Hammerhead-PCI

Quad ADSP-21160 PCI 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-PCI User's Guide

Hardware Revision 1

Printed in the USA
August 22, 2001 Edition UG-HHPC01-02

Contents

Chapter 1 Introduction

Chapter 2 Preparing the Hammerhead-PCI for Operation

Figures

Tables

Chapter 1
Introduction

The Hammerhead-PCI is a PCI-format DSP board combining the power of four ADSP-21160 SHARC® processors from Analog Devices with the speed of a 64-bit, 66 MHz PCI interface. The board supports up to 512 MB of SDRAM, a bank of Flash memory, and two PMC mezzanine sites. The board's PCI interface features BittWare's Sharc®FIN™ ASIC. The Sharc FIN flexibly interfaces the ADSP-21160s to the 64-bit, 66 MHz PCI interface, the SDRAM, the Flash memory, and a general-purpose expansion bus.

This chapter:

- Overviews the basic architecture of the Hammerhead-PCI system
- Gives an overview of each chapter in this user's guide
- Lists documents that provide more information about the Hammerhead-PCI's components and software

1.1 Overview of the Hammerhead-PCI System

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

1.1.1 Hammerhead-PCI Features

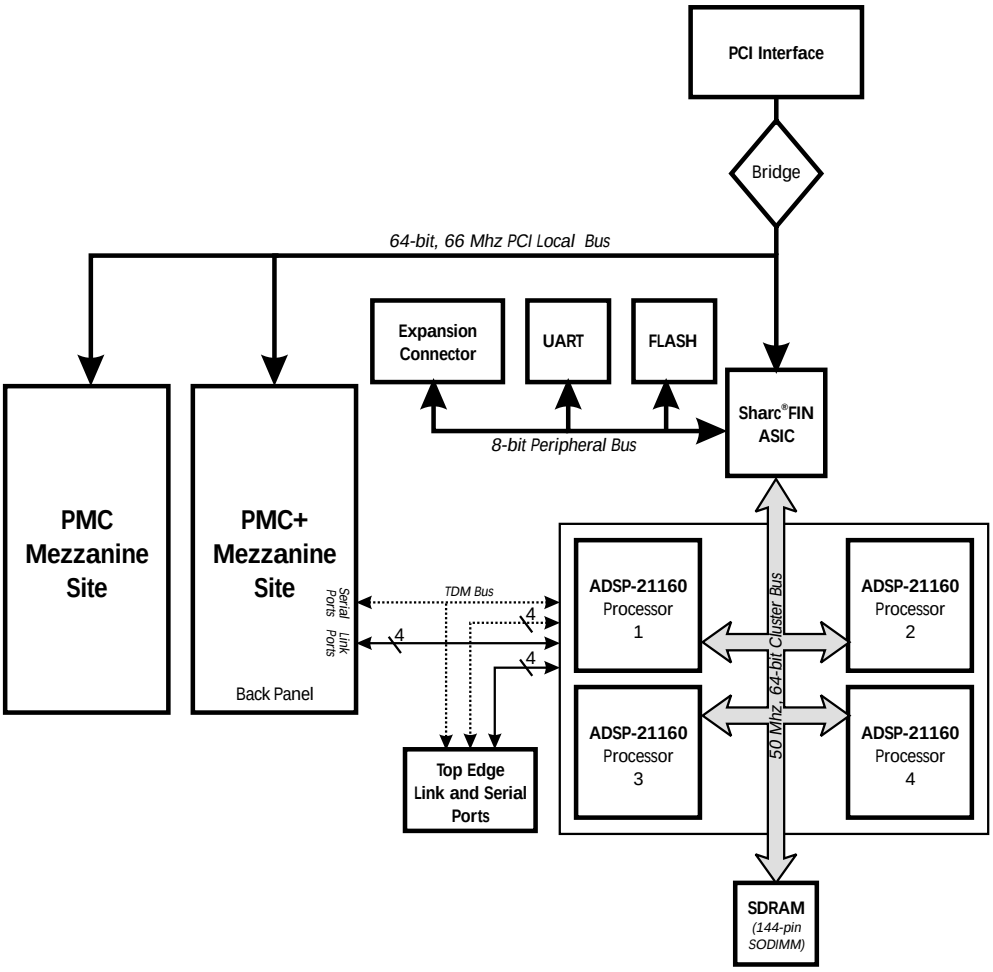
The Hammerhead-PCI features:

- Four 100 MHz ADSP-21160 SHARC processors (2400 MFLOPS)
- 64-bit, 66 MHz PCI interface
- Up to 512 MB SDRAM (standard 144-pin SODIMM)
- Two PMC sites, one with PMC+ extensions for BittWare's PMC+ I/O modules
- Four link ports and one serial TDM bus for integrating PMC+ I/O with on-board SHARCs
- Four 100 MB/s external link ports
- Four 50 Mb/s external serial ports; one 50 Mb/s external serial TDM bus
- BittWare's Sharc FIN ASIC
- RS-232 UART
- 2 MB Flash RAM
- Standalone operation

1.1.2 Hammerhead-PCI System Architecture

This section gives a basic overview of the Hammerhead-PCI system, describing how all of its features work together. Figure 1–1 is a detailed block diagram of the Hammerhead-PCI board and its features.

Figure 1–1 *Block Diagram of the Hammerhead-PCI System*



BittWare's Hammerhead-PCI board features four Analog Devices' ADSP-21160 processors, 64-512 MB of SDRAM, 2 MB of Flash memory, and two PMC mezzanine sites.

Sharc FIN ASIC

The Hammerhead-PCI incorporates BittWare's FIN ASIC for SHARC DSPs. The Sharc FIN flexibly interfaces the ADSP-21160 DSPs to the 64-bit, 66 MHz PCI bus, the SDRAM, the Flash memory, and a peripheral bus. It 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-PCI board is configured with four 100 MHz ADSP-21160 processors. The ADSP-21160 processors are code-compatible with the ADSP-2106x SHARC DSPs, making it easier to integrate existing code. The four ADSP-21160 processors share a common 50 MHz, 64-bit cluster bus, which gives them access to the board's SDRAM, the PCI bus interface, and the other three SHARC DSPs. For additional I/O, each processor also has four flags, three interrupts, six link ports, and two serial ports.

Note

The ADSP-21160 DSPs on Revision 1 of the Hammerhead-PCI run at 80 MHz. Therefore, the ADSP-21160 cluster bus is 40 MHz, the serial ports are 40 Mbits/s, and the link ports are 80 Mbytes/s.

I/O Support

The Hammerhead-PCI offers a variety of user I/O options in addition to the 64-bit, 66 MHz PCI interface. It features a PMC+ site, external serial ports, an RS-232 interface, and external link ports.

The Hammerhead-PCI board is configured with two PMC (PCI Mezzanine Card) sites. One PMC site has back-panel access and allows you to attach a standard PMC module to the board, adding I/O or additional processors and memory. The other PMC site has front panel access and functions as both a standard PMC and a proprietary interface for BittWare's PMC+ I/O modules. It features PMC+ extensions that allow you to attach PMC+ modules for low-latency, high-performance I/O via four 100 MB/s link ports and one 50 Mb/s serial port.

One serial port on each SHARC processor is dedicated as an external serial port. The remaining serial port on each SHARC DSP connects to a TDM serial bus.

The board's dual UART allows the ADSP-21160 processors to communicate with external serial devices via RS-232 ports, facilitating remote debugging, command, and control.

The Hammerhead-PCI features four 100 MB/s external link ports. One link port on each DSP connects to the external connectors, four links per DSP are dedicated for interprocessor communication, and one link per DSP connects to the PMC+ interface.

1.1.3 Hammerhead-PCI Software Architecture

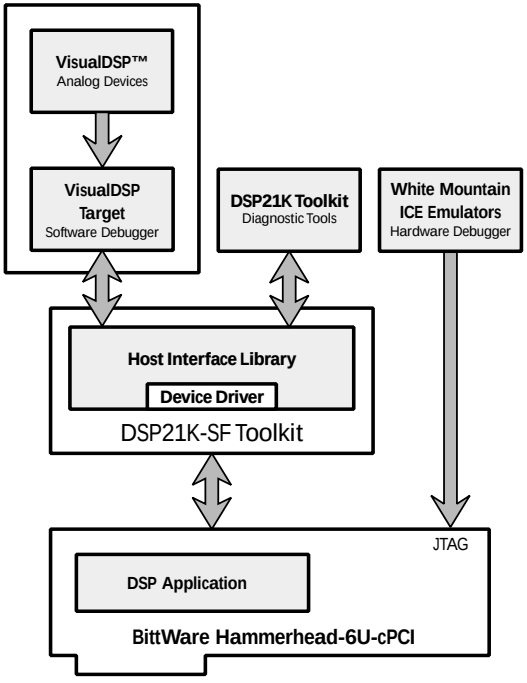
You will need three types of software development tools for the Hammerhead-PCI: code development tools, debugging tools, and host interface tools. Figure 1–2 is a general block diagram of how the software development tools work together with the Hammerhead-PCI.

To begin developing code for the Hammerhead-PCI, 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 White Mountain DSP, 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-PCI 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-PCI and its Software*



This section provides an overview of each chapter's content, and it 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-PCI board, which supports four ADSP-21160 SHARC processors operating at 100 MHz. The purpose of this document is to provide details about the Hammerhead-PCI's major hardware components, to describe how to install and properly operate the Hammerhead-PCI, and to discuss important issues that relate to programming the board.

We assume that 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

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

- Since the Hammerhead-PCI has four processors, we refer to them as 21160-1, 21160-2, etc.
- 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 Two: Preparing the Hammerhead-PCI 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 Three: Overview of the Hardware Components
This chapter shows the location of the Hammerhead-PCI's major components and connectors and briefly discusses their function.

Chapter Four: Hammerhead-PCI Board Architecture
This chapter discusses the board's architecture and the board's serial ports, link ports, flags and interrupts, and bus interfaces.

Chapter Five: Programming Details
This chapter provides programming details for the DSPs, the Sharc FIN ASIC, and the RS-232 UART.

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 discusses common operating problems and provides solutions to those problems. It also 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-PCI or its components. It also lists several third party software development tools that you may find useful.

1.3.1 Documents for Further Reference

ADSP-21160 SHARC Data Sheet – Analog Devices, Inc.

ADSP-21160 SHARC User's Guide – Analog Devices, Inc.

Intel 21154 Chip Data Sheet – Intel

Sharc FIN ASIC User's Guide – BittWare, Inc.

DSP21k-SF Toolkit User's Guide (Version 6.0) – BittWare, Inc.

1.3.2 Software DevelopmentTools

VisualDSP® and BittWare VisualDSP Target

The Hammerhead-PCI 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 White Mountain DSP provide real-time hardware emulation and debugging. White Mountain offers emulators in ISA bus, PCI bus, USB, and Ethernet formats that are compatible with VisualDSP. 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-PCI, 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-PCI 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-PCI 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 (Version 6.0) User's Guide* from BittWare, Inc. contains complete information about the DSP21k-SFToolkit.

Chapter 2
Preparing the Hammerhead-PCI for Operation

This chapter describes how to prepare your board for installation, install the software for the board, install the board, and run diagnostic tests on the board to make sure it is working properly. This chapter does not provide comprehensive instructions for all of the tasks; 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 the host PC.

Warning!

You must install the DSP21k-SF Toolkit and run the BittWare Configuration Manager ***before*** *you install the Hammerhead-PCI board*

To prepare your Hammerhead-PCI board for operation,

1. Unpack the Hammerhead-PCI (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. If you are using I/O module(s) on the board's PMC or PMC+ sites, attach the module(s) to the board (section 2.2.3).
5. Install the VisualDSP software tools (section 2.3).
6. Install BittWare's DSP21k-SF Toolkit (section 2.4).
7. Attach any desired cables to the board (section 2.6).
8. Install the board in a 32- or 64-bit PCI slot in the PC (section 2.5).
9. Run diagnostic tests on the board to ensure it is operating properly (section 2.7.1).
10. Run the example software included with the Hammerhead-PCI (section 2.7.2).

Warning!

The Hammerhead-PCI 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-PCI 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-PCI

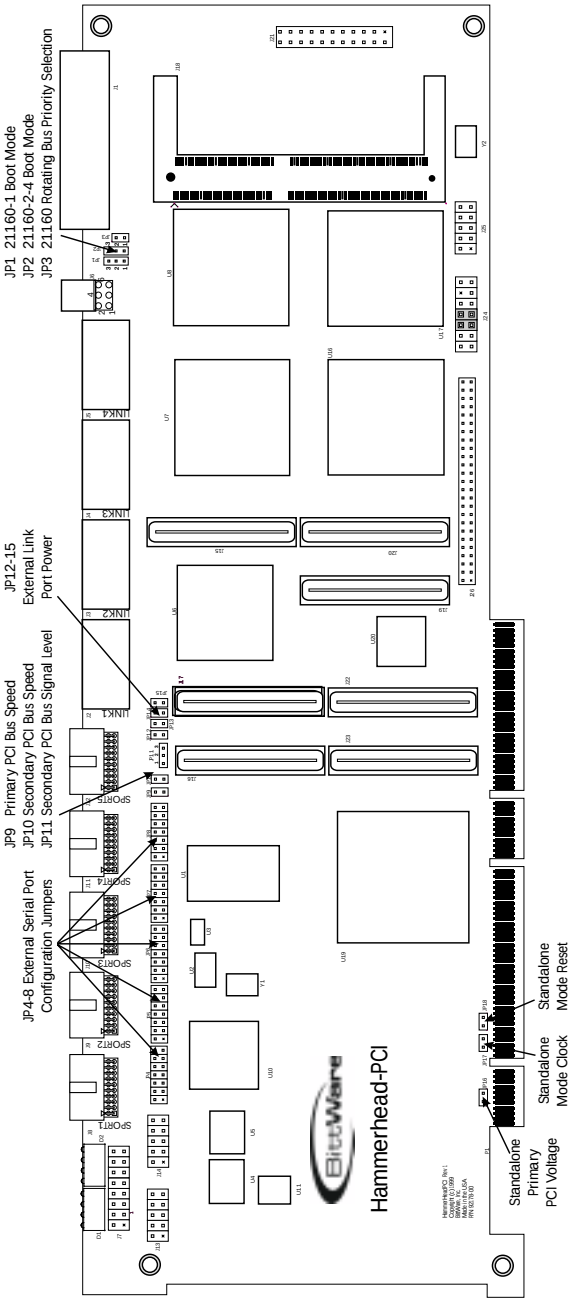
This section explains how to set up the physical features of the board to get it ready for installation. It includes instructions for the following tasks:

- setting the configuration jumpers
- configuring the serial ports
- connecting PMC and PMC+ modules
- setting the board up for standalone operation

2.2.1 Setting the Hammerhead-PCI Configuration Jumpers

The Hammerhead-PCI has eighteen configuration jumpers that allow you to control and enable certain features on the board. Before installing the Hammerhead-PCI in the PC, 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 *Layout of the Hammerhead-PCI Configuration Jumpers*



Setting the Boot Mode for the Processors

JP1 and JP2 configure the boot mode for the ADSP-21160 processors. The processors can boot in three boot modes:

- link booting
- host booting
- Flash booting

JP1 configures the boot mode for 21160-1, and JP2 configures the boot mode for 21160s 2–4. Table 2–4 below shows their settings.

Table 2–1 *Settings for the Boot Mode Selection Jumpers*

Jumper	Jumper Position	Setting	Default
JP1	No Jumper	21160-1 will boot from host computer	✔
	Pins 1–2	21160-1 will boot from on-board Flash	
	Pins 2–3	21160-1 will boot via link booting	
JP2	No Jumper	21160 2–4 will boot from host computer	✔
	Pins 1–2	21160 2–4 will boot from on-board Flash	
	Pins 2–3	21160 2–4 will boot via link booting	

Setting the ADSP-21160 Bus Arbitration

JP3 configures the rotating priority bus arbitration (RPBA) mode for the 64-bit ADSP-21160 cluster bus. The ADSP-21160 bus has 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 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 would have priority, then 21160-2 would have priority, and so on.

