9 DMA Address Register

The DMA Address register (offset 0x48) configures the address of the current DMA location. This register is incremented as the DMA progresses, allowing you to monitor the DMA engine's current address. You must reset this register each time if you wish to repeat a DMA on the same address range as last time. This register cannot be written to while the DMA start bit is set, but it can be read from at any time.

Table 6–18 *Contents of the DMA Address Register*

Bit	Name	Type	Reset Value	Function
0	Unused	R/O	0	
27:1	A[27:1]	R/W	0	Indicates the current DMA location
31:28	Unused	R/O	0	

6.5.10 DMA Configuration Register

The DMA Configuration register (offset 0x49) controls various features of the SharcFIN DMA engine, including starting a DMA, size of the DMA, increment size of the DMA, and other various configuration bits. Table 6–19 describes the contents of the register. The *SharcFIN ASIC User's Manual* explains this register in more detail. This register can not be written while the DMA start bit is set, but it can be read from at any time.

Table 6–19 *Contents of the DMA Configuration Register (Continues on next page)*

Bit	Name	Type	Reset Value	Description
15:0	DMACntB[15:0]	R/W	X*	DMA transfer count (in 64-bit words) bits. All are settable.
22:16	DMA Stride B[6:0]	R/W	X	Stride or address increment bits. These bits will typically be set to 0x01.
23	Unused			Unwritable; fixed at 0.
24	DMAStart	R/W	0	Setting the bit to 1 starts the DMA; it resets to 0 when the DMA is complete.
25	DMA Channel Select	R/W	0	Selects the PCI channel being operated on (0 or 1)
26	DMA Direction	R/W	0	Selects whether a PCI transmit or receive DMA is being performed. 0 = PCI receive DMA (PCI to 21160/SDRAM) 1 = PCI transmit DMA (21160/SDRAM to PCI)
27	DMA Interrupt	R/W	0	If this bit is set at the start of the DMA, The SharcFIN generates an interrupt in the interrupt multiplexer on completion of the DMA. Any write to this register clears the interrupt.
28	Burst Disable	R/W	0	1 = Disable bursting on the ADSP-21160 side. Must be set to 1 when the address increment > 0x01. If it is set when the address increment <= 0x01, it functions but will slow things down. 0 = Bursting enabled
29	DMA Buslock		0	1 = SharcFIN requests the ADSP-21160 cluster bus when the start bit is set, and once it obtains the bus, keeps it until the DMA completes.
(Sheet 1 of 2)				

Bit	Name	Type	Reset Value	Description
31:30	DMA Xfer Length B[1:0]		00	Set the DMA transfer length. These bits must be set along with the bits in the PCI control register at 0x1A and 0x1B (0x68 byte address). **B30 B31 Result** 0 0 Data transferred during each access of the bus = eight 64-bit words 1 0 Data transferred during each access of the bus = sixteen 64-bit words 0 1 Data transferred during each access of the bus = thirty-two 64-bit words 1 1 Data transferred during each access of the bus = sixty-four 64-bit words Corresponding Receive FIFO almost empty or Transmit FIFO almost full flags must be set with corresponding value (length –1, so for B30, B31 must be set to 7).
(Sheet 2 of 2)				

* X = Unknown

6.5.11 Interrupt Configuration Registers

The SharcFIN features an interrupt multiplexer for each ADSP-21160, the PCI interface, and the PMC+ interface. Inputs to the multiplexers are flags from each ADSP-21160, a PCI side flag, PMC flags, and UART flags, a DMA flag, and a peripheral bus flag. The registers at offsets 0x50 to 0x5A (see Table 6–20) provide the interrupt multiplexers (see the *SharcFIN ASIC User's Manual* for additional details on the interrupt multiplexer). Table 6–21 lists the settings for the ADSP-21160 interrupt multiplexers, Table 6–22 lists the settings for the PCI interrupt multiplexer, and Table 6–23 lists the settings for the PMC+ interrupt multiplexer.

