Hammerhead-PMC+

Quad ADSP-21160 64-bit 66 MHz PMC DSP 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-PMC+ Reference Manual

Hardware Revision 1

Printed in the USA
October 23, 2001 Edition UG-HHPM01–00

Contents

Chapter 3 Overview of the Hardware Components

Chapter 4 Hammerhead-PMC+ Board Architecture

Chapter 5 Programming Details for the ADSP-21160s

Chapter 6 SharcFIN Programming Details

Figures

Tables

1

Chapter 1

Introduction

BittWare's Hammerhead-PMC+ board packs the processing power of four ADSP-21160 SHARC® DSPs and the speed of a 64-bit, 66 MHz PCI interface on a PMC+ board. The board supports up to 64–512 MB of SDRAM, and a 2MB bank of Flash memory. The board also features BittWare's SharcFIN™ ASIC, which flexibly interfaces the ADSP-21160s to the 64-bit, 66 MHz PCI interface, the SDRAM, the Flash memory, and an I/O expansion bus.

This chapter covers the following topics:

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

1.1 Overview of the Hammerhead-PMC+

This section gives a brief overview of the architecture of the Hammerhead-PMC+ board and describes its software.

1.1.1 Hammerhead-PMC+ Features

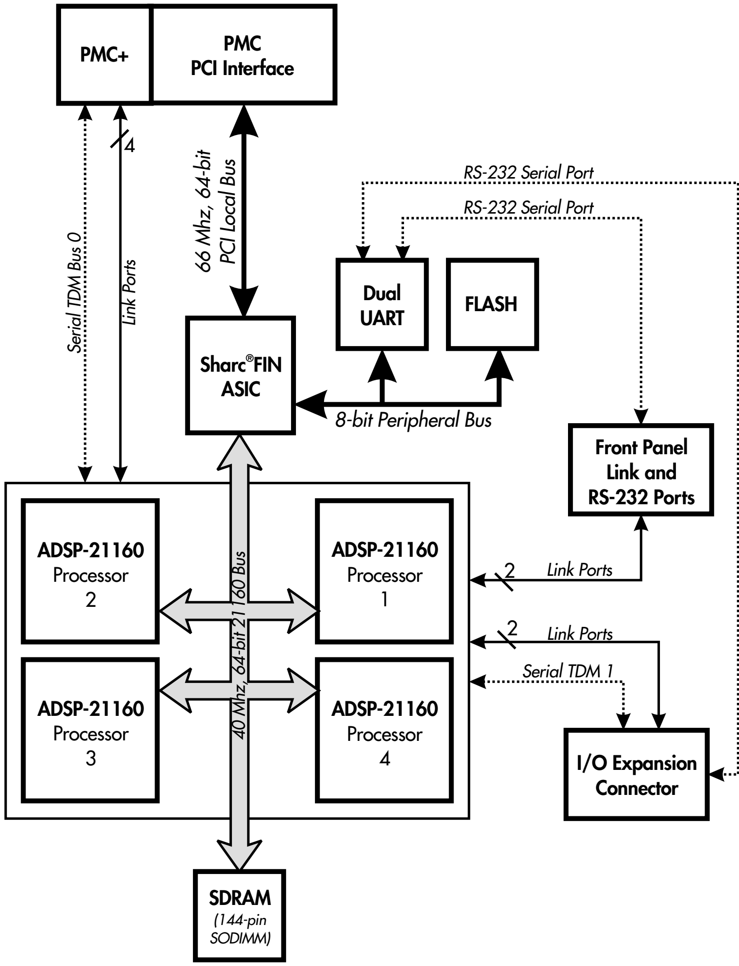
The Hammerhead-PMC+ features:

- Four 80 MHz ADSP-21160 SHARC processors
- 64-bit, 66 MHz PCI interface
- 64 to 512 MB SDRAM (standard 144-pin SODIMM)
- Two 80 MB/s link ports and one 40 Mb/s serial port) extend to front panel I/O and optional I/O expansion cards
- Two link ports and one serial TDM port extend to the PMC+ interface
- RS-232 UART
- PMC (PCI Mezzanine Card) form factor
- 2 MB Flash memory

1.1.2 Hammerhead-PMC+ System Architecture

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

Figure 1–1 *Block Diagram of the Hammerhead-PMC+*



BittWare's Hammerhead-PMC+ board features four ADSP-21160 SHARC DSPs and a 64-bit, 66 MHz PCI interface. It also features the BittWare SharcFIN ASIC, up to 512 MB of SDRAM, a 2 MB bank of Flash memory, and an I/O expansion site.

SharcFIN ASIC

The Hammerhead-PMC+ incorporates a BittWare SharcFIN ASIC to flexibly interface the ADSP-21160 DSPs to the 64-bit, 66 MHz PCI bus, the SDRAM, and a peripheral bus that interfaces to the board's UART and 2 MB bank of Flash memory. The SharcFIN also provides a feature-rich set of DMA functions and interrupt options to support very high-speed, real-time data flow with minimum processor overhead.

ADSP-21160 DSPs

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

I/O Options

In addition to the PMC+ interface, the Hammerhead-PMC+ has several other options for I/O: external link ports, an RS-232 port, and an I/O expansion