Table 2–2 *Selecting the Rotating Priority Bus Mode for the ADSP-21160s (JP3)*

Jumper Position	Setting	Default
IN	Fixed priority scheme	
OUT	Rotating priority scheme	✔

Selecting 33 MHz or 66 MHz Operation for the PCI Bus

JP9 configures the primary PCI bus to run at either 33 MHz or 66 MHz, and JP10 configures the secondary PCI bus. If the primary PCI bus is set to 33 MHz, then the secondary PCI bus must also operate at 33 MHz. However, if the primary PCI bus is set to 66 MHz, the secondary bus can operate at either 33 or 66 MHz. The settings for the jumpers are in Table 2–3.

Table 2–3 *Settings for the PCI Bus Jumpers (JP9 and JP10)*

Jumper	Jumper Position	Setting	Default
JP9 (Primary PCI)	IN	33 MHz	
	OUT	66 MHz	✔
JP10 (Secondary PCI)	IN	33 MHz	
	OUT	66 MHz	✔

Selecting the Signal Level of the Secondary PCI Bus

JP11 is a 3-pin jumper that configures the signal level of the secondary PCI bus to either 3.3 volts or 5 volts. Table 2–4 below gives the jumper's settings.

Warning!

Be sure to set this jumper correctly. If you set this jumper incorrectly, some power supplies could short together and ruin the board.

Table 2–4 *Settings for the Secondary PCI Signal Level Jumper (JP11)*

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 or the board will not function properly.

Setting the Link Port Connector Power Jumpers

JP12–JP15 are link port connector power jumpers. These jumpers gate power to the external link port connectors. Leave these jumpers off unless you have specific requirements for the connectors. Table 2–5 shows their settings.

Table 2–5 *Settings for the Link Port Connector Power Jumper (JP12–15)*

Jumper Position	Setting	Default
IN	5 Volts	
OUT	Unpowered	✔

Setting the Standalone Operation Jumpers

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

JP16 Standalone primary PCI voltage
JP17 Standalone mode clock
JP18 Standalone mode reset

Table 2–6 *Settings for the Standalone Mode Jumpers (JP16, JP17, JP18)*

Jumper	Jumper Position	Setting	Default
JP16 (PCI voltage)	IN	Standalone	
	OUT	Normal Operation	✔
JP17 (Clock)	IN	Standalone	
	OUT	Normal Operation	✔
JP18 (Reset)	IN	Standalone	
	OUT	Normal Operation	✔

2.2.2 Configuring the External Serial Ports

The Hammerhead-PCI has five 20-pin right-angle IDC external serial port connectors to provide a communication route between the ADSP-21160s and synchronous serial devices. Each connector has a corresponding jumper block that you can use to configure the connector (see Table 2–7). Figure 2–1 shows where the jumper blocks are located on the Hammerhead-PCI board, and Figure 3–2 shows where each connector is located.

Table 2–7 *External Serial Port and Jumper Block Connections and Usage*

Serial Port	Jumper Block	Connections
J8 (SPORT1)	JP4	Connected to 21160-1 as a standard serial port
J9 (SPORT2)	JP5	Connected to 21160-2 as a standard serial port
J10 (SPORT3)	JP6	Connected to 21160-3 as a standard serial port
J11(SPORT4)	JP7	Connected to 21160-4 as a standard serial port
J12 (SPORT5)	JP8	Connects to all four processors as a TDM serial port

Setting the Standard Serial Port Jumper Blocks

Table 2–8 gives the pinout of the jumper blocks, and Table 2–10 shows the settings for the jumper blocks. The sections following the tables gives examples of how to put the jumper settings to use in various serial port configurations.

Figure 2–2 *Location of Pin 1 on the Serial Port Jumper Blocks*

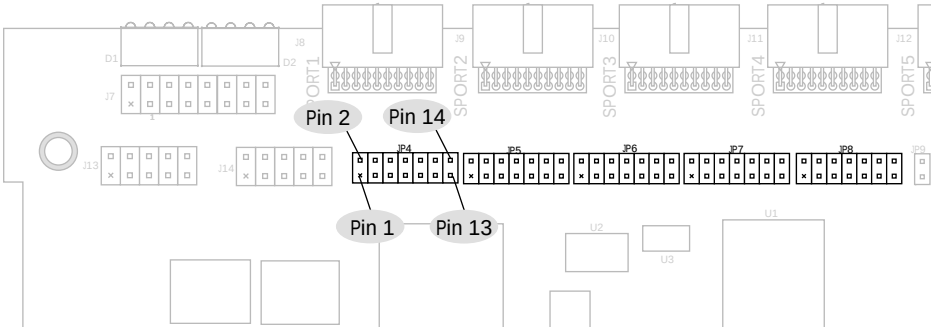


Table 2–8 *TDM Serial Port Jumper Block Pinout (J8)*

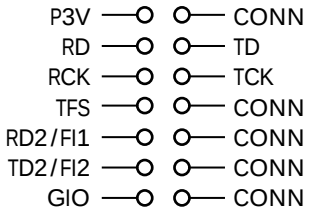


Table 2–9 *Standard Serial Port Jumper Block Pinout (J9–J12)*

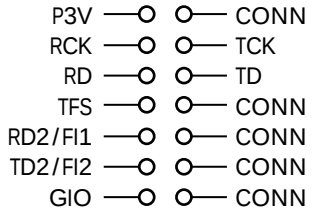


Table 2–10 *Standard Jumper Block Settings*

Jumper	Settings
P3V	IN: If using active external adapter that needs power OUT: Otherwise
TCK/RCK	IN: TDM mode OUT: Standard mode
TD/RD	IN: TDM 1-wire OUT: TDM 2-wire, standard mode
TFS	IN: Standard with null-modem cable OUT: Otherwise
FI1	Depends on implementation
FI2	Depends on implementation
GI0	Depends on implementation

2.2.3 Connecting a PMC or PMC+ Module to the Board

The Hammerhead-PCI board features two PMC interfaces. One PMC interface allows you to attach standard PMC (PCI Mezzanine Card) modules to the board. The second PMC interface allows you to attach standard PMC modules or BittWare's PMC+ I/O modules. The second PMC interface features 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 P4 of the PMC connectors (see section 3.2.7) for our PMC+ extensions. If you are not mounting a BittWare PMC+ card on the Hammerhead-PCI board, the PMC board may have incompatibilities with the PMC+ (P4) connector. Call BittWare technical support for assistance.

To attach a PMC module to the Hammerhead-PCI,

1. Plug the PMC module onto either of the Hammerhead-PCI's PMC interfaces.
2. Secure the PMC module to the mounting holes.
3. Check the required PCI signalling and width.
4. Set JP10 and JP11 appropriately (see section 2.2.1).

2.3 Installing the Analog Devices Software Tools

The Hammerhead-PCI 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' 4.0 SHARC C compiler, C runtime library, assembler, linker, loader, simulator, and splitter.

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-PCI. This section describes where to find installation instructions for the VisualDSP IDE and the BittWare VisualDSP Target.

2.3.1 Installing the VisualDSP® IDE

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

2.3.2 Installing the BittWare VisualDSP Target

If you will be using the VisualDSP Target debugger with the Hammerhead-PCI, you will need to install BittWare's VisualDSP Target after installing the VisualDSP IDE. The VisualDSP Target allows the VisualDSP debugger to communicate directly with the ADSP-21160 processors on the Hammerhead-PCI. The *VisualDSP Target User's Guide* gives detailed installation instructions.

2.4 Installing the BittWare DSP21k-SF Toolkit

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

2.4.1 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-PCI 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.

2.4.2 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.

2.4.3 Verifying Board Configuration

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-PCI board. The *DSP21k-SF Toolkit User's Guide* explains how to run the BittWare Configuration Manager.

2.5 Installing the Board in a PC

The Hammerhead-PCI plugs into a full-length 32- or 64-bit slot in your PC.

Note

Before installing the Hammerhead-PCI in your PC, be sure to follow the instructions in sections 2.1–2.4.

Warning!

The Hammerhead-PCI requires 3.3V from the PCI backplane. If the PCI backplane does not supply 3.3V, the board will not function properly. If in question, use a multi-meter on an external power connector and check the 3.3V signal.

1. Open the computer chassis and access the expansion slots. (Refer to your computer's documentation for instructions.)
2. Select a 32- or 64-bit slot and remove the slot cover bracket.
3. Plug the Hammerhead-PCI into the slot you selected.
4. Secure the Hammerhead-PCI's mounting bracket to the chassis with the screw that you removed from the slot cover bracket (see step 2 above).
5. Reassemble the computer chassis.
6. Power up the system.

2.6 Connecting Cables to the Board

After you have set the configuration jumpers and installed the board and its software, the next step is to attach any cables to the board to connect it to external devices.

2.6.1 Connecting Link Port Cables

The Hammerhead-PCI board has four external link port connectors (J2–J5), which allow the processors on the Hammerhead-PCI board to communicate directly with processors on other boards. Section 3.2.2 shows the location and pinout of the connectors. Table 2–11 below gives the part numbers of the link port cables that are compatible with the Hammerhead-PCI's link port connectors.

Table 2–11 *Link Port Cables Compatible with the Hammerhead-PCI Link Ports*

Location	Type	Manufacturer	Part Number
Hammerhead-PCI (J2–J5)	Coax Ribbon, 12" or 36"	BittWare	HH-LP-12 (12") HH-LP-36 (36")

2.6.2 Connecting Serial Port Cables

The Hammerhead-PCI has five external serial port connectors (J8–J12), which provide a communication route between the ADSP-21160 processors and other synchronous serial devices. Section 3.2.5 shows the location and pinout of the connectors. Table 2–12 below gives the part numbers of the serial port cables that are compatible with the Hammerhead-PCI's serial ports.

Table 2–12 *Serial Port Cable Compatible with the Hammerhead-PCI Serial Ports*

Location	Type	Manufacturer	Part Number
Hammerhead-PCI (J8–J12)	Coax Ribbon, TDM or Standard	BittWare	HH-SP-TDM-04 HH-SP-STD-04

2.6.3 Connecting RS-232 Cables

To connect the Hammerhead-PCI to your PC via the RS-232 interface, attach a ribbon cable connected to a mass-terminated DB-9 connector. The cable provides a straight-through connection from the Hammerhead-PCI's dual UART to the PC. Since the connector's pinout (see Table 3–8) 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-PCI 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 (J13 or J14). 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 3–8 shows where pin 1 is located).
3. Making sure that the PC's power is off, connect the serial interface cable to the PC.

2.6.4 Connecting an External Power Supply

The Hammerhead-PCI 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 (J1) supplies +3.3V, +5V, –12V, and +12V to the Hammerhead-PCI. Figure 3–3 gives the pinout of the connector and shows where it is located.

To connect an external power source to the Hammerhead-PCI,

1. Plug a power adapter cable into the Hammerhead-PCI's external power connector (J1). Be sure to align pin 1 (+12V) on the cable with the +12V pin on J22.
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-PCI. Section 2.8 explains in more detail how to reset the board.

2.7 Testing the Board to Make Sure it is Operating Properly

This section discusses running diagnostic tests on the board after installation to make sure it is operating properly. It runs you through a Diag21k example and discusses the contents of the example files included with the Hammerhead-PCI.

2.7.1 Testing the Board with the DSP21k-SF Toolkit 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. Diag21k 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 Installation Guide* explains DspBad commands in more detail.

Testing the Board with Diag21k

The example below shows you basic Diag21k commands to test the Hammerhead-PCI's memory and load and run a DSP program. Be sure to

1. The processor number is the *device number * 10 + id number*. See Chapter 3 of the *DSP21k-SF User's Manual* for further explanation.

follow the example steps below in the order they appear. The *DSP21k-SF Toolkit User's Manual* describes the Diag21k commands in more detail.

Step 1: Start Diag21k

1. 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.

2. 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 Sharc FIN boards under Windows 95/98 and
Windows NT/2000. Release 6.00 [ DSP21K-SF Beta, Jun 28 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)
```

Step 2: Display Board Information

3. Use the board information command to display information about the Hammerhead-PCI'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: 0x08000000  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*
```

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-PCI's Memory

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

4. 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
   ```

5. 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
   ```

6. 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-PCI board. Next, load a program and execute it.

7. The `dsp21k32\etc` directory contains an example program that calculates the first twenty prime numbers. The source code is in the `examples\Hammerhead-PCI\primes` directory. Load the pre-compiled executable file with the file load (**`fl`**) command.

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

        "\dsp21ksf\etc\prm21160.dxe" loaded
```

8. 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
```

9. 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

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)
```

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>
```

2.7.2 Testing the Board with the Hammerhead-PCI Example Files

The example software provided with the Hammerhead-PCI 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-PCI CD-ROM.

2.8 Resetting the Board

You can use three methods to reset the Hammerhead-PCI.

- Enable the Hammerhead-PCI's watchdog timer
- Send a reset pulse to the Hammerhead-PCI via the external reset connector
- Reset the Hammerhead-PCI via the host interface

2.8.1 Resetting the Board with the Watchdog Timer

The Hammerhead-PCI's watchdog timer helps to ensure that the Hammerhead-PCI is operating properly. It is 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 Sharc FIN ASIC, enables and disables the watchdog timer. 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 Sharc FIN 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 FLAG0 from a configured processor toggles from 0 to 1 or from 1 to 0. The FLAG0 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 5.3.4). 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 FLAG0 within the given time frame. The Watchdog Configuration register allows you to

select the watchdog's time-out time (see "Enabling the Watchdog and Setting its Time-out" on page 92). If the watchdog timer elapses, it will generate a system reset and the normal boot process will begin.

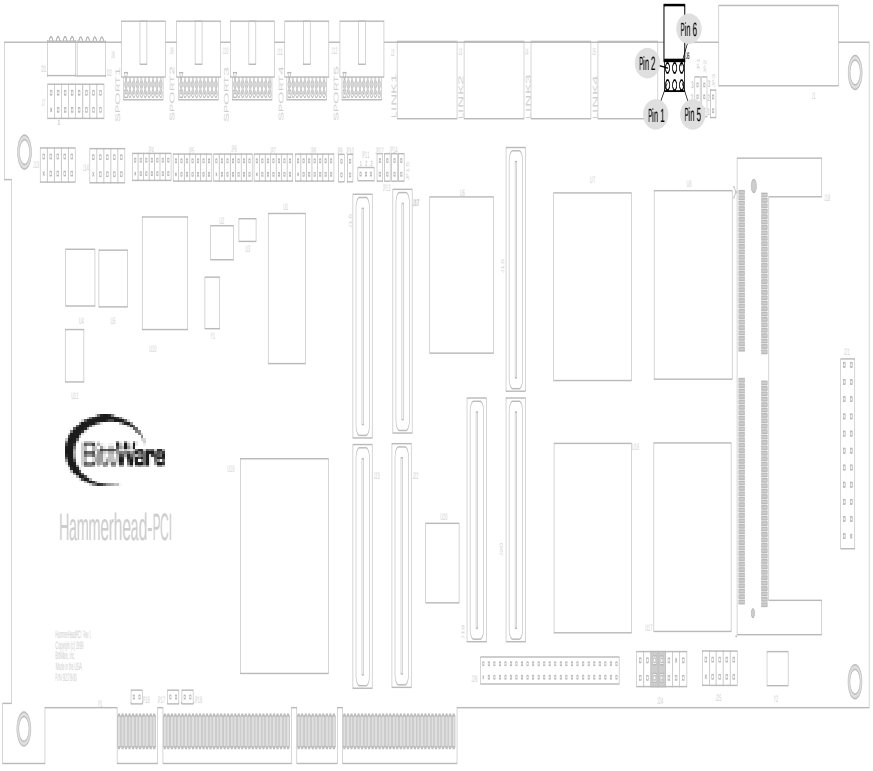
2.8.2 Resetting the Board with the External Reset Connector

The external reset connector (J6) allows the Hammerhead-PCI board to reset or be reset by other system boards. The connector supports an input reset line to allow the Hammerhead-PCI to receive reset signals from other boards; it also supports an output reset line to allow the Hammerhead-PCI to reset other boards.

To reset the Hammerhead-PCI with the external reset connector,

1. Connect a cable (see Figure 2–3 below) from the external reset connector (J6) on the Hammerhead-PCI board to another system board.
2. Any reset that occurs on the Hammerhead-PCI reset source causes a reset on all Hammerhead-PCI reset targets.

Figure 2–3 *Cable Details for the Hammerhead-PCI's External Reset Connector*



2.8.3 Resetting the Board from the Host Interface

A register bit in the Sharc FIN ASIC allows the board to reset from the host PC. When the register is written, all components on the board will be reset.