The interrupt multiplexer registers are 32-bit registers that allow you to select the desired input sources. The first 16 bits (0–15) are read/write and select the source that will generate an interrupt to the processor; each of the bits corresponds to one of the flag inputs to the multiplexer. The second 16 bits (16–31) are read only and show which of the enabled interrupts are generating an interrupt, each bit corresponding to one of the flag inputs. Bits 16–31 are masked interrupt lines; when one of the flag inputs and its corresponding bit in bits 0–15 of the configuration register is high, the corresponding bit in bits 16–31 is also set to indicate the source of the input. Bits 16–31 are masked by 21160-1 IRQ0's interrupt mask.

Table 6–20 *SharcFIN Interrupt Configuration Registers*

Address	Register	Description
0x50	H1I0	Configures the direction of 21160-1 IRQ0
0x51, 53, 55, 57	Unused	
0x52	H2I0	Configures the direction of 21160-2 IRQ0
0x54	H3I0	Configures the direction of 21160-3 IRQ0
0x56	H4I0	Configures the direction of 21160-4 IRQ0
0x58	PCInt	Configures the direction of the PCI interrupt
0x59	Unused	
0x5A	PMCI0	Configures the direction of PMC+ IRQ0

Table 6–21 *Contents of the ADSP-21160 Interrupt Configuration Registers (0x50, 0x52, 0x54, 0x56) (Continues on next page)*

Bit	Name	Type	Reset Value	Description*
0	H1F0	R/W	0	Enables 21160-1 FLAG0 to cause an interrupt
1	H1F1	R/W	0	Enables 21160-1 FLAG1 to cause an interrupt
2	H2F0	R/W	0	Enables 21160-2 FLAG0 to cause an interrupt
3	H2F1	R/W	0	Enables 21160-2 FLAG1 to cause an interrupt
4	H3F0	R/W	0	Enables 21160-3 FLAG0 to cause an interrupt
5	H3F1	R/W	0	Enables 21160-3 FLAG1 to cause an interrupt
6	H4F0	R/W	0	Enables 21160-4 FLAG0 to cause an interrupt
7	H4F1	R/W	0	Enables 21160-4 FLAG1 to cause an interrupt
8	PCFlg	R/W	1	Enables the PCI interface to cause an interrupt
9	PMCFlg0	R/W	0	Enables FLAG0 from the PMC interface to cause an interrupt
10	Unused		0	
11	PRFlg	R/W	0	Enables a flag from the peripheral bus to cause an interrupt
12	UART0	R/W	0	Enables a flag from UART0 to cause an interrupt
13	UART1	R/W	0	Enables a flag from UART1 to cause an interrupt
14	DMAInterrupt	R/W	0	Enables a DMA interrupt
15	Unused		0	
16	H1F0	R/O	0	Indicates that 21160-1 FLAG0 is generating an interrupt
17	H1F1	R/O	0	Indicates that 21160-1 FLAG1 is generating an interrupt
18	H2F0	R/O	0	Indicates that 21160-2 FLAG0 is generating an interrupt
19	H2F1	R/O	0	Indicates that 21160-2 FLAG1 is generating an interrupt
(Sheet 1 of 2)				

Bit	Name	Type	Reset Value	Description*
20	H3F0	R/O	0	Indicates that 21160-3 FLAG0 is generating an interrupt
21	H3F1	R/O	0	Indicates that 21160-3 FLAG1 is generating an interrupt
22	H4F0	R/O	0	Indicates that 21160-4 FLAG0 is generating an interrupt
23	H4F1	R/O	0	Indicates that 21160-4 FLAG1 is generating an interrupt
24	PCFlg	R/O	0	Indicates that PCI flag is generating an interrupt
25	PMCFlg0	R/O	0	Indicates that PMC+ FLAG0 is generating an interrupt
26	Unused		0	
27	PRFlg	R/O	0	Indicates that Peripheral bus flag is generating an interrupt
28	UART0	R/O	0	Indicates that UART0 flag is generating an interrupt
29	UART1	R/O	0	Indicates that UART1 flag is generating an interrupt
30	DMAInterrupt	R/O	0	Indicates that DMA interrupt is generating an interrupt
31	Unused		0	
(Sheet 2 of 2)				

* All descriptions in this column apply when bits are set to 1.

Table 6–22 *Contents of the PCI Interrupt Configuration Register (0x58) (Continues on next page)*

Bit	Name	Type	Reset Value	Description*
0	H1F0	R/W	1	Enables 21160-1 FLAG0 to cause an interrupt
1, 3, 5, 7	Unused		0	
2	H2F0	R/W	1	Enables 21160-2 FLAG0 to cause an interrupt
4	H3F0	R/W	1	Enables 21160-3 FLAG0 to cause an interrupt
6	H4F0	R/W	1	Enables 21160-4 FLAG0 to cause an interrupt
8	PCFlg	R/W	0	Enables the PCI interface to cause an interrupt
9	PMCFlg0	R/W	0	Enables FLAG0 from the PMC interface to cause an interrupt
10	Unused		0	
11	PRFlg	R/W	0	Enables a flag from the peripheral bus to cause an interrupt
12	UART0	R/W	0	Enables a flag from UART0 to cause an interrupt
13	UART1	R/W	0	Enables a flag from UART1 to cause an interrupt
14	DMAInterrupt	R/W	0	Enables a DMA interrupt
15, 17, 19, 21, 23	Unused		0	
16	H1F0	R/O	0	Indicates that 21160-1 FLAG0 is generating an interrupt
18	H2F0	R/O	0	Indicates that 21160-2 FLAG0 is generating an interrupt
20	H3F0	R/O	0	Indicates that 21160-3 FLAG0 is generating an interrupt
22	H4F0	R/O	0	Indicates that 21160-4 FLAG0 is generating an interrupt
24	PCFlg	R/O	0	Indicates that PCI flag is generating an interrupt
(Sheet 1 of 2)				

Bit	Name	Type	Reset Value	Description*
25	PMCFlg0	R/O	0	Indicates that PMC+ FLAG0 is generating an interrupt
26	Unused		0	
27	PRFlg	R/O	0	Indicates that Peripheral bus flag is generating an interrupt
28	UART0	R/O	0	Indicates that UART0 flag is generating an interrupt
29	UART1	R/O	0	Indicates that UART1 flag is generating an interrupt
30	DMAInterrupt	R/O	0	Indicates that DMA interrupt is generating an interrupt
31	Unused		0	
(Sheet 2 of 2)				

* All descriptions in this column apply when bits are set to 1.

Table 6–23 *Contents of the PMC+ Interrupt Configuration Register (0x5A) (Continues on next page)*

Bit	Name	Type	Reset Value	Description*
0	H1F0	R/W	0	Enables 21160-1 FLAG0 to cause an interrupt
1, 3, 5, 7	Unused		0	
2	H2F0	R/W	0	Enables 21160-2 FLAG0 to cause an interrupt
4	H3F0	R/W	0	Enables 21160-3 FLAG0 to cause an interrupt
6	H4F0	R/W	0	Enables 21160-4 FLAG0 to cause an interrupt
8	PCFlg	R/W	0	Enables the PCI interface to cause an interrupt
9	PMCFlg0	R/W	0	Enables FLAG0 from the PMC interface to cause an interrupt
10	Unused		0	
11	PRFlg	R/W	0	Enables a flag from the peripheral bus to cause an interrupt
12	UART0	R/W	0	Enables a flag from UART0 to cause an interrupt
13	UART1	R/W	0	Enables a flag from UART1 to cause an interrupt
14	DMAInterrupt	R/W	0	Enables a DMA interrupt
15, 17, 19, 21, 23	Unused		0	
16	H1F0	R/O	0	Indicates that 21160-1 FLAG0 is generating an interrupt
18	H2F0	R/O	0	Indicates that 21160-2 FLAG0 is generating an interrupt
20	H3F0	R/O	0	Indicates that 21160-3 FLAG0 is generating an interrupt
22	H4F0	R/O	0	Indicates that 21160-4 FLAG0 is generating an interrupt
24	PCFlg	R/O	0	Indicates that PCI flag is generating an interrupt
(Sheet 1 of 2)				