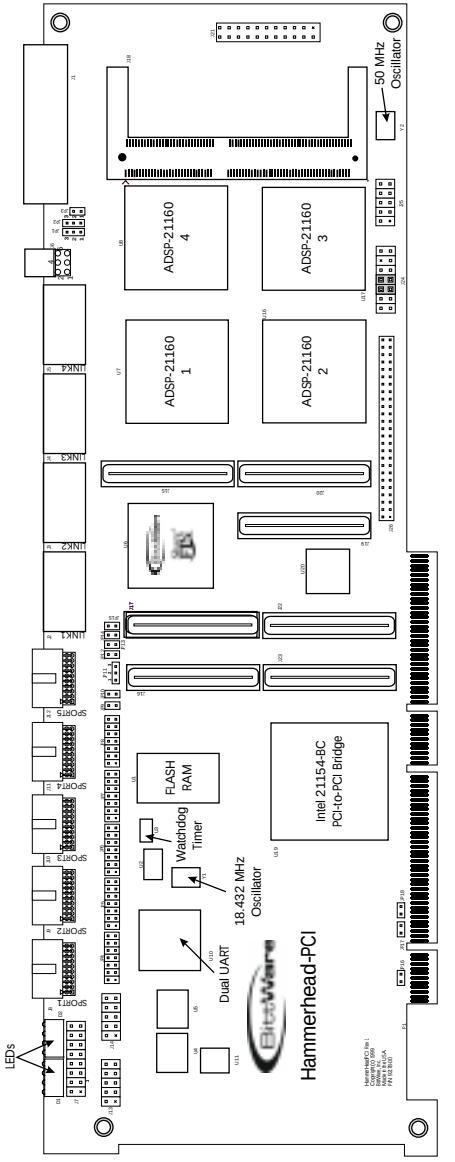
Chapter 3
Overview of the Hardware Components

This chapter shows where the Hammerhead-PCI'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:

- ADSP-21160 SHARC processors
- FLASH RAM
- SDRAM
- Sharc FIN ASIC
- On-board oscillators
- Dual UART
- PCI-to-PCI bridge chip
- LEDs
- Watchdog timer
- PMC and PMC+ interfaces
- JTAG header
- External power connector
- External link ports
- External serial ports
- External reset connector
- RS-232 ports
- Peripheral bus expansion connectors
- Flag test points

Figure 3–1 highlights the major components on the Hammerhead-PCI. The sections that follow describe the features highlighted in the diagram.

Figure 3–1 *Location of the Hammerhead-PCI's Major Components*



3.1.1 ADSP-21160 SHARC Processors

The Hammerhead-PCI features one to four ADSP-21160 SHARC processors from Analog Devices, Inc. The Hammerhead-PCI's processors have a total of 2400 MFLOPs of processing power and operate at 100 MHz. Each processor supports two I^2S 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.

3.1.2 Memory

FLASH RAM

A 2 MB bank of Flash memory stores boot programs that the processor can load, enabling the Hammerhead-PCI to boot without a host computer (see Appendix B). The ADSP-21160s can also read, write, and erase the Flash, which allows them to use the Flash as non-volatile storage space.

SDRAM

An optional bank of up to 512 MB of SDRAM is available to the ADSP-21160s for banked external memory.

3.1.3 Sharc FIN ASIC

BittWare's Sharc FIN ASIC flexibly interfaces Analog Devices' SHARC DSPs to a wide range of the Hammerhead-PCI'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 Sharc FIN 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 Sharc FIN's features:

- 64-bit, 66 MHz PCI rev. 2.2 compliant interface (528 MB/s burst, 400 MB/s sustained)
- Connected to 64-bit, 50 MHz ADSP-21160 cluster bus
- Connected to the Hammerhead-PCI's peripheral bus
  - 8 bits wide @ 25 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: 16 inputs, 11 outputs (one of each dedicated to PCI)
- SDRAM controller on SHARC bus; supports up to 512 MB
- Standard UART and I^2C

3.1.4 PCI Interface

The Hammerhead-PCI features a rev. 2.1 compliant 66 MHz, 64-bit, bus-mastering-capable PCI bus interface. This interface gives host computers direct access to the ADSP-21160 processors, the external SDRAM, the FLASH RAM, the dual UART, and the expansion connector. The PCI interface also gives access to PMC and PMC+ modules on the PMC and PMC+ sites.

The PCI bus can operate at either 64 bits, 66 MHz or 32 bits, 33 MHz. You can configure the speed of both the primary and secondary PCI buses with jumper settings (see section 2.2.1).

Note

If the primary PCI bus is set to 33 MHz, then the secondary PCI bus must also operate at 33 MHz. However, if the primary PCI bus is set to 66 MHz, the secondary bus can operate at either 33 or 66 MHz.

3.1.5 On-Board Oscillator

SHARC Oscillator

A 50 MHz system oscillator chip (Y1) provides the 1× clock for the four on-board SHARC processors. Figure 3–1 shows its location.

UART Oscillator

An 18.432 MHz oscillator chip (Y2) provides the clock for the UART. Figure 3–1 shows its location.

3.1.6 Dual UART

The dual UART interfaces serial data from the RS-232 ports to the 64-bit ADSP-21160 cluster bus. The dual UART acts as two independent 16550 standard UARTs. Section 3.2.6 describes the RS-232 ports.

3.1.7 PCI-to-PCI Bridge Chip

The PCI-to-PCI bridge chip (Intel 21154-BC from Intel Corporation) provides the bridge between the primary and secondary PCI bus.

3.1.8 LEDs

The Hammerhead-PCI has eight user LEDs, which you can use to indicate certain conditions in the software or to provide feedback. Two LEDs are connected to each ADSP-21160. Section 4.4.1 shows the LEDs' connections to the ADSP-21160 flags.

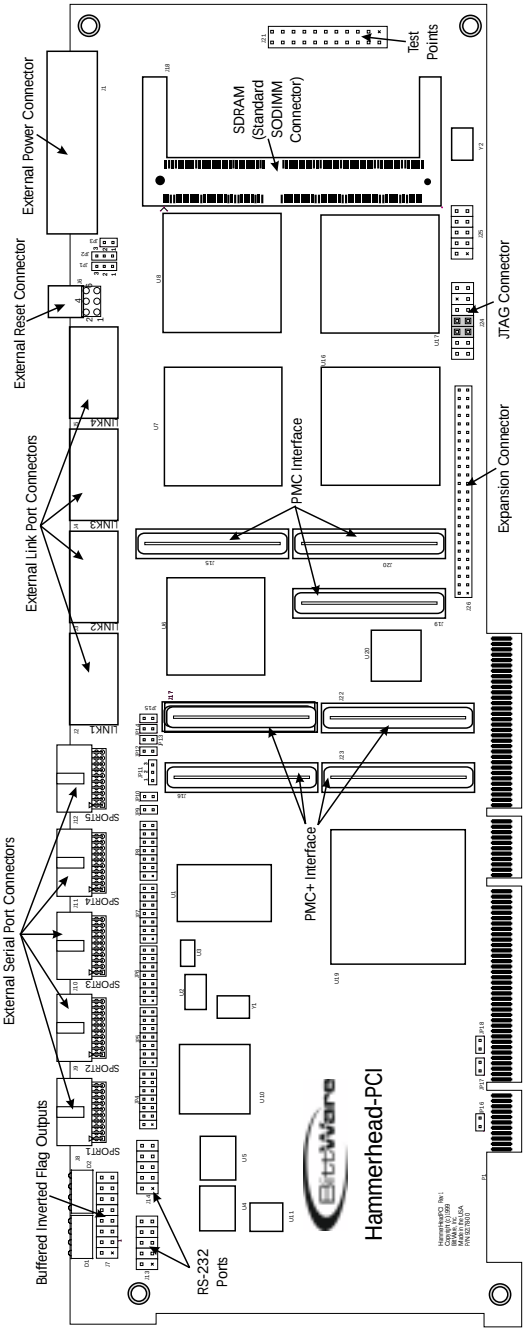
3.2 Layout and Function of the External Connectors

Table 3–1 gives an overview of the external connectors, and Figure 3–2 shows where the Hammerhead-PCI's external connectors are located.

Table 3–1 *Overview of the External Connectors*

Connector	Ref Des	Type	Description
External Power	J1	8-pin	Connection for +3.3V, +5V, and ±12V external power supply
External Link Ports	J2–J5	26-pin	100 MB/s communication route between external devices and ADSP-21160s
External Reset	J6	6-pin	Connection for a panel-mounted reset switch
Buffered Inverted Flag Outputs	J7	16-pin	Access to FLAG2 and FLAG3 on each processor
External Serial Ports	J8–J12	20-pin	Communication route between ADSP-21160s and synchronous serial devices
RS-232 Ports	J13, J14	10-pin	External RS-232 serial interface to ADSP-21160s via dual UART
PMC Interface	J15, J19, J20	64-pin	Connection for standard PMC module
PMC+ Interface	J16, J17, J22, J23	64-pin	Connection for BittWare PMC+ I/O module or for standard PMC module
SDRAM (SODIMM)	J18	144-pin	Connection for standard 144-pin SODIMM SDRAM modules
Test Points	J21	22-pin	Manufacturer use only
JTAG	J24	14-pin	Connection for ICE in-circuit emulator
JTAG Boundary Scan	J25	10-pin	Manufacturer use only
Expansion Connector	J26	50-pin	Connection for adding expansion module onto 8-bit peripheral bus

Figure 3–2 *Location of the External Connectors*



3.2.1 External Power Connector

The Hammerhead-PCI has an external power connector (J1) 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-PCI. Figure 3–3 shows the location of pin 1 on the external power connector (J1), and Table 3–2 gives the connector pinout. Section 2.6.4 provides more information on the connector.

Figure 3–3 *Location of Pin 1 on the External Power Connector*

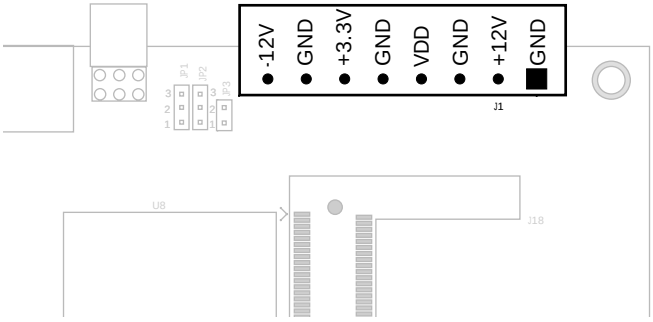


Table 3–2 *External Power Connector Pinout (J1)*

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

3.2.2 External Link Ports

The Hammerhead-PCI has four 26-pin external link port connectors: J2–J5. The external link ports allow you to connect the Hammerhead-PCI directly to other boards. Figure 3–4 shows the location of the external link port connector pins, and Table 3–3 gives the connector pinout.

Figure 3–4 *Location of the External Link Port Connector Pins*

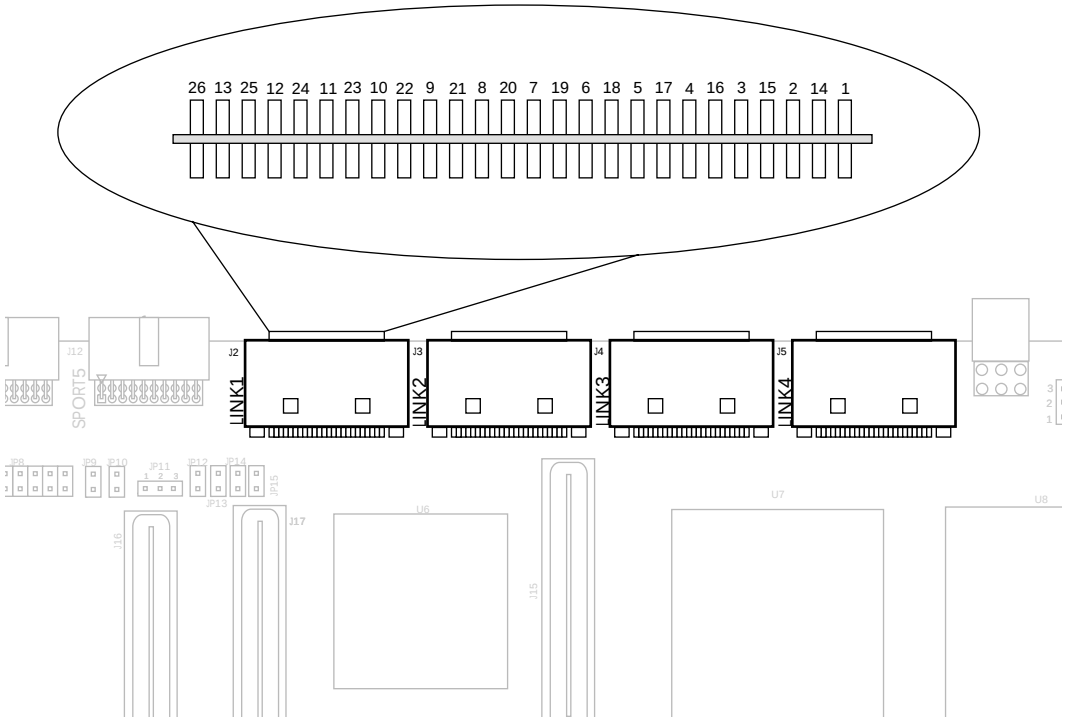


Table 3–3 *Link Port Connector Pinout (J2–J5)*

J2		J3		J4		J5	
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
1	NC	1	NC	1	NC	1	NC
2	EA_L1CLK	2	EA_L2CLK	2	EA_L3CLK	2	EA_L4CLK
3	EA_L1ACK	3	EA_L2ACK	3	EA_L3ACK	3	EA_L4ACK
4	EA_L1DAT0	4	EA_L2DAT0	4	EA_L3DAT0	4	EA_L4DAT0
5	EA_L1DAT1	5	EA_L2DAT1	5	EA_L3DAT1	5	EA_L4DAT1
6	EA_L1DAT2	6	EA_L2DAT2	6	EA_L3DAT2	6	EA_L4DAT2
7	EA_L1DAT3	7	EA_L2DAT3	7	EA_L3DAT3	7	EA_L4DAT3
8	EA_L1DAT4	8	EA_L2DAT4	8	EA_L3DAT4	8	EA_L4DAT4
9	EA_L1DAT5	9	EA_L2DAT5	9	EA_L3DAT5	9	EA_L4DAT5
10	EA_L1DAT6	10	EA_L2DAT6	10	EA_L3DAT6	10	EA_L4DAT6
11	EA_L1DAT7	11	EA_L2DAT7	11	EA_L3DAT7	11	EA_L4DAT7
12	NC	12	NC	12	NC	12	NC
13	NC	13	NC	13	NC	13	NC
14	GND	14	GND	14	GND	14	GND
15	GND	15	GND	15	GND	15	GND
16	GND	16	GND	16	GND	16	GND
17	GND	17	GND	17	GND	17	GND
18	GND	18	GND	18	GND	18	GND
19	GND	19	GND	19	GND	19	GND
20	GND	20	GND	20	GND	20	GND
21	GND	21	GND	21	GND	21	GND
22	GND	22	GND	22	GND	22	GND
23	GND	23	GND	23	GND	23	GND
24	GND	24	GND	24	GND	24	GND
25	GND	25	GND	25	GND	25	GND
26	JP12	26	JP13	26	JP14	26	JP15

3.2.3 External Reset Connector

The Hammerhead-PCI has an external reset connector (J6) to allow one Hammerhead-PCI board to reset other Hammerhead-PCI boards in the same system. The external reset connector is a 6-pin connector.

The connector supports an output reset line, which allows the board to reset other boards. It 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. 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.

Section 2.8 explains the Hammerhead's reset events. Figure 3–5 shows where pin 1 is located, and Table 3–4 gives the connector pinout.

Figure 3–5 *Location of Pin 1 on the External Reset Connector*

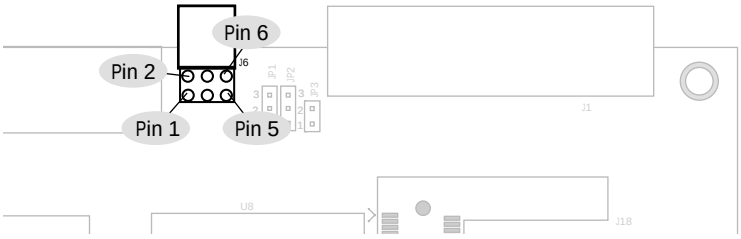


Table 3–4 *External Reset Connector Pinout (J6)*

Pin	Signal	Pin	Signal
1	NC	2	HA_EXTGRPRSTIN
3	NC	4	GND
5	GND	6	HA_EXTGRPRSTOUT

3.2.4 Buffered Inverted Flag Outputs

The buffered inverted flag output connector (J7) allows access to the FLAG2 and FLAG3 signals on each ADSP-21160 DSP. The ADSP-21160s' flag signals are routed through an inverting buffer and are outputs only. Figure 3–6 shows where pin 1 is located on the connector, and Table 3–5 gives the connector pinout.