Bit	Name	Type	Reset Value	Description*
25	PMCFlg0	R/O	0	Indicates that PMC+ FLAG0 is generating an interrupt
26	Unused		0	
27	PRFlg	R/O	0	Indicates that Peripheral bus flag is generating an interrupt
28	UART0	R/O	0	Indicates that UART0 flag is generating an interrupt
29	UART1	R/O	0	Indicates that UART1 flag is generating an interrupt
30	DMAInterrupt	R/O	0	Indicates that DMA interrupt is generating an interrupt
31	Unused		0	
(Sheet 2 of 2)				

* All descriptions in this column apply when bits are set to 1.

Flag and Interrupt Status Registers

The registers at offsets 0x5E and 0x5F are 16-bit unmasked registers that show the status of all flags and interrupts. The register at 0x5E shows the status of the flags, and 0x5F shows the status of the interrupts. Table 6–24 and Table 6–25 describe the bits in the registers. For additional explanation of these registers, refer to the *SharcFIN ASIC User's Manual*.

Table 6–24 *Contents of the Flag Status Register*

Bit	Name	Type	Description
0	H1F0	R/O	Status of 21160-1 FLAG0
1	H1F1	R/O	Status of 21160-1 FLAG1
2	H2F0	R/O	Status of 21160-2 FLAG0
3	H2F1	R/O	Status of 21160-2 FLAG1
4	H3F0	R/O	Status of 21160-3 FLAG0
5	H3F1	R/O	Status of 21160-3 FLAG1
6	H4F0	R/O	Status of 21160-4 FLAG0
7	H4F1	R/O	Status of 21160-4 FLAG1
8	PCFlg	R/O	Status of PCI flag
9	PMCFlg0	R/O	Status of PMC+ FLAG0
10	Unused		
11	PRFlg	R/O	Status of peripheral bus flag
12	UART0	R/O	Status of UART0 flag
13	UART1	R/O	Status of UART1 flag
14	DMAInterrupt	R/O	Status of DMA interrupt
15	Unused		

Table 6–25 *Contents of the Interrupt Status Register*

Bit	Name	Type	Description
0	H1I0	R/O	Status of 21160-1 IRQ0
1	H1I1	R/O	Status of 21160-1 IRQ1
2	H2I0	R/O	Status of 21160-2 IRQ0
3	H2I1	R/O	Status of 21160-2 IRQ1
4	H3I0	R/O	Status of 21160-3 IRQ0
5	H3I1	R/O	Status of 21160-3 IRQ1
6	H4I0	R/O	Status of 21160-4 IRQ0
7	H4I1	R/O	Status of 21160-4 IRQ1
8	PCInt	R/O	Status of PCI interrupt
9	PMCI0	R/O	Status of PMC+ IRQ0
15:10	Unused		

Appendix A
Debugging Your DSP Programs

This appendix provides information on debugging DSP programs with either a hardware or a software emulator.

A.1 Debugging with a Hardware (In-Circuit) Emulator

This section discusses attaching an in-circuit emulator (ICE) from Analog Devices to the Hammerhead-6U-cPCI board. To attach an ICE to the Hammerhead-6U-cPCI, follow the steps below:

1. Connect the probe on the ICE card to the Hammerhead-6U-cPCI's JTAG connector.
2. Depending on the type of ICE card you are using, either install it in or connect it to your PC.
3. Install the Hammerhead-6U-cPCI in a 6U slot in your CompactPCI chassis.
4. Apply power to the Hammerhead-6U-cPCI.
5. Start the emulator software on the PC.

A.1.1 Overview of the ICE Emulator

The Hammerhead-6U-cPCI is compatible with White Mountain DSP's ICE emulators, which are ISA bus, PCI bus, ethernet, or USB cards that connect to the Hammerhead-6U-cPCI's JTAG connector and either install in or connect to your PC. The emulator provides a controlled environment for observing, debugging, and testing real-time activities in a target hardware environment by connecting directly to the target processor through its JTAG interface.

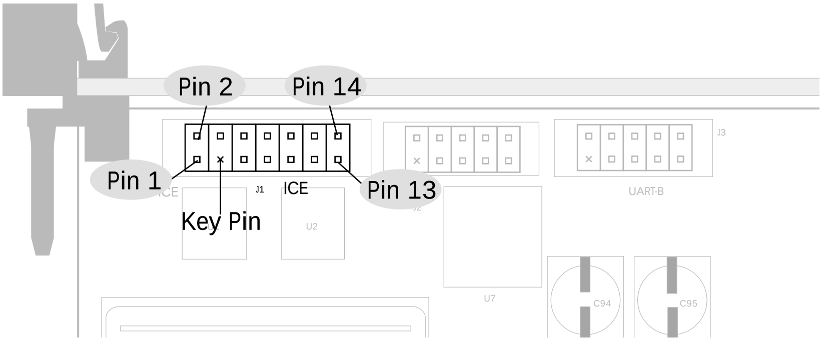
When the ICE is connected to the Hammerhead-6U-cPCI, the Hammerhead-6U-cPCI becomes the target system for the emulator, allowing you to operate it completely from the emulator's user interface. A powerful tool for debugging programs running on the ADSP-21160 processors, the emulator monitors system behavior while running at full speed, and you can use it to download programs, start and stop program execution, set breakpoints, and observe and change the contents of the registers and memory.

A.1.2 Connecting the ICE to the Hammerhead-6U-cPCI

To connect the ICE to the Hammerhead-6U-cPCI, follow the instructions below.

1. Locate the JTAG connector (J1) on the Hammerhead-6U-cPCI (see Figure 3–3).
2. A cable extends from the ICE card to a probe that connects to the JTAG connector on the Hammerhead-6U-cPCI. Connect the ICE probe to the JTAG connector. Figure A–1 shows the location of the pins on the connector.

Figure A–1 *JTAG Connector*



Pin 3 on the JTAG connector is missing (see Figure A–1) to prevent you from installing the emulator incorrectly. One of the sockets in the ICE probe has a plug inserted in place of the pin. Table 3–6 in Chapter 3 shows the connector pinout.

A.1.3 Installing the ICE and Hammerhead-6U-cPCI in a PC

Once you have connected the ICE to the Hammerhead-6U-cPCI, install the Hammerhead-6U-cPCI and install or connect the ICE to the PC. The Hammerhead-6U-cPCI requires a 6U slot in a CompactPCI chassis (see section 2.3.3). How you install the ICE card depends on the form factor of the ICE card you are using. The *ICE Hardware User's Guide* (from Analog Devices) explains how to install the ICE.

A.1.4 Operating the ICE

To start operating the ICE with the Hammerhead-6U-cPCI,

1. Apply power to the Hammerhead-6U-cPCI.

Note

As long as the emulator software is not running, you can safely attach and remove the ICE probe while the Hammerhead-6U-cPCI is running.

2. Start the emulator software on your PC. To download and run programs, follow the instructions in the ICE documentation.

A.2 Debugging with a Software Emulator

BittWare's VisualDSP Target is a fully functional software emulator, which allows you to debug your DSP projects right on your BittWare board without installing a hardware (in-circuit) emulator.

A.2.1 About the VisualDSP Target

If you have installed Analog Devices' VisualDSP++ integrated development environment (IDE), you can use BittWare's VisualDSP Target to debug your DSP programs instead of using a hardware emulator. BittWare's VisualDSP Target is a plug-in to Analog Devices' VisualDSP++ that allows the VisualDSP++ debugger to communicate directly with your BittWare DSP board.