Figure 3–6 *Location of Pin 1 on the Buffered Inverted Flag Output*

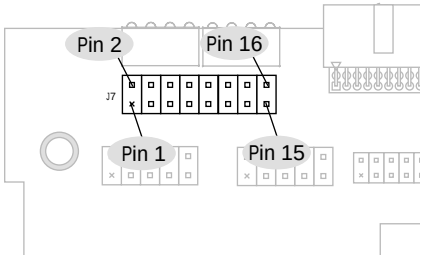


Table 3–5 *Buffered Inverted Flag Output Pinout (J7)*

Pin	Signal	Pin	Signal
1	HA1_F2	2	GND
3	HA1_F3	4	GND
5	HA2_F2	6	GND
7	HA2_F3	8	GND
9	HA3_F2	10	GND
11	HA3_F3	12	GND
13	HA4_F2	14	GND
15	HA4_F3	16	GND

3.2.5 External Serial Ports

Five 20-pin, 50mm, right-angle IDC external serial port connectors (J8–J12) provide a communication route between the ADSP-21160 DSPs and synchronous serial devices at rates up to 50 Mbits/sec. The serial ports on the Hammerhead-PCI conform to BittWare's "*Universal Serial Port Specification*" (available from BittWare upon request). Each connector has a 14-pin jumper block that configures it (see section 2.2.2 and Table 3–6 below).

External Serial Port Connections and Usage

Table 3–6 below explains how each serial port is connected to the DSPs, which jumper block configures it, and whether it is connected as a standard or TDM serial port. Section 2.2.2 explains the jumper blocks in detail.

Table 3–6 *Serial Port Connections and Usage*

Serial Port	Jumper Block	Connection
J8	JP4	Standard connection to 21160-1
J9	JP5	Standard connection to 21160-2
J10	JP6	Standard connection to 21160-3
J11	JP7	Standard connection to 21160-4
J12	JP8	TDM connection to all four 21160 DSPs

Serial Port Connector Pinout

The Hammerhead-PCI's five external serial ports are labeled as J8–J12. Figure 3–7 shows the location of pin 1 on the connectors, and Table 3–7 gives the pinout for the connectors. The signals come directly from the ADSP-21160 pins they are associated with, and they exhibit the same timing characteristics described in the *ADSP-21160 User's Manual* (Analog Devices, Inc.). The signals are series terminated with 82Ω.

Figure 3–7 *Location of Pin 1 on Serial Port Connectors*

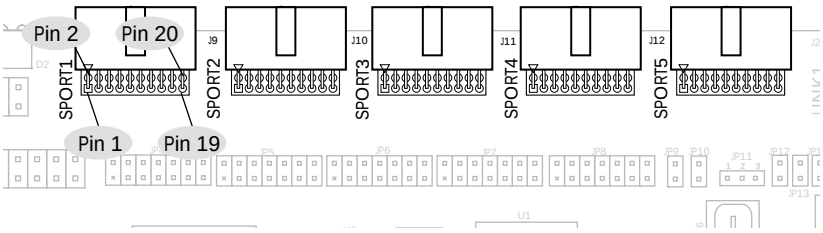


Table 3–7 *Serial Port Connector Pinout (J8–J12)*

Pin	Signal	Pin	Signal
1	+3.3V	2	GND
3	RCLK	4	GND
5	RFS	6	GND
7	RD	8	GND
9	RD2/FI1	10	GND
11	GND	12	TD2/FI2
13	GND	14	TD
15	GND	16	TFS
17	GND	18	TCLK
19	GND	20	GIO

3.2.6 RS-232 Connectors

The RS-232 ports (J13, J14) connect with the Hammerhead-PCI's dual UART to transmit data to the SHARC processor bus. Figure 3–8 shows where pin 1 is located on the RS-232 connectors, and Table 3–8 gives the pinout for the connectors.

Figure 3–8 *Location of Pin 1 on the RS-232 Connectors*

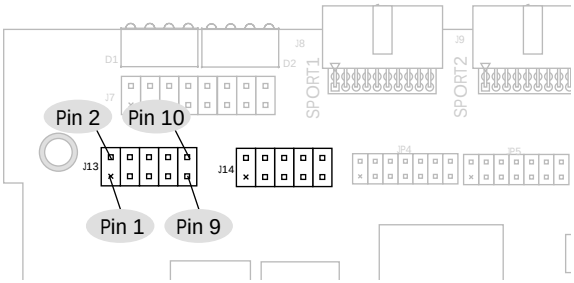


Table 3–8 *RS-232 Connector Pinout (J13, J14)*

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

3.2.7 PMC Interfaces

The Hammerhead-PCI features two PMC sites: a standard PMC interface and a PMC+ interface. The PMC interface allows you to connect a standard PMC module to the board. The PMC+ interface allows you to attach a standard PMC module or one of BittWare's PMC+ I/O modules.

Standard PMC Site

The standard PMC site consists of three 64-pin connectors that connect standard PMC modules directly to the 64-bit, 66 MHz PCI interface. Table 3–9 gives the connector pinout.

Table 3–9 *PMC Interface Pinout*

J20 (PMC-1)				J19 (PMC-2)			
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
1	TCK	2	N12V	65	P12V	66	$\overline{\text{TRST}}$
3	GND	4	$\overline{\text{INTA}}$	67	TMS	68	TDO
5	$\overline{\text{INTB}}$	6	$\overline{\text{INTC}}$	69	TDI	70	GND
7	$\overline{\text{BUSMODE1}}$	8	VDD	71	GND	72	$\overline{\text{PCI-RSVD}}$
9	$\overline{\text{INTD}}$	10	$\overline{\text{PCI-RSVD}}$	73	$\overline{\text{PCI-RSVD}}$	74	$\overline{\text{PCI-RSVD}}$
11	GND	12	$\overline{\text{PCI-RSVD}}$	75	$\overline{\text{BUSMODE2}}$	76	P33V
13	CLK	14	GND	77	$\overline{\text{RST}}$	78	$\overline{\text{BUSMODE3}}$
15	GND	16	$\overline{\text{GNT}}$	79	P33V	80	$\overline{\text{BUSMODE4}}$
17	$\overline{\text{REQ}}$	18	VDD	81	$\overline{\text{PCI-RSVD}}$	82	GND
19	V(I/O)	20	AD[31]	83	AD[30]	84	AD[29]
21	AD[28]	22	AD[27]	85	GND	86	AD[26]
23	AD[25]	24	GND	87	AD[24]	88	P33V
25	GND	26	$\overline{\text{C/BE[03]}}$	89	IDSEL	90	AD[23]
27	AD[22]	28	AD[21]	91	P33V	92	AD[20]
29	AD[19]	30	VDD	93	AD[18]	94	GND
31	V(I/O)	32	AD[17]	95	AD[16]	96	C/BE[2]
33	$\overline{\text{FRAME}}$	34	GND	97	GND	98	PMC-RSVD
35	GND	36	$\overline{\text{IRDY}}$	99	$\overline{\text{TRDY}}$	100	P33V
37	$\overline{\text{DEVSEL}}$	38	VDD	101	GND	102	$\overline{\text{STOP}}$
39	GND	40	$\overline{\text{LOCK}}$	103	$\overline{\text{PERR}}$	104	GND
41	$\overline{\text{SDONE}}$	42	$\overline{\text{SBO}}$	105	P33V	106	$\overline{\text{SERR}}$
43	PAR	44	GND	107	$\overline{\text{C/BE[1]}}$	108	GND
45	V(I/O)	46	AD[15]	109	AD[14]	110	AD[13]
47	AD[12]	48	AD[11]	111	GND	112	AD[10]
49	AD[09]	50	VDD	113	AD[08]	114	P33V
51	GND	52	$\overline{\text{C/BE[0]}}$	115	AD[07]	116	PMC-RSVD
53	AD[06]	54	AD[05]	117	P33V	118	PMC-RSVD
55	AD[04]	56	GND	119	PMC-RSVD	120	GND
57	V(I/O)	58	AD[03]	121	PMC-RSVD	122	PMC-RSVD
59	AD[02]	60	AD[01]	123	GND	124	PMC-RSVD
61	AD[00]	62	VDD	125	$\overline{\text{ACK64}}$	126	P33V
63	GND	64	$\overline{\text{REQ64}}$	127	GND	128	PMC-RSVD

J15 (PMC-3)

Pin	Signal	Pin	Signal
129	PCI-RSVD	130	GND
131	GND	132	$\overline{C/BE[7]}$
133	$\overline{C/BE[6]}$	134	$\overline{C/BE[5]}$
135	$\overline{C/BE[4]}$	136	GND
137	V(I/O)	138	PAR64
139	AD[63]	140	AD[62]
141	AD[61]	142	GND
143	GND	144	AD[60]
145	AD[59]	146	AD[58]
147	AD[57]	148	GND
149	V(I/O)	150	AD[56]
151	AD[55]	152	AD[54]
153	AD[53]	154	GND
155	GND	156	AD[52]
157	AD[51]	158	AD[50]
159	AD[49]	160	GND
161	GND	162	AD[48]
163	AD[47]	164	AD[46]
165	AD[45]	166	GND
167	V(I/O)	168	AD[44]
169	AD[43]	170	AD[42]
171	AD[41]	172	GND
173	GND	174	AD[40]
175	AD[39]	176	AD[38]
177	AD[37]	178	GND
179	GND	180	AD[36]
181	AD[35]	182	AD[34]
183	AD[33]	184	GND
185	V(I/O)	186	AD[32]
187	PCI-RSVD	188	PCI-RSVD
189	PCI-RSVD	190	GND
191	GND	192	PCI-RSVD

PMC+ Site

The PMC+ site consists of four 64-pin connectors. Three connectors are standard 64-pin PMC connectors (J16, J17, J23) that provide the 64-bit, 66 MHz PCI interface. The fourth connector (J22) 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 3–10 gives the connector pinout.

Warning!

BittWare uses P4 of the PMC connectors (see section 3.2.7) for our PMC+ extensions. If you are not mounting a BittWare PMC+ card on the Hammerhead-PCI board, the PMC board may have incompatibilities with the PMC+ (P4) connector. Call BittWare technical support for assistance.

Table 3–10 *PMC+ Connector Pinout*

J16 (PMC+ 1)				J17 (PMC+ 2)			
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

J23 (PMC+ 3)

Pin	Signal	Pin	Signal
1	PCI-RSVD	2	GND
3	GND	4	C/BE[7]
5	C/BE[6]	6	C/BE[5]
7	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

J22 (PMC+ 4)

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	RST
63	SCL	64	SDA

3.2.8 SDRAM

The Hammerhead-PCI has an industry-standard 144-pin connection for a standard SODIMM module. The SODIMM modules are available in 64, 128, 256, and 512 MB modules.

3.2.9 Test Points

The test point header (J21) provides access to various signals on the Hammerhead-PCI. Figure 3–9 shows where pin 1 is located on the connector, and Table 3–11 gives the connector pinout.

Figure 3–9 *Location of Pin 1 on the Test Point Header*

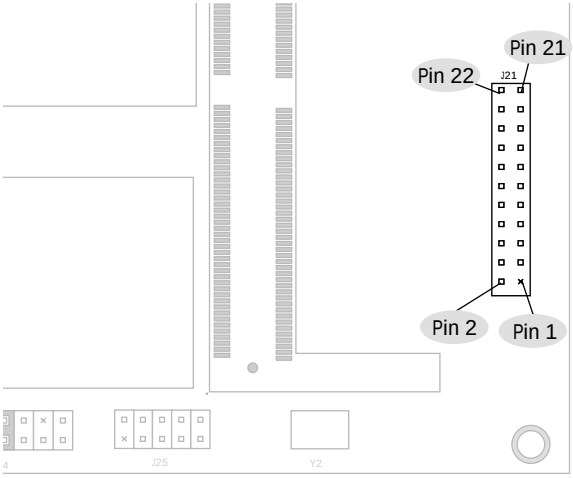


Table 3–11 *Test Point Connector Pinout (J21)*

Pin	Signal	Pin	Signal
1	MF1	2	HA_PRD0
3	MF2	4	HA_PRD1
5	HA_CIF	6	FPA_SPARECLR
7	GND	8	GND
9	HA1_TDMTFS	10	HA2_TDMTFS
11	HA3_TDMTFS	12	HA4_TDMTFS
13	GND	14	GND
15	CLK_MON	16	PRA_UBSEL
17	GND	18	GND
19	HA1_CLKOUT	20	HA2_CLKOUT
21	HA3_CLKOUT	22	HA4_CLKOUT

3.2.10 JTAG Header

The JTAG header (J24) allows in-circuit emulation with an optional ICE emulator (available from White Mountain DSP). All four ADSP-21160 DSPs are connected to the JTAG connector. Figure 3–10 shows where pin 1 is located on the JTAG header, and Table 3–12 gives the connector pinout.

Figure 3–10 *Location of Pin 1 on the JTAG Header*

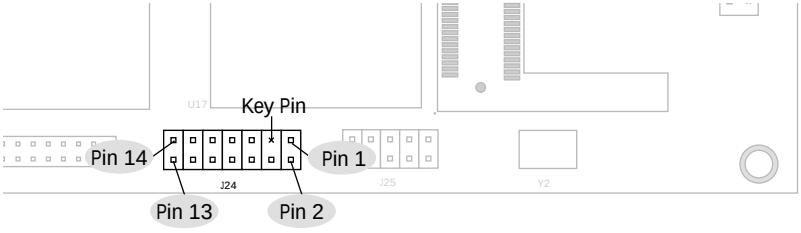


Table 3–12 *JTAG Header Pinout (J24)*

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

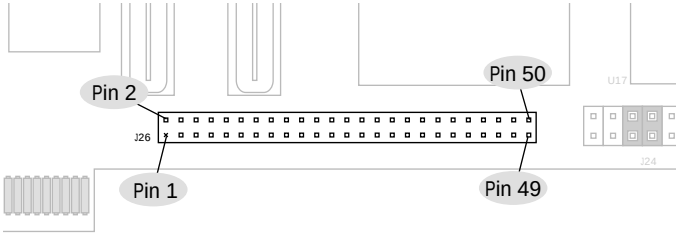
3.2.11 Expansion Connector

J26 is a 50-pin expansion connector. The expansion connector allows you to add another device onto the 8-bit peripheral bus via a ribbon cable.Table 3–13 gives the pinout, and Figure 3–11 shows where pin 1 is located.

Table 3–13 *Expansion Connector Pinout (J26)*

Pin	Signal	Pin	Signal
1	$\overline{\text{PRA\_INT}}$	2	GND
3	$\overline{\text{PRA\_RST}}$	4	GND
5	$\overline{\text{PRA\_WR}}$	6	GND
7	$\overline{\text{PRA\_RD}}$	8	GND
9	$\overline{\text{PRA\_ACK}}$	10	GND
11	PRA_D00	12	GND
13	PRA_D01	14	GND
15	PRA_D02	16	GND
17	PRA_D03	18	GND
19	PRA_D04	20	GND
21	PRA_D05	22	GND
23	PRA_D06	24	GND
25	PRA_D07	26	GND
27	$\overline{\text{PRA\_UBSEL}}$	28	GND
29	PRA_A00	30	GND
31	PRA_A01	32	GND
33	PRA_A02	34	GND
35	PRA_A03	36	GND
37	PRA_A04	38	GND
39	PRA_A05	40	GND
41	PRA_A06	42	GND
43	PRA_A07	44	GND
45	PRA_A08	46	GND
47	PRA_A09	48	GND
49	PRA_A10	50	GND

Figure 3–11 *Location of Pin 1 on the Expansion Connector*



The Hammerhead-PCI has eighteen configuration jumpers, which allow you to control and enable certain features on the board. Figure 3–12 shows where the Hammerhead-PCI's configuration jumpers are located on the board.

Figure 3–12 *Layout of the Hammerhead-PCI Configuration Jumpers*

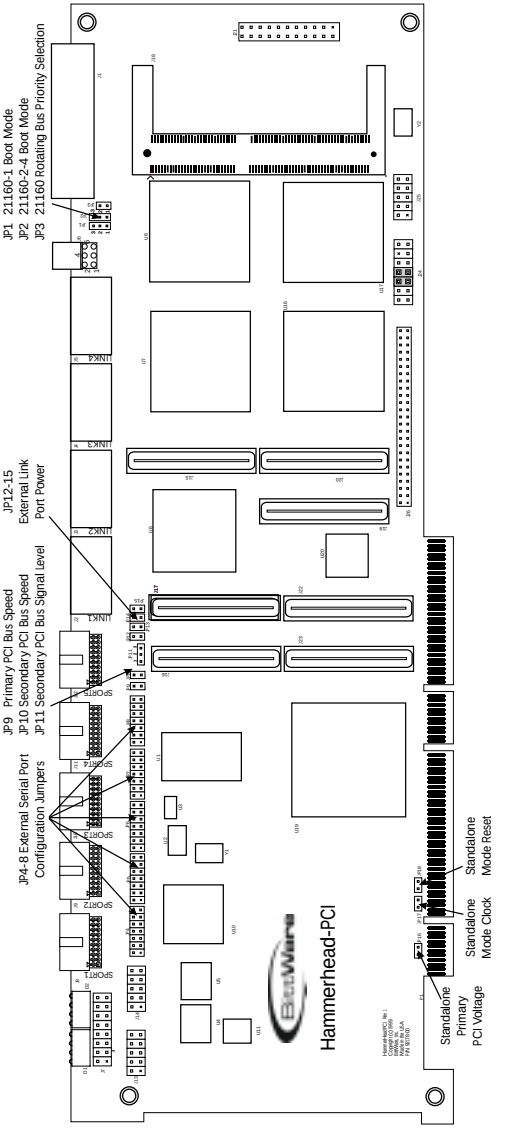


Table 3–14 below gives an overview of the jumpers, and section 2.2 describes their settings in more detail.

Table 3–14 *Overview of the Configuration Jumpers*

Jumper	Name	Description
JP1	21160-1 Boot Mode Select	Sets 21160-1 to boot from host computer, on-board FLASH, or remote processor via link port
JP2	21160-2–4 Boot Mode Select	Sets 21160-2–4 to boot from host computer, on-board FLASH, or remote processor via link port
JP3	21160 Bus Priority	Selects "fixed" or "rotating" bus priority scheme for ADSP-21160 processors
JP4–JP8	Serial Port Jumper Blocks	JP4 configures serial port J8 as a standard connection to 21160-1 JP5 configures serial port J9 as a standard connection to 21160-2 JP6 configures serial port J10 as a standard connection to 21160-3 JP7 configures serial port J11 as a standard connection to 21160-4 JP8 configures serial port J12 as a TDM serial connection to all four ADSP-21160 DSPs
JP9	Primary PCI Bus Speed	Sets the speed of the primary PCI bus to either 33 MHz or 66 MHz
J10	Secondary PCI Bus Speed	Sets the speed of the secondary PCI bus to either 33 MHz or 66 MHz
J11	Secondary PCI Signal Level	Sets the signal level of the secondary PCI bus to 3.3 or 5 Volts
J12–15	External Link Port Connector Power	Gate power to the external link port connectors
J16	Standalone Primary PCI Voltage	Jumper must be on to operate board in standalone mode
J17	Standalone Mode Clock	Jumper must be on to operate board in standalone mode
J18	Standalone Mode Reset	Jumper must be on to operate board in standalone mode

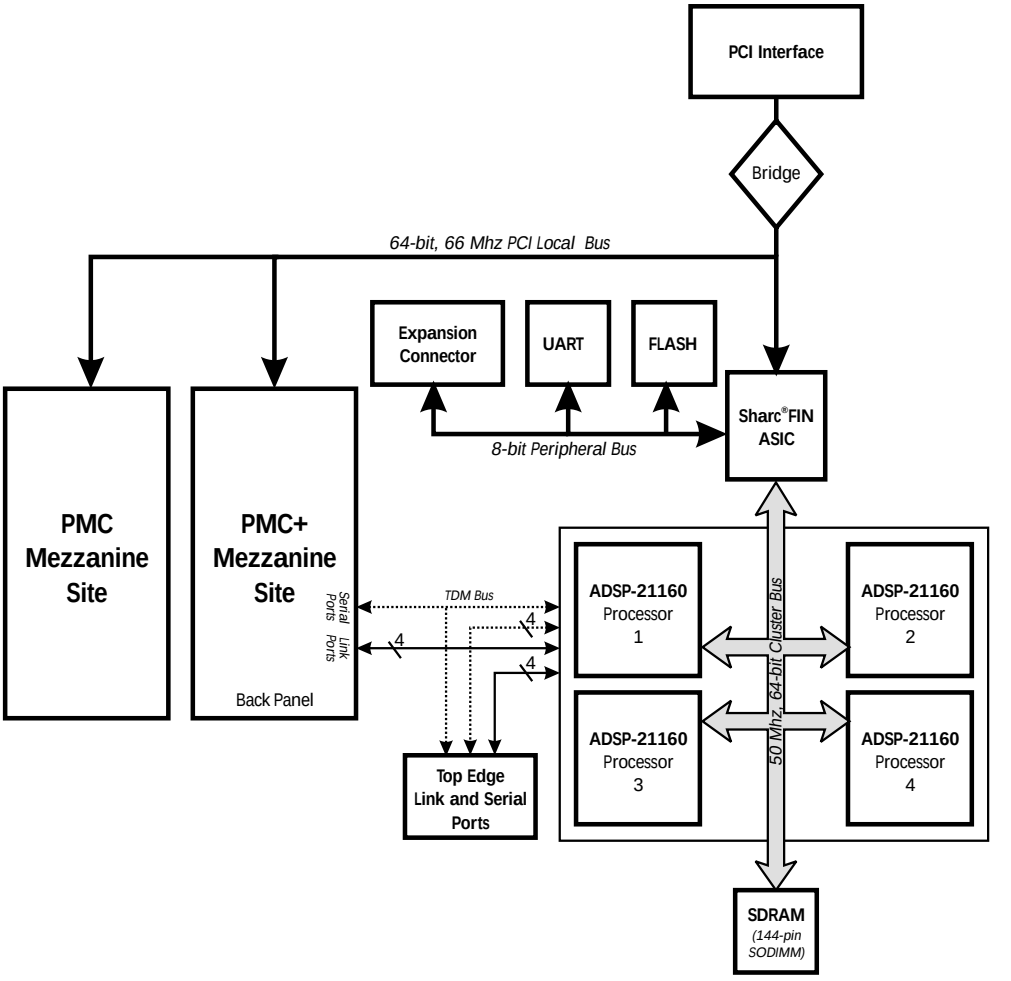
Chapter 4
Hammerhead-PCI Board Architecture

This chapter discusses the architecture of the board, describing how the DSPs communicate with other processors and the host and how they access memory. It discusses four modes of communication:

- Serial ports
- Link ports
- Flags and interrupts
- Data Buses

This section gives a brief overview of the entire board and discusses how the DSPs communicate with all of the components on the board.

Figure 4–1 *Block Diagram of the Hammerhead-PCI System*



4.1.1 ADSP-21160 Architecture Overview

The Hammerhead-PCI's DSPs can communicate with memory and processors on the board and on other boards via serial ports, link ports, flags and interrupts, and data buses. Each ADSP-21160 DSP has six 100 Mbyte/sec link ports, two 50 Mbit/sec serial ports, four flags, and three interrupts. Each processor also connects to the JTAG emulator port for in-circuit emulation.

4.1.2 Bus Architecture Overview

Each processor is connected to a common 50 MHz, 64-bit *ADSP-21160 cluster bus*, which gives it access to the other three processors and to up to 512 MB of SDRAM. The ADSP-21160 cluster bus is connected to the Sharc FIN ASIC, which allows devices on the ADSP-21160 cluster bus to communicate with devices on the *PCI bus*.

The 66 MHz, 64-bit PCI bus is broken into two separate buses – a primary and a secondary PCI bus. A PCI-to-PCI "bridge chip" (Intel 21154) connects the primary and secondary PCI buses. The PCI bus connects the host computer with devices on the ADSP-21160 cluster bus, devices on the *peripheral bus*, and the PMC and PMC+ interfaces.

The 25 MHz, 8-bit peripheral bus gives the host and the ADSP-21160 DSPs access to the Hammerhead-PCI's FLASH memory, dual UART, and expansion connector.

4.1.3 PMC+ Interface Architecture Overview

The Hammerhead-PCI's PMC+ interface is connected to the 64-bit, 66 MHz secondary PCI bus, which allows it to access the FLASH memory, the UART, the PCI interface, and all four ADSP-21160 DSPs. The PMC+ interface connector also connects directly to the ADSP-21160 DSPs via four link ports, two PMC-to-DSP interrupts, two DSP-to-PMC interrupts, a TDM serial bus, and a reset line.

Each ADSP-21160 DSP has two 50 Mbit/sec serial ports. One serial port from each processor is connected to a TDM serial bus, which interconnects all four DSPs, the PMC+ interface, and the RS-232 interface. The remaining serial port on each processor is connected to one of the board's four external serial ports. Figure 4–2 to Figure 4–4 show the serial port connections on the quad, dual, and single-processor versions of the Hammerhead-PCI board.

Figure 4–2 *Block Diagram of Serial Port Connections: Quad Processor Board*

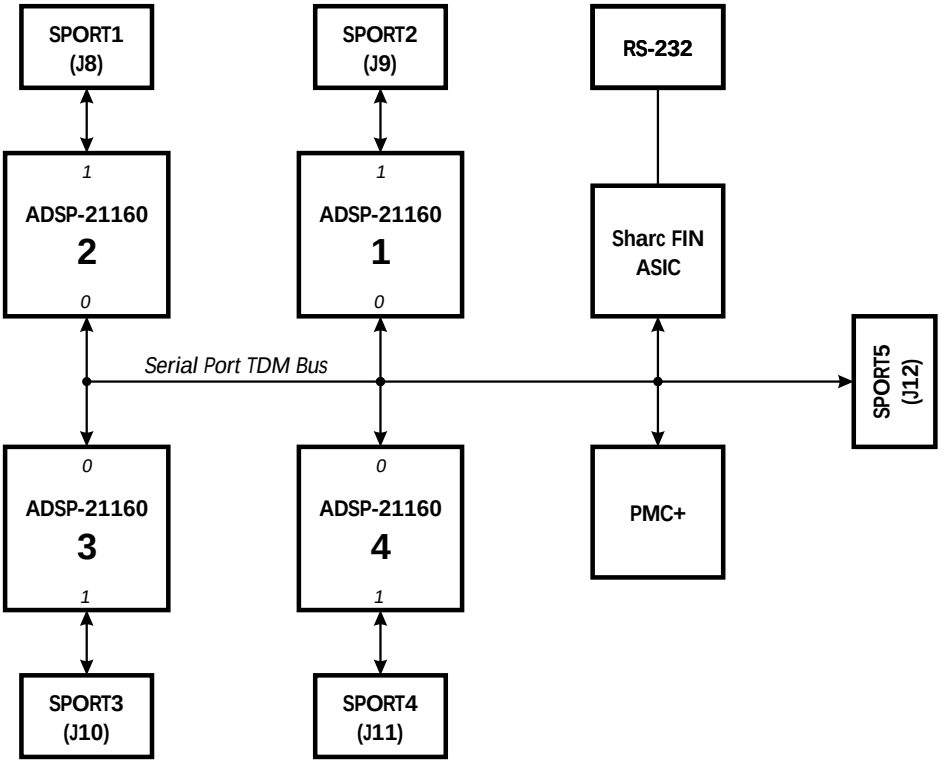


Figure 4–3 *Block Diagram of Serial Port Connections: Dual Processor Board*

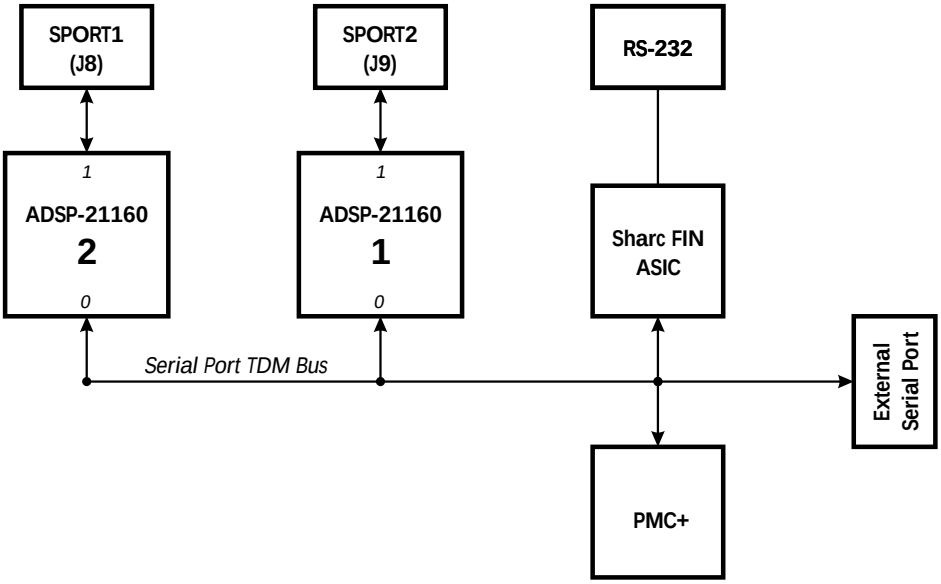
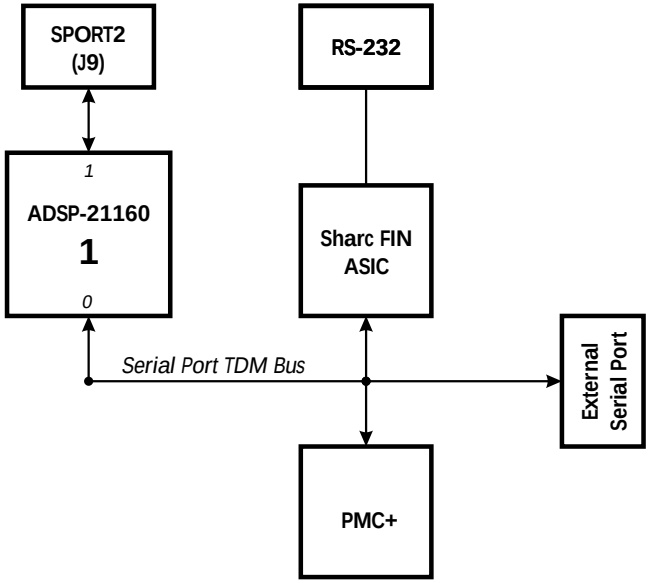


Figure 4–4 *Block Diagram of Serial Port Connections: Single Processor Board*



Each ADSP-21160 DSP has four 100 Mbyte/sec link ports, which allow the DSPs to communicate with the other DSPs on the Hammerhead-PCI board, DSPs on a PMC module, and DSPs on other boards. Figure 4–5 below shows how the link ports on the four DSPs are connected.