Since the BittWare VisualDSP Target is integrated right into the VisualDSP++ debugger, you can compile and link your code in the VisualDSP++ integrated development environment and immediately debug your code directly on the BittWare board. A full-featured software debugger, the VisualDSP Target allows you to set breakpoints, single-step through your code, view memory, and run code on multiple processors.

A.2.2 Installing the VisualDSP Target

To install the VisualDSP Target, insert the VisualDSP Target CD-ROM into your computer's CD-ROM drive, and follow the installation instructions on the screen. Once you have installed the Target, follow the instructions in the *VisualDSP Target User's Guide* to prepare your DSP program for debugging.

Note

The Hammerhead-6U-cPCI is compatible with versions 2.0 and greater of the BittWare VisualDSP Target.

Appendix B
Setting Up for Standalone Operation

The Hammerhead-6U-cPCI can boot via link ports or from a boot program stored in its Flash memory (see section 5.2), allowing it to operate in standalone mode, free from a host computer. This section lists the steps necessary to prepare your Hammerhead-6U-cPCI to operate in standalone mode.

Note

If you are not planning to operate the Hammerhead-6U-cPCI in standalone mode, follow the instructions in Chapter 2.

1. While in development mode, develop a boot loader and a standalone operating program for the DSPs (see B.1.1).
2. Program the boot Flash (or other external peripheral for link booting) with the boot loader (see B.1.2).
3. Power down and set the boot mode jumpers to “Flash Boot” or “Link Boot” (see “Selecting the Boot Mode” on page 16).
4. Set the Standalone Mode jumpers (see “Setting the Standalone Mode Jumpers” on page 18).
5. Apply power to the Hammerhead-6U-cPCI (see B.4).

B.1 Developing and Loading a Boot Program

B.1.1 Developing the Boot Program

Use the Analog Devices software development tools and BittWare's host interface tools (DSP21k-SF Toolkit) to develop a boot program for the ADSP-21160 processors. The example software included with the Hammerhead-6U-cPCI includes programs that demonstrate how to create a DSP program in the proper form and then load it into Flash memory so that it automatically boots and begins executing after the board is powered on or the DSPs are reset.

B.1.2 Loading a Boot Program into the Flash Memory

Use programs provided with the Hammerhead-6U-cPCI to load the Flash memory with a boot program. The `flash` directory contains utilities that provide easy access to the Flash memory on the Hammerhead-6U-cPCI board. The `flash` directory also contains a test program that uses the utilities.

B.1.3 Loading a Link Boot Program

Using the external link ports, load the boot program onto the DSPs. Analog Devices supplies loading routines (loader kernels) that load an entire program through the selected port. These routines come with the Analog Devices development tools. For more information on loader kernels, refer to the development tools documentation.

B.2 Configuring the Board for Standalone Mode

B.2.1 Selecting the Boot Mode

The Hammerhead-6U-cPCI has four configuration jumpers (JP5 and JP6 for cluster A, and JP8 and JP9 for cluster B) that configure its boot mode. See "Selecting the Boot Mode" on page 16 for instructions on setting the jumpers.

B.2.2 Setting the Standalone Mode Jumpers

The Hammerhead-6U-cPCI has three jumpers for configuring the board to operate in standalone mode:

JP13	Standalone primary PCI voltage
JP14	Standalone mode clock
JP15	Standalone reset

All three jumpers must be "ON" for the board to operate properly in standalone mode. Refer to section 2.2.1 for more detail.

B.3 Booting the Board via Link Port

Connect a cable to link 4 on either of the rear panel I/O connectors (see Table 3–4 and Table 3–5 for pinouts). P3 link 4 provides a link port connection for booting cluster A, and P5 link 4 provides a link port connection for booting cluster B. When booting via link port, DSP 1 will boot and then boot the other three DSPs in its cluster.

B.4 Supplying Power to the Hammerhead-6U-cPCI

The Hammerhead-6U-cPCI requires a +3.3 and +5V power supply for normal operation. When operating with a PMC module, it requires +12V and –12V. The external power connector (J13) supplies +3.3V, +5V, –12V, and +12V to the Hammerhead-6U-cPCI. Section 3.2.9 gives the pinout of the connector and shows where it is located.