Figure 4–5 *Block Diagram of Link Port Connections: Quad Processor Board*

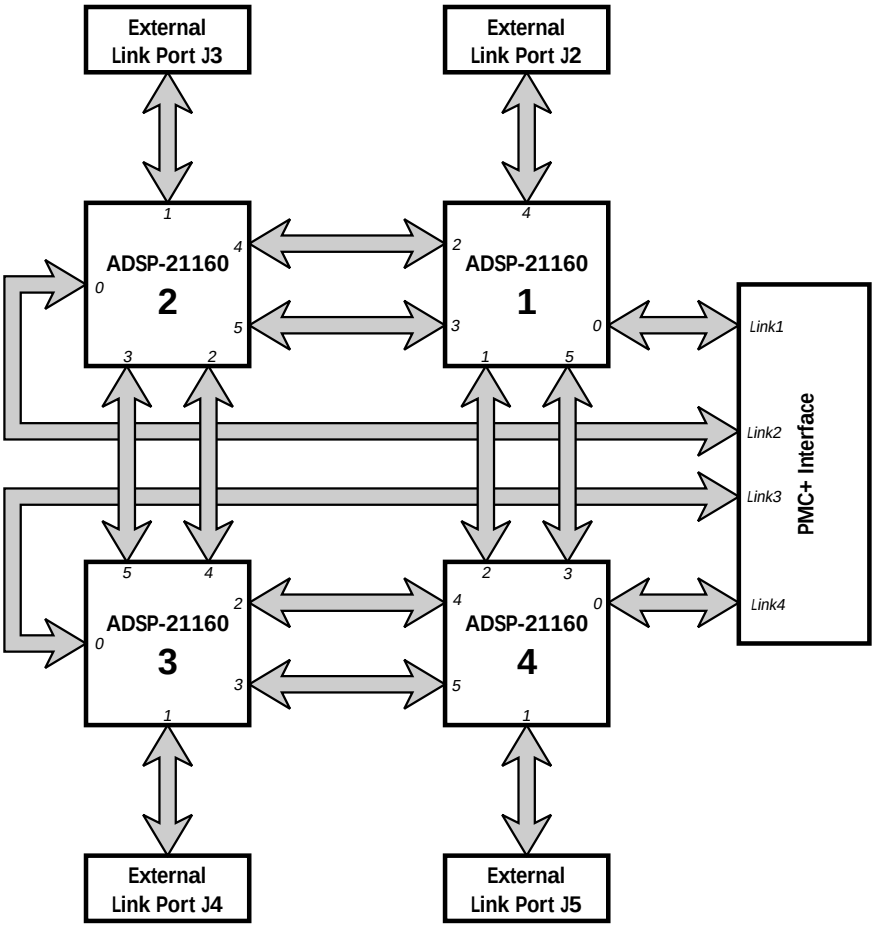


Figure 4–6 *Block Diagram of Link Port Connections: Dual Processor Board*

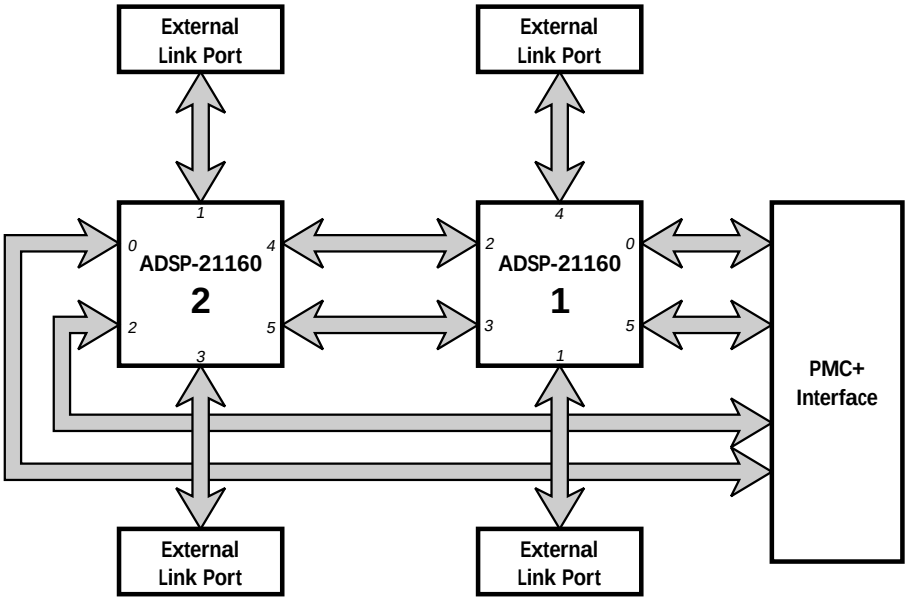
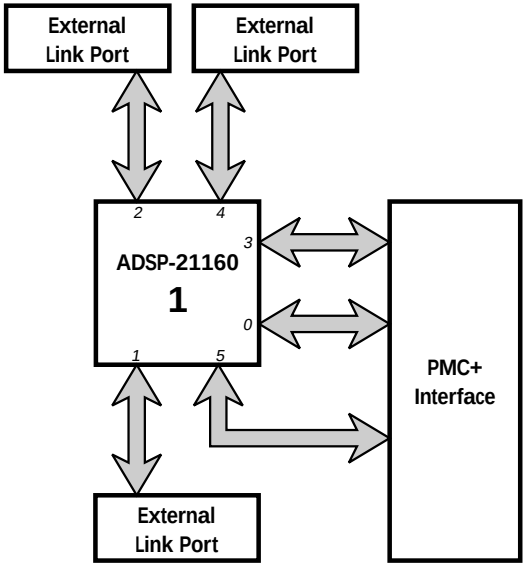


Figure 4–7 *Block Diagram of Link Port Connections: Single Processor Board*



4.3.1 Link Port Connections to DSPs in the Same Cluster

Four link ports on each processor are dedicated for interprocessor communication.

4.3.2 Link Port Connections to the PMC+ Interface

One link port from each processor extends to the PMC+ interface.

4.3.3 Link Port Connections to External Connectors

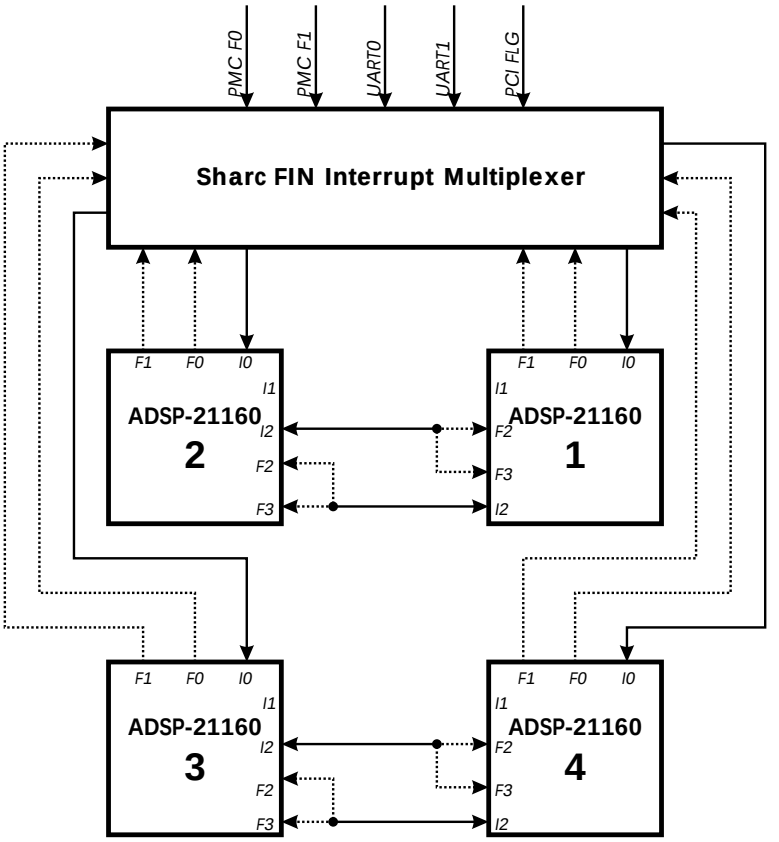
One link port from each DSP extends to an external link port connector; the processors can boot from a remote processor via these link ports (see section 2.8.2).

4.4 Flag and Interrupt Connections

This section covers the DSPs' flag and interrupt connections. It discusses generating DSP-to-DSP interrupts via flags, interrupts to and from the PCI interface, interrupts to and from ADSP-21160 DMA channels, interrupts to and from ADSP-21160 mailbox registers, interrupts to and from the host, and interrupts to and from the PMC interface.

Two flags and one interrupt from each DSP connect to the Sharc FIN ASIC. Using registers in the Sharc FIN, you can change the routing of those flags and interrupts (see section 5.3.9).

Figure 4–8 *Block Diagram of Flag and Interrupt Connections*



4.4.1 Flag Connections

Figure 4–8 and Table 4–1 illustrate the flag connections on the Hammerhead-PCI. Each ADSP-21160 DSP has four flag signals. Two flags from each DSP connect to the Sharc FIN ASIC, where you can configure their routing. The remaining two flags from each DSP connect to LEDs and to the other ADSP-21160 DSPs for interprocessor communication.

Table 4–1 *ADSP-21160 Flag Connections*

	21160-1	**21160-2**	**21160-3**	**21160-4**
FLAG0	Sharc FIN*	Sharc FIN	Sharc FIN	Sharc FIN
FLAG1	Sharc FIN	Sharc FIN	Sharc FIN	Sharc FIN
FLAG2	21160-2 I2 LED	21160-1 I2 LED	21160-4 I2 LED	21160-3 I2 LED
FLAG3	21160-2 I2 LED	21160-1 I2 LED	211604 I2 LED	21160-3 I2 LED

* FLAG0 and FLAG1 on each DSP connect to the Sharc FIN ASIC interrupt multiplexer. From the Sharc FIN, you can route them to different locations on the board.

4.4.2 Interrupt Connections

Figure 4–8 and Table 4–2 illustrate the interrupt connections on the Hammerhead-PCI. Each ADSP-21160 DSP has three interrupts. One interrupt from each DSP is dedicated for interprocessor communication. One interrupt from each DSP goes to the Sharc FIN, where you can configure its routing in the Sharc FIN's configuration space. You can program the DSPs and host interface to generate and receive interrupts from the following sources:

- Flags from other ADSP-21160 DSPs
- Local interrupt from the Sharc FIN ASIC, which is also programmable and has many sources (for example, the DMA channels)
- Sharc FIN Mailbox registers
- Peripheral bus
- RS-232 port

Table 4–2 *ADSP-21160 Interrupt Connections*

	21160-1	21160-2	21160-3	21160-4
IRQ0	Sharc FIN*	Sharc FIN	Sharc FIN	Sharc FIN
IRQ1				
IRQ2	21160-2 F2 21160-2 F3	21160-1 F2 21160-1 F3	21160-4 F2 21160-4 F3	21160-3 F2 21160-3 F3

* IRQ0 on each DSP connects to the Sharc FIN ASIC interrupt multiplexer. From the Sharc FIN, you cann route them to different locations on the board.

DSP-to-DSP Interrupts

Each DSP can generate and receive interrupts to and from the other three DSPs. One interrupt (IRQ2) on each processor receives interrupts from FLAG2 and FLAG3 on another processor. IRQ2 and FLAG2–3 on 21160-1 and 21160-2 are interconnected, and IRQ2 and FLAG2–3 on 21160-3 and 21160-4 are interconnected. The routing of the remaining two interrupts on each processor (IRQ0 and IRQ3) is user-configurable in the Sharc FIN ASIC.

DSP-to-Host and Host-to-DSP Interrupts

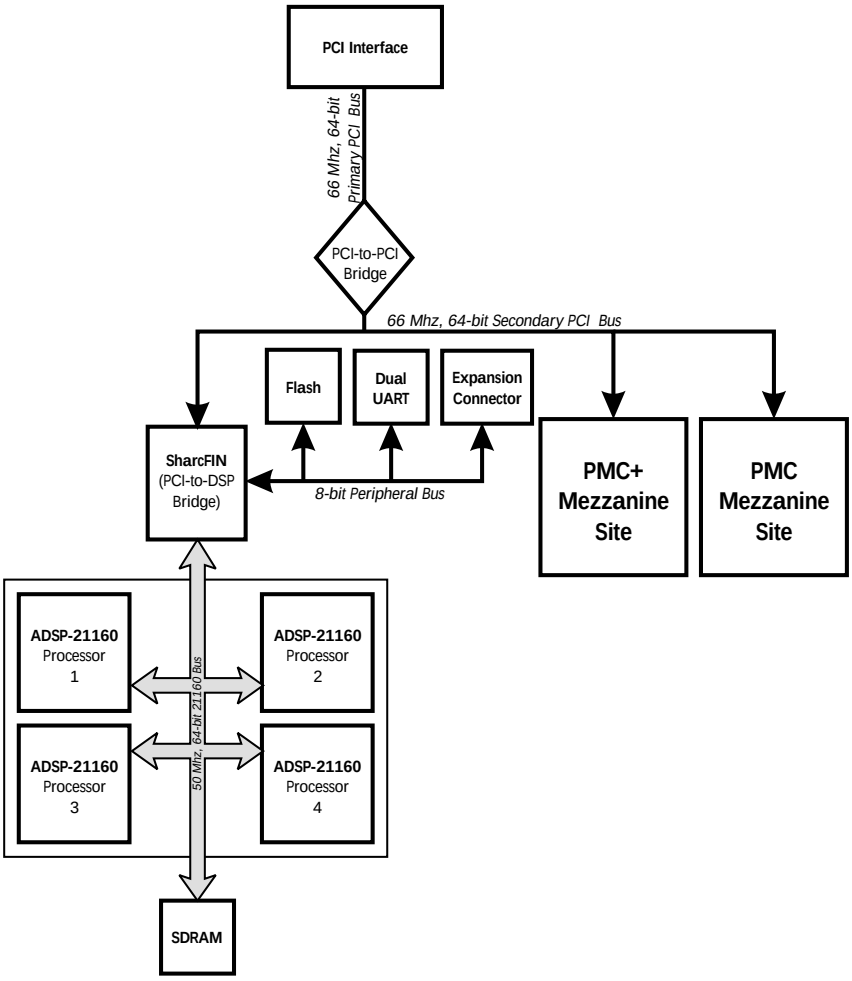
Registers in the Sharc FIN's configuration space allow the host and the ADSP-21160 DSPs to generate and receive interrupts from several sources. Depending on how you configure the interrupts in the Sharc FIN, each ADSP-21160 DSP will be able to receive up to two interrupts from the host, and the host will be able to receive one interrupt from the DSPs. Two flags on each DSP are available to generate interrupts to the host.

DSP-to-PMC+ and PMC+-to-DSP Interrupts

You can configure the interrupt connections between the PMC+ interface and the DSPs with registers in the Sharc FIN. Section 5.3.9 explains the register settings in detail.

The Hammerhead-PCI's PCI interface consists of three buses: the primary and the secondary PCI buses and the ADSP-21160 cluster bus. A PCI-to-PCI bridge chip connects the primary and secondary PCI buses. The Sharc FIN ASIC connects the secondary PCI bus with the ADSP-21160 cluster bus. An 8-bit peripheral bus is also connected to the Sharc FIN, providing access to the FLASH memory, the dual UART, and the expansion connector. Figure 4–9 below is a diagram of the PCI interface, and sections 4.5.1 and 4.5.2 explain it in more detail.

Figure 4–9 *Block Diagram of the PCI Interface*



4.5.1 PCI-to-PCI Interface

The PCI-to-PCI interface consists of the primary PCI bus and the PCI-to-PCI bridge chip (Intel 21154), which connect the host to the Sharc FIN (see section 4.5.2 below). The PCI-to-PCI bridge chip is a bridge between the primary and secondary PCI buses.

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.

4.5.2 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 Sharc FIN ASIC (PCI-to-PCI bridge) and two buses: the secondary PCI bus and the ADSP-21160 cluster bus.

Secondary PCI Bus

The Sharc FIN ASIC connects the secondary PCI bus to the ADSP-21160 cluster bus. The 66 MHz, 64-bit secondary PCI bus operates at 5 volts, but is jumper-configurable to run at 3.3 volts. By default, this bus operates at the same speed as the primary PCI bus, but a jumper setting will allow you to use the bus at half the speed of the primary PCI bus. However, this bus cannot run at 66 MHz when the primary bus is running at 33 MHz.

Note

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

ADSP-21160 Cluster Bus

The ADSP-21160 cluster bus is a 50 MHz, 64-bit bus that connects the four ADSP-21160 processors and a bank of up to 512 MB SDRAM. It is connected to the PCI interface through the Sharc FIN. 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

- BMS
- MS1
- MS3
- Unbanked memory

4.5.3 Peripheral Bus

The 25 MHz, 8-bit peripheral bus connects to low-speed peripherals such as the FLASH RAM, the dual UART, and the expansion connector. 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 connects to the Sharc FIN, which connects it to the ADSP-21160 DSPs and the PCI interface. The peripheral bus operates at either 3.3 volts or 5 volts.

Chapter 5
Programming Details

This chapter provides programming details for the DSPs and the Sharc FIN ASIC. It provides the information you will need when you are writing programs that allow the DSPs and the host to communicate. This section is not comprehensive. Instead, it provides only programming information that is specific to the Hammerhead-PCI board. You will also need to refer to documentation for the DSP21k-SF Toolkit, the ADSP-21160 SHARC processors, the Analog Devices development tools, and the Sharc FIN ASIC.

This section gives programming details for programs that run on the ADSP-21160 DSPs.

5.1.1 Resources Available to the ADSP-21160s

This section discusses the resources available to each processor: memory banks, flags and interrupts, serial ports, and link ports. The following tables summarize how the DSPs' resources are used on the Hammerhead-PCI. The row labeled "MS" refers to the DSPs' external memory select lines (MS0–MS3).

Table 5–1 *Resources for 21160-1*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM Bus	Sharc FIN*	Sharc FIN	PMC
1	FLASH	External		Sharc FIN	21160-4 L5
2	Sharc FIN		21160-2 F2 21160-2 F3	21160-2 I2 LED	21160-4 L6
3	UART			21160-2 I2 LED	External
4					21160-2 L3
5					21160-2 L4

* IRQ0, FLAG0, and FLAG1 on each processor connect to the Sharc FIN ASIC interrupt multiplexer. From the Sharc FIN, you can route them to different locations on the board.

Table 5–2 *Resources for 21160-2*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM Bus	Sharc FIN	Sharc FIN	PMC
1	FLASH	External		Sharc FIN	External
2	Sharc FIN		21160-1 F2 21160-1 F3	21160-1 I2 LED	21160-1 L5
3	UART			21160-1 I2 LED	21160-1 L6
4					21160-3 L3
5					21160-3 L4

Table 5–3 *Resources for 21160-3*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM Bus	Sharc FIN	Sharc FIN	PMC
1	FLASH	External		Sharc FIN	External
2	Sharc FIN		21160-4 F2 21160-4 F3	21160-4 I2 LED	21160-2 L5
3	UART			21160-4 I2 LED	21160-2 L6
4					21160-4 L3
5					21160-4 L4

Table 5–4 *Resources for 21160-4*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM Bus	Sharc FIN	Sharc FIN	PMC
1	FLASH	External		Sharc FIN	External
2	Sharc FIN		21160-3 F2 21160-3 F3	21160-3 I2 LED	21160-3 L5
3	UART			21160-3 I2 LED	21160-3 L6
4					21160-1 L2
5					21160-1 L3

5.1.2 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. The sections below describe each type of memory.

Internal Memory

Internal memory addresses an ADSP-21160 DSP's on-chip, dual-ported SRAM. 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 the 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-PCI 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.

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. Section 5.1.3 discusses the external memory in more detail.

5.1.3 Accessing External Memory

Each ADSP-21160 DSP has four memory select lines, MS0–MS3, which allow it to access the external memory banks located on the 64-bit ADSP-21160 cluster bus and on the 8-bit peripheral bus.

Setting the Size of the External Memory Banks

The MSIZE bits of the ADSP-21160's SYSCON register define the size of the four banked sections of memory, which are numbered 0 to 3. All four memory banks are the same size (refer to the *ADSP-21160 SHARC User's Manual* for more details). Bank 0 starts at 0x0800 0000, and its MSIZE determines the starting address of each of the other banks.

Use the Hammerhead-PCI's configuration data to set the MSIZE (see section 2.4.3). The default setting for MSIZE should be equal to the size of the largest external memory device, which is the SDRAM. Table 5–5 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 5–5 *Recommended MSIZE Settings for the Hammerhead-PCI*

TBD

Accessing the SDRAM

The Hammerhead-PCI supports a bank of up to 512 MB of SDRAM, which is located on the 64-bit ADSP-21160 cluster bus. The SDRAM is accessible via MS0.

Accessing the FLASH Memory

The Hammerhead-PCI supports a 2 MB bank of FLASH memory, which is located on the 8-bit peripheral bus. The FLASH memory is accessible via MS1.

Accessing the Sharc FIN ASIC

The DSPs access the Sharc FIN ASIC via MS2. The Sharc FIN's user-configurable registers begin at offset 0x40 from the base address of MS2.

Accessing the Dual UART

The Hammerhead-PCI's dual UART is located on the 8-bit peripheral bus. The DSPs access the UART via MS3.

5.2 Programming Details for the Sharc FIN ASIC PCI Interface

BittWare's Sharc FIN ASIC flexibly interfaces Analog Devices' SHARC DSPs to a wide range of the Hammerhead-PCI's interfaces, including: 64/66 MHz PCI bus (rev. 2.2 compliant), SDRAM, UART, I^2S serial ports, FLASH, and a general-purpose expansion bus (the 8-bit peripheral bus).

The Sharc FIN consists of two main sections: The first section acts as the bridge between the PCI interface and the ADSP-21160 DSPs and covers the address space from 0x00 to 0x3F. The second section contains the logic for the board and several user-configurable registers, which cover the address space from 0x40 to 0x5F.

5.3 Setting the Sharc FIN User-Configurable Registers

This section describes the memory locations and settings for the Sharc FIN's user-configurable registers. All addresses described in this section are offsets from the base address of MS2 and are accessible from the ADSP-21160 DSPs and from the PCI interface. Table 5–6 below gives the memory mapping for the user-configurable registers in the Sharc FIN.

Note

Most of the user configurable 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 5–6 *Memory Map for the Sharc FIN User-Configurable Registers*

Address	Register	Read / Write	Description
0x40	Address Override Register*	W	Allows addressing of IOP registers when ADSP-21160 is using a host packing mode
0x41	Status Register	R	Indicates the number of processors, ADSP-21160 cluster bus status, and last reset source
0x42	Peripheral Bus Configuration Register	R / W	Configures and shows status of wait cycles of the 8-bit peripheral bus
0x43	Watchdog Config Register	WORM†	Enables and disables the watchdog timer
0x44	PMCA Config Register	R / W	Configures the PMC+ interface
0x45	PMCB Status Register‡		Reserved
0x46	Onboard I^2C Control Register	R / W	Controls the I^2C™ interface
0x47	PMC I^2C Control Register	R / W	Controls the I^2C interface to the PMC+ interface
0x48	SD Size Config Register*	W	Resets and reinitializes the SDRAM controller
0x49	SD Config Word Register*	W	Configures the SDRAM
0x4A	SD Window Register*	W	Selects which 16 MB of SDRAM the ADSP-21160 DSPs 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, 59	Unused		
0x58	PCInt	R / W	Configures PCI interrupts
0x5A	PMCI0	R / W	Configures PMC interrupts
0x5B	PMCI1	R / W	Configures PMC interrupts
0x5C–5D	Unused		
0x5E	Flags	R	Shows state of all flags
0x5F	IRQ	R	Shows state of all interrupts

* Use with caution.
† Write Once Read Many
‡ This register is currently unimplemented.

5.3.1 Setting the Address Override Register

The Address Override register 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 SYSCON register has been configured.

Note

Only use this register if you are writing your own programs.

Table 5–7 *Contents of the Address Override Register*

Bit	Name	Description
B0	A0 Override En	?
B1	Overridden A0	?
B2	Bus Lock Request	Requests that Sharc FIN acquire the ADSP-21160 cluster bus and not give it up
B3	Destructive FIFO Read Enable	When bit=1, a read to the DMA FIFOs causes the FIFOs to advance. When bit=0, a read to the DMA FIFOs does not cause the FIFOs to advance.

Setting the Address Override Bits

TBD

Setting the Bus Lock Request Bit

The Bus Lock Request bit allows the Sharc FIN to acquire the ADSP-21160 cluster bus and locks access to the bus so that only the Sharc FIN can access it. Table 5–8 below shows the settings for bit 2, the Bus Lock Request bit.

Table 5–8 *Bit Settings for the Bus Lock Request Bit*

B2 Setting	Description
0	Disabled
1	Requests that the Sharc FIN acquire the ADSP-21160 cluster bus and not give it up

5.3.2 Reading the Status Register

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

Table 5–9 *Contents of the Status Register*

Bit	Name	Description
B0	HHPres0	Combined settings of B0 and B1 indicate how many processors are installed on the board
B1	HHPres1	
B2	Bus Locked	Indicates whether the Sharc FIN has locked and acquired the ADSP-21160 cluster bus
B3	Last Reset Source	Indicates whether the PCI interface or the watchdog was the source of the last board reset
B4	SPCI Done	Refer to Quick Logic documentation for details
B5	RCV0_FIFO_EF	Receive FIFO 0 Empty Flag; refer to Quick Logic documentation for details
B6	RCV1_FIFO_EF	Receive FIFO 1 Empty Flag; refer to Quick Logic documentation for details

Determining How Many Processors are Installed on the Board

Bits 0 and 1 of the status register show how many processors are on the Hammerhead-PCI board. Table 5–10 below shows their combined settings.

Table 5–10 *Settings for the Processor Bits*

B0	B1	Description
0	0	1 processor
1	0	2 processors
1	1	4 processors
0	1	Reserved

Determining Whether the ADSP-21160 Bus is Locked

Bit 2, the Bus Locked bit, shows whether the processors on the board have granted ownership of the ADSP-21160 cluster bus to the Sharc FIN. Table 5–11 below shows the settings for the Bus Locked bit.

Table 5–11 *Bit Settings for the Bus Locked Bit*

B2 Setting	Description
0	ADSP-21160 cluster bus is not locked
1	ADSP-21160 cluster bus is locked

Determining the Last Reset Source

Bit 3, the Last Reset Source bit, shows whether the PCI interface or the watchdog timer was the source of the last board reset. Table 5–12 below shows the settings for the Last Reset Source bit.

Table 5–12 *Bit Settings for the Last Reset Source Bit*

B3 Setting	Description
0	PCI reset
1	Watchdog reset

5.3.3 Setting the Peripheral Bus Configuration Register

The Peripheral Bus Configuration register allows you to configure the wait cycles of the Hammerhead-PCI's 8-bit peripheral bus.

Table 5–13 *Contents of the Peripheral Bus Configuration Register*

Bit	Name	Description
B0–B3	PCI to Pbus Wait	Select the number of wait cycles the Sharc FIN will wait before completing a transaction on the peripheral bus
B4	Pbus Ack Enable	Selects whether the Sharc FIN will monitor the peripheral bus Ack line after the peripheral bus wait time has expired
B5	Pbus Reset	Resets the peripheral bus reset line, the Flash, and the UART

Selecting the Number of Wait Cycles

Bit 0 – Bit 3 set the number of wait cycles the Sharc FIN must 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). Table 5–14 below shows the settings for the bits.

Table 5–14 *Default Setting for Selecting the Number of Wait Cycles*

B0	B1	B2	B3	Description
1	0	1	0	6 wait cycles

Enabling the Peripheral Bus Ack Line

The setting of the Pbus Ack Enable bit (B4) determines whether the Sharc FIN will wait five wait cycles and then monitor the peripheral bus Ack line or whether it will wait five wait cycles and then consider the transaction complete. Table 5–15 gives the settings for the Pbus Ack Enable bit.

Note

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

Table 5–15 *Settings for the Pbus Ack Enable Bit*

B4	Description
0	Sharc FIN will wait the selected number of wait cycles and consider the transaction complete
1	Sharc FIN will wait the selected number of wait cycles and then monitor the Ack line

Setting the Peripheral Bus Reset Bit

When set to 1, bit 5, the Peripheral Bus Reset bit, resets the Flash, the UART, and all devices on the peripheral bus. The reset stays active until cleared by another write to the register. You can also reset the Flash, the UART, and all devices on the peripheral bus via a board reset.

5.3.4 Setting the Watchdog Configuration Register

The Watchdog Configuration register 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 can not be disabled except by a board reset, which can be from the PCI interface, the watchdog, or an external source.

Table 5–16 *Contents of the Watchdog Configuration Register*

Bit	Name	Description
B0	WatchdogEn0	Enable the watchdog timer and select its time-out time
B1	WatchdogEn1	
B2–B3	Unused	
B4–B7	H1F0 En, H2F0 En, H3F0 En, H4F0 En	Selects which processor will strobe the watchdog timer

Enabling the Watchdog and Setting its Time-out

Bit 0 and bit 1 enable and disable the watchdog and set its time-out time. Table 5–17 below shows their settings.

Table 5–17 *Settings for the Watchdog Enable Bits*

B0	B1	Description
0	0	Disabled
0	1	Enabled; short time-out (200 ms)
1	0	Enabled; medium time-out (600 ms)
1	1	Enabled; long time-out (1.2 s)

Selecting the Processor to Run the Watchdog

Bits 4–7 select the processor that will run the watchdog. You can select more than one processor, but it is not recommended. Table 5–18 below shows the settings for bits 4–7.

Table 5–18 *Settings for the Processor Selection Bits*

B4	B5	B6	B7	Description
1	0	0	0	21160-1 FLAG0 will strobe the watchdog timer
0	1	0	0	21160-2 FLAG0 will strobe the watchdog timer
0	0	1	0	21160-3 FLAG0 will strobe the watchdog timer
0	0	0	1	21160-4 FLAG0 will strobe the watchdog timer

5.3.5 Setting the PMCA Configuration Register

The PMCA Configuration register is a read/write register that configures the bus mode lines of the PMC+ interface and allows you to read their status. Table 5–19 below shows the contents of the PMCA Configuration register.

Table 5–19 *Contents of the PMCA Configuration Register*

Bit	Name	Description
B0	PMC Flg/Int En	Configures the PMC+ interface's bus mode lines to be used as flag interrupts
B1–4	BusMode1, BusMode2, BusMode3, BusMode4	BusMode lines per the PMC spec if the flag interrupts are disabled

Using the Bus Mode Lines as Flag Interrupts

The PMC Flg/Int En bit (B0) allows you to configure the PMC+ interface's bus mode lines as flag interrupts. The option of using the bus mode lines as flag interrupts is a feature of the PMC+ form factor; to work properly, it must be enabled on both the PMC+ card and the host board. Table 5–20 below shows the settings for the bit.

Table 5–20 *Settings for the PMC Flg/Int En Bit*

B0 Setting	Description
0	The PMC+ interface's bus mode lines will be used as flag interrupts
1	The PMC+ interface's bus mode lines will be used as bus mode lines

Using the Bus Mode Lines According to the PMC Specification

When the flag interrupts are disabled, the BusMode lines work according to the PMC specification. Bits 1–3 are Bus Mode lines 2–4. Bit 4 is Bus Mode line 1 and is an input; it is used to show the presence of a PMC card on the board. 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. Table 5–21 describes the bits.

Table 5–21 *Default Setting for the Processor Selection Bits*

Bit	Setting	Description
B1*	1	Bus Mode line 2
B2	0	Bus Mode line 3
B3	0	Bus Mode line 4
B4	Input	Bus Mode line 1 (PMC presence detect line); indicates whether a PMC board is present

* Bit 4 is an input. Bits1–3 are outputs.

5.3.6 Setting the Onboard I²C Control Register

The Onboard I²C Control register controls the I²C interface. The I²C 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 5–22 below shows the contents of the register.

As per standard I²C, both the clock and data lines are pulled high. Devices on the I²C bus either do not drive the bus or they drive it low. Any device on the I²C bus can drive either the clock or data line low when required. You can also read the actual status of the lines.

Table 5–22 *Contents of Onboard I²C Control Register*

Bit	Name	Description
B0	Clock	Read / Write; On write, drives the clock line. On read, shows the state of the clock line
B1	Data	Read / Write; On write, drives the data line. On read, shows the state of the data line

When you write a 1 to either the clock or data line in this register, the Sharc FIN does not drive the corresponding line. When you write a 0 to either the clock or the data line, the Sharc FIN 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 5–23 below shows the effect of the values written to the Clock and Data bits.

Table 5–23 *Settings for Writing the Clock and Data Bits*

Value Written	Description
0	Drives the line low; when read back, shows 0
1	When read back, shows the actual state of the I²C line

5.3.7 Setting the PMC I^2C Control Register

The PMC I^2C control register controls the I^2C interface to the PMC+ interface. The settings for this register are the same as the settings for the Onboard I^2C register, except that all settings apply to the PMC+ interface I^2C instead of the on-board I^2C.

5.3.8 Configuring the SDRAM

The Sharc FIN contains three registers that configure the SDRAM:

- the SDRAM Size Configuration register
- the SDRAM Configuration Word register
- the SDRAM Window register

Setting the 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 SODIMM modules used on the board. Table 5–24 below shows the contents of the register.

Table 5–24 *Contents of the SDRAM Size Configuration Register*

Bit	Name	Description
B0	SD Bank Size 0	??
B1	SD Bank Size 1	??
B2	SD RF Size	
B3	SD Reset*	Resets the SDRAM controller and reinitializes the SDRAM

* Do not change this bit.