To connect an external power source to the Hammerhead-6U-cPCI,

1. Plug a power adapter cable into the Hammerhead-6U-cPCI's external power connector (J13). Be sure to align pin 1 (+12V) on J13 with the +12V pin on the cable.
2. Connect the remaining end of the cable to an external power source, such as a switching standalone power supply or the PC's power supply.
3. Apply power to the system.
4. Reset the Hammerhead-6U-cPCI. Section 2.5 explains in more detail how to reset the board.

Appendix C
Troubleshooting Tips

This section lists the information you should have ready before calling technical support at BittWare. It also provides the phone numbers and e-mail address for technical support.

C.1 Before You Call Technical Support

To allow us to serve you better, please perform the following checks and record any significant results before contacting BittWare for assistance.

- Run DspBad on the board and note the results.
- Run Diag21k on the board; enter **`br`** at the first prompt, **`pc`** at the next, and then initiate memory tests by entering **`mt aa`**.
- If you are getting "file not found" or similar errors, try re-installing the tools and checking your path.
- Try changing the hardware to see if the problem tracks with the board or the PC:
  - If you have access to a different board, please try it.
  - Try the board in a different PC.
  - Try a different operating system.
- Finally, when contacting BittWare please have the results of the tests listed above and the following information ready:
  - Information identifying the hardware and software you purchased (see the BittWare packing list)
  - Which operating system you are using: DOS, Windows 3.1, Windows 95, Windows 95B (OSR2), Windows 98, Windows NT Version 3.51, or Windows NT Version 4.0
  - The release number of your DSP21k-SF Toolkit (enter **`diag21k -v`** at a DOS prompt)
- If you could be at the PC that is experiencing problems when making the call, we will be better able to start diagnosing the problem.

C.2 Contacting Technical Support

To reach technical support at BittWare, Inc., use one of the following methods:

- Phone (8:30 a.m. – 5:30 p.m. EST): (603) 226-0404
- FAX: (603) 226-6667
- E-mail: support@bittware.com

Bittware also maintains the following internet sites:

http://www.bittware.com	Contains product information, technical notes, support files available for download, and answers to frequently asked questions (FAQ).
ftp://ftp.bittware.com	Contains technical notes and support files. Login as “anonymous,” and use your email address for the password.

Appendix D
Glossary of Terms

ADSP-21160 cluster buses

The ADSP-21160 cluster buses are 40 MHz, 64-bit buses that connect the four ADSP-21160 processors in each cluster and a 64–512 MB bank of SDRAM. They are connected to the PCI interface through the SharcFIN ASICs.

DSP21k-SF Toolkit

BittWare’s DSP21k-SF Toolkit is a set of libraries and utilities that enable you to develop DSP applications for the Hammerhead-6U-cPCI more quickly and easily. It contains a host interface library of C-callable functions for PC-based programs, diagnostic utilities, demo programs, and a DSP library that provides standard I/O extensions to DSP programs.

in-circuit emulator (ICE)

The Hammerhead-6U-cPCI is compatible with Analog Devices’ in-circuit emulators (ICE), which are ISA bus, PCI bus, ethernet, or USB cards that connect to the Hammerhead-6U-cPCI’s JTAG connector. The ICE emulators provide a controlled environment for observing, debugging, and testing real-time activities in a target hardware environment by connecting directly to the target processor through its JTAG interface.

IOP registers

The IOP registers are control, status, or data buffer registers of the ADSP-21160’s on-chip I/O processor.

mailbox registers

The mailbox registers are registers in the SharcFIN ASIC that cause an interrupt when written to. The processor that was interrupted can read the registers to find out about the interrupt.

multiprocessor memory space (MMS)

Multiprocessor memory space is the memory of other ADSP-21160 processors in the same cluster. A cluster is up to six ADSP-21160 processors that share a common processor bus. Any ADSP-21160 processor that is connected to the bus shares the MMS.