Setting the SDRAM Configuration Word Register

The SDRAM Configuration Word register configures the SDRAM. This register is already set with appropriate settings. User modification should not be required.

Warning!

Do not change the value of this register. Modifying this register will cause the SDRAM not to function properly.

Setting the SDRAM Window Register

The SDRAM Window Register lets you select which 16 MB section of memory in the SDRAM to view from the host over the PCI interface. However, you will not need to configure this register since the Diag21k utility, which is included with the DSP21k-SF Toolkit, will take care of setting these bits. Table 5–25 lists the bits included in this register.

Table 5–25 *Contents of the SDRAM Window Register*

Bit	Name	Description
B0	Window A0	Selects window A0 of the SDRAM
B1	Window A1	Selects window A1 of the SDRAM
B2	Window A2	Selects window A2 of the SDRAM
B3	Window A3	Selects window A3 of the SDRAM
B4	Window A4	Selects window A4 of the SDRAM

5.3.9 Configuring the ADSP-21160, PCI, and PMC+ Interrupts

The registers from offset 0x50 to 0x5B configure the direction of the ADSP-21160, PCI, and PMC interrupts and show their status. Each register is identical except that each one corresponds to a different interrupt on the PCI interface, PMC+ interface, or ADSP-21160 processors. Table 5–26 shows which register corresponds to which interrupt.

Table 5–26 *ADSP-21160 Interrupt Configuration Registers*

Address	Register	Description
0x50	H1I0	Configures the direction of 21160-1 IRQ0
0x51	Unused	
0x52	H2I0	Configures the direction of 21160-2 IRQ0
0x53	Unused	
0x54	H3I0	Configures the direction of 21160-3 IRQ0
0x55	Unused	
0x56	H4I0	Configures the direction of 21160-4 IRQ0
0x57	Unused	
0x58	PCInt	Configures the direction of the PCI interrupt
0x59	Unused	
0x5A	PMCI0	Configures the direction of PMC+ IRQ0
0x5B	PMCI1	Configures the direction of PMC+ IRQ1

The registers are 32-bit registers. The first 16 bits (0–15) are read/write and select the source that will generate an interrupt to the processor. The second 16 bits (16–31) are read only and show which of the enabled interrupts are generating an interrupt. Bits 16–31 are masked interrupt lines and are masked by 21160-1 IRQ0's interrupt mask. Table 5–27 shows the bits included in each register.

Table 5–27 *Settings for the Interrupt Configuration Registers*

Bit	Name	Description*
B0	H1F0	The interrupt will respond to 21160-1 FLAG0
B1	H1F1	The interrupt will respond to 21160-1 FLAG1
B2	H2F0	The interrupt will respond to 21160-2 FLAG0
B3	H2F1	The interrupt will respond to 21160-2 FLAG1
B4	H3F0	The interrupt will respond to 21160-3 FLAG0
B5	H3F1	The interrupt will respond to 21160-3 FLAG1
B6	H4F0	The interrupt will respond to 21160-4 FLAG0
B7	H4F1	The interrupt will respond to 21160-4 FLAG1
B8	PCFlg	The interrupt will respond to a flag from the PCI interface
B9	PMCFlg0	The interrupt will respond to FLAG0 from the PMC interface
B10	PMCFlg1	The interrupt will respond to FLAG1 from the PMC interface
B11	PRFlg	The interrupt will respond to a flag from the peripheral bus
B12	UART0	The interrupt will respond to UART0
B13	UART1	The interrupt will respond to UART1
B14–15	Unused	
B16	H1F0	21160-1 FLAG0
B17	H1F1	21160-1 FLAG1
B18	H2F0	21160-2 FLAG0
B19	H2F1	21160-2 FLAG1
B20	H3F0	21160-3 FLAG0
B21	H3F1	21160-3 FLAG1
B22	H4F0	21160-4 FLAG0
B23	H4F1	21160-4 FLAG1
B24	PCFlg	PCI flag
B25	PMCFlg0	PMC+ FLAG0
B26	PMCFlg1	PMC+ FLAG1
B27	PRFlg	Peripheral bus flag
B28	UART0	UART0 flag
B29	UART1	UART1 flag
B30–31	Unused	

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

5.3.10 Reading the Status of All Flags and Interrupts

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 5–28 and Table 5–29 describe the bits in the registers.

Table 5–28 *Reading the Status of the Flags*

Bit	Name	Description*
B0	H1F0	Status of 21160-1 FLAG0
B1	H1F1	Status of 21160-1 FLAG1
B2	H2F0	Status of 21160-2 FLAG0
B3	H2F1	Status of 21160-2 FLAG1
B4	H3F0	Status of 21160-3 FLAG0
B5	H3F1	Status of 21160-3 FLAG1
B6	H4F0	Status of 21160-4 FLAG0
B7	H4F1	Status of 21160-4 FLAG1
B8	PCFlg	Status of PCI flag
B9	PMCFlg0	Status of PMC+ FLAG0
B10	PMCFlg1	Status of PMC+ FLAG1
B11	PRFlg	Status of peripheral bus flag
B12	UART0	Status of UART0 flag
B13	UART1	Status of UART1
B14–15	Unused	

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

Table 5–29 *Reading the Status of the Interrupts*

Bit	Name	Description*
B0	H1I0	Status of 21160-1 IRQ0
B1	Unused	
B2	H2I0	Status of 21160-2 IRQ0
B3	Unused	
B4	H3I0	Status of 21160-3 IRQ0
B5	Unused	
B6	H4I0	Status of 21160-4 IRQ0
B7	Unused	
B8	PCInt	Status of PCI interrupt
B9	PMCI0	Status of PMC+ IRQ0
B10	PMCI1	Status of PMC+ IRQ1
B11–15	Unused	

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

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 White Mountain DSP to the Hammerhead-PCI board. To attach an ICE to the Hammerhead-PCI, follow the steps below:

1. Connect the probe on the ICE card to the Hammerhead-PCI's JTAG connector.
2. If you are using an ISA bus ICE card, install it in an 8-bit slot in your PC; if you are using a PCI bus ICE card, install it in a 32-bit slot.
3. Install the Hammerhead-PCI in a 32-bit slot in your PC.
4. Apply power to the Hammerhead-PCI.
5. Start the emulator software on the PC.

A.1.1 Overview of the ICE Emulator

The Hammerhead-PCI is compatible with White Mountain DSP's ICE emulators, which are separate ISA bus or PCI bus cards[1] that connect to the Hammerhead-PCI's JTAG connector and install in an IBM PC's 8-bit (for ISA) or 32-bit (for PCI) expansion slot. 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.

1. ICE cards with ethernet or USB connections are also available from White Mountain DSP.

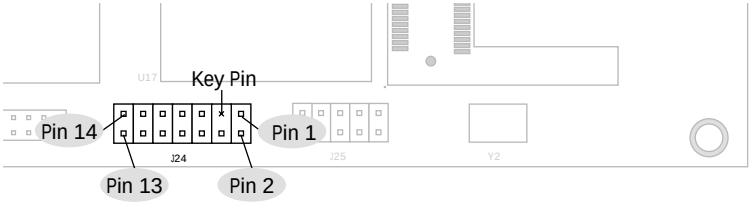
When the ICE is attached to the Hammerhead-PCI, the Hammerhead-PCI becomes the target system for the emulator, allowing you to operate it completely from the emulator's user interface. A powerful tool for debugging programs running on the ADSP-21160 processors, the emulator monitors system behavior while 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 Attaching the ICE to the Hammerhead-PCI

To attach the ICE to the Hammerhead-PCI, follow the instructions below.

1. Locate the JTAG connector (J24) on the Hammerhead-PCI (see Figure 3–2).
2. A cable extends from the ICE card to a probe that connects to the JTAG connector on the Hammerhead-PCI. 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 3–12 in Chapter 3 shows the connector pinout.

A.1.3 Installing the ICE and Hammerhead-PCI in a PC

Once you have connected the ICE to the Hammerhead-PCI, install the boards in your PC. The Hammerhead-PCI requires a 32- or 64-bit slot in your IBM-compatible PC. The ICE card requires either an 8-bit slot for an ISA bus card or a 32-bit slot for a PCI bus card. Section 2.5 explains how to install the Hammerhead-PCI, and the *ICE Hardware User's Guide* (from White Mountain DSP) explains how to install the ICE.

A.1.4 Operating the ICE

To start operating the ICE with the Hammerhead-PCI,

1. Apply power to the Hammerhead-PCI.

Note

As long as the emulator software is not running, you can safely attach and remove the ICE probe while the Hammerhead-PCI 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. BittWare's VisualDSP Target is a plug-in to ADI's 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.

Appendix B
Setting Up for Standalone Operation

The Hammerhead-PCI can boot via link ports or from a boot program stored in its FLASH memory (see section 2.9.1), which allows it to operate in standalone mode, free from a host computer. This section lists the steps necessary to prepare your Hammerhead-PCI to operate in standalone mode.

Note

If you are not planning to operate the Hammerhead-PCI 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 with the boot loader (see B.1.2).
3. Power down and set the boot mode jumpers to “Flash Boot” or “Link Boot” (see “Setting the Boot Mode for the Processors” on page 15).
4. Set the Standalone Mode jumpers (see “Setting the Standalone Operation Jumpers” on page 17).
5. Mount standoffs on the board (see B.5).
6. Apply power to the Hammerhead-PCI (see B.6).
7. Initialize the PCI interface (see B.7).
8. Boot the board in Flash or Link boot mode.

B.1 Developing and Loading a Flash 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-PCI 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.

B.1.2 Loading a Boot Program into the Flash Memory

Use programs provided with the Hammerhead-PCI to load the Flash memory with a boot program.

B.1.3 Testing the Flash

The `flash` directory contains utilities and a test program that uses the utilities. These programs provide easy access to the Flash memory on the Hammerhead-PCI board.

B.2 Setting the Boot Mode Jumpers

The Hammerhead-PCI has two configuration jumpers (JP1 and JP2) that configure its boot mode. See "Setting the Boot Mode for the Processors" on page 15 for instructions on setting the jumpers.

B.3 Setting the Standalone Mode Jumpers

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

JP16 Standalone primary PCI voltage
JP17 Standalone mode clock
JP18 Standalone mode reset

All three jumpers must be “ON” for the board to operate properly in standalone mode. See “Setting the Standalone Operation Jumpers” on page 17 for more detail.

B.4 Booting the Board via Link Port

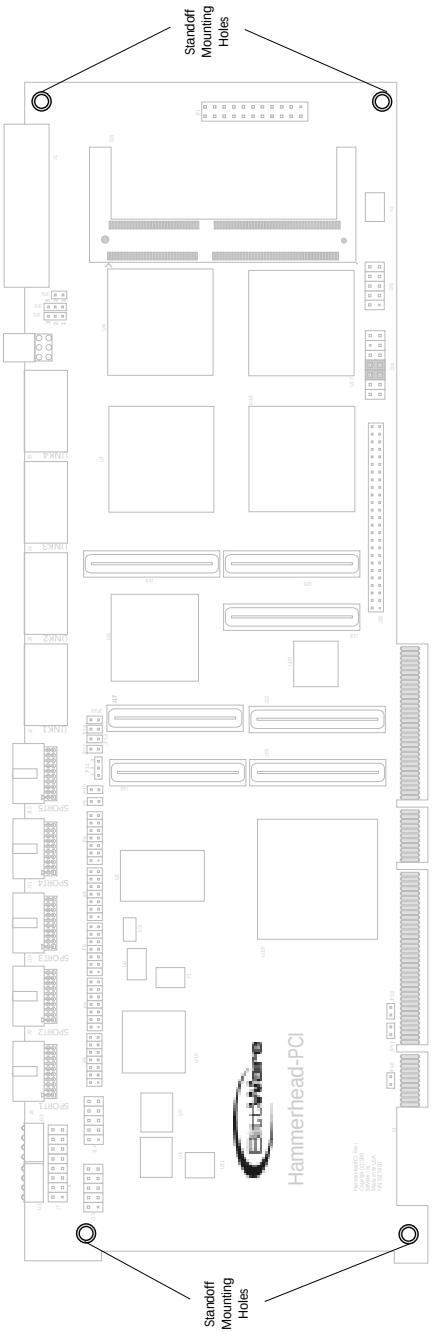
TBD

B.5 Mounting Standoffs on the Hammerhead-PCI

Mount standoffs on the Hammerhead-PCI to provide adequate clearance between the work surface and the Hammerhead-PCI’s components.

1. Place ¼" standoffs in the standoff mounting holes on the Hammerhead-PCI. Figure B–1 shows where the mounting holes are located.
2. Secure each standoff with a ¼" screw.

Figure B–1 *Location of the Standoff Mounting Holes*



B.6 Supplying Power to the Hammerhead-PCI

The Hammerhead-PCI 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 (J1) supplies +3.3V, +5V, –12V, and +12V to the Hammerhead-PCI. Figure 3–3 gives the pinout of the connector and shows where it is located.

To connect an external power source to the Hammerhead-PCI,

1. Plug a power adapter cable into the Hammerhead-PCI's external power connector (J1). Be sure to align pin 1 (+12V) on the cable with the +12V pin on J1.
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-PCI. Section 2.9 explains in more detail how to reset the board.

B.7 Initializing the PCI Interface

TBD

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`**.
- Try re-installing the tools and checking your path if you are getting "file not found" or similar errors.
- 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 with problems when making the call, we would better be able to start investigating the problem.

C.2 Contacting Technical Support

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

- Phone (9am – 6pm ET): (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

This appendix defines certain terms used throughout the manual.

ADSP-21160 cluster bus — The ADSP-21160 cluster bus is a 50 MHz, 64-bit bus that connects the four ADSP-21160 processors and a 128 MB bank of SDRAM. It is connected to the PCI interface through the Sharc FIN ASIC.

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-PCI 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.

ICE (in-circuit emulator) — The Hammerhead-PCI is compatible with White Mountain DSP's in-circuit emulators (ICE), which are separate ISA or PCI bus cards that connect to the Hammerhead-PCI's JTAG connector and install in an IBM PC's 8-bit (for ISA) or 32-bit (for PCI) expansion slot. 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 reigsters 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 Sharc FIN ASIC that cause an interrupt when written to. The processor that was interrupted can read the registers to find out about the interrupt.

MMS — (Multiprocessor Memory Space). 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 line 0) allows the DSPs to access the Hammerhead-PCI's SDRAM, which is located on the 64-bit ADSP-21160 cluster bus.

MS1 MS1 (memory select line 1) allows the DSPs to access the Hammerhead-PCI's 1 MB bank of FLASH memory, which is located on the 8-bit peripheral bus.

MS2 MS2 (memory select line 2) allows the DSPs to access the Sharc FIN ASIC.

MS3 MS3 (memory select line 3) allows the DSPs to access the dual UART, which is located on the 8-bit peripheral bus.

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

PCI-to-DSP bridge BittWare's Sharc FIN ASIC functions as a bridge (PCI-to-DSP) between the PCI interface and the ADSP-21160 DSPs, connecting the secondary PCI bus, the ADSP-21160 cluster bus, and the peripheral bus 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 Sharc FIN ASIC and three buses: the secondary PCI bus, the ADSP-21160 cluster bus, and the peripheral bus.

PCI-to-PCI bridge The PCI-to-PCI bridge is a chip manufactured by Intel (21154) that provides a bridge between the primary and secondary PCI buses.

PCI-to-PCI interface The PCI-to-PCI interface consists of the primary PCI bus and the PCI-to-PCI bridge chip.

Peripheral bus The 25 MHz, 8-bit peripheral bus connects to low-speed peripherals such as the FLASH RAM, the dual UART, and an expansion connector. 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.

Primary PCI bus The primary PCI bus is a 66 MHz, 64-bit bus between the host and the Intel 21154 PCI-to-PCI bridge.

Secondary PCI bus	The secondary PCI bus is a 66 MHz, 64-bit bus between the Intel 21154 PCI-to-PCI bridge and the Sharc FIN.
Sharc FIN ASIC	BittWare's Sharc FIN ASIC flexibly interfaces Analog Devices' SHARC DSPs to a wide range of the Hammerhead-PCI's interfaces, including: 64/66 MHz PCI bus (rev. 2.1 compliant), SDRAM, UART, I^2S serial ports, FLASH, and a general-purpose expansion bus (the 8-bit peripheral bus). The Sharc FIN 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.
SPORT	SHARC synchronous serial port
SYSCON register	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-PCI.