MS0

MS0 (memory select 0) allows the DSPs to access the Hammerhead-6U-cPCI's SDRAM, which is located on the 64-bit ADSP-21160 cluster bus.

MS1

MS1 (memory select 1) allows the DSPs to access the Hammerhead-6U-cPCI's 1 MB bank of Flash memory, the dual UART, and the peripheral bus.

MS2

MS2 (memory select 2) allows the DSPs to access the SharcFIN ASIC.

MS3

MS3 (memory select 3) is unused on the Hammerhead-6U-cPCI

MSIZE

The MSIZE bits of the ADSP-21160's SYSCON register define the size of the Hammerhead-6U-cPCI's four banked sections of memory, which are accessible to the DSPs via their memory select lines (MS0–MS3).

PCI-to-DSP bridges

BittWare's SharcFIN ASICs function as bridges (PCI-to-DSP) between the PCI interface and the ADSP-21160 DSPs, interfacing the secondary PCI buses, the ADSP-21160 cluster buses, and the peripheral buses to the primary PCI bus.

PCI-to-DSP interface

The PCI-to-DSP interface allows the ADSP-21160 processors to communicate with the PCI bus, and vice-versa. It consists of the SharcFIN ASICs and secondary PCI buses A and B.

PCI-to-PCI bridge

The PCI-to-PCI bridges are chips manufactured by Intel Corporation (21154) that provide bridges between the primary PCI bus and the secondary PCI buses.

PCI-to-PCI interface

The PCI-to-PCI interface consists of the primary PCI bus and PCI-to-PCI bridge chips A–C. Bridge A interfaces secondary PCI bus A to bus C. Bridge B interfaces secondary PCI bus B to bus C. Bridge C interfaces secondary PCI bus C to the primary PCI bus.

peripheral buses

The 25 MHz, 8-bit peripheral buses connect to low-speed peripherals such as the Flash memory, the dual UARTs, and an expansion connector. The purpose of the peripheral buses is to allow additional components to communicate with the ADSP-21160s without affecting the signal quality of the ADSP-21160 cluster bus. A peripheral bus extends from each SharcFIN ASIC.

primary PCI bus

The primary PCI bus is a 66 MHz, 64-bit bus between the host and PCI-to-PCI bridge C.

secondary PCI buses

The Hammerhead-6U-cPCI has three 64-bit, 66 MHz secondary PCI buses. Secondary PCI bus A connects PCI-to-PCI bridge A to SharcFIN A. Secondary PCI bus B connects PCI-to-PCI bridge B to SharcFIN B. Secondary PCI bus C interconnects PCI-to-PCI bridges A and B with PCI-to-PCI bridge C.

SharcFIN ASIC

BittWare's SharcFIN ASIC flexibly interfaces Analog Devices' SHARC DSPs to a wide range of the Hammerhead-6U-cPCI's interfaces, including: 64/66 MHz PCI bus (rev. 2.2 compliant), SDRAM, UART, I^{2}C interface, Flash, and a general-purpose expansion bus (the 8-bit peripheral bus). The SharcFIN also provides a feature-rich set of DMA functions and interrupt options to support very high-speed, real-time data flow with a minimum of processor overhead. The Hammerhead-6U-cPCI features a SharcFIN for each DSP cluster.

SPORT

SHARC synchronous serial port

SYSCON register

The SYSCON register is a register in the ADSP-21160 DSPs that contains the MSIZE bits and is used to select the packing mode for synchronous and asynchronous transfers performed by the host.

VisualDSP++

Analog Devices' VisualDSP++ is an easy-to-use project management environment comprised of an integrated development environment (IDE) and debugger.

VisualDSP Target

BittWare's VisualDSP Target is a plug-in for VisualDSP++ that works with the VisualDSP++ debugger to allow direct communication with the DSPs on the Hammerhead-6U-cPCI.

WORM

Write Once Read Many

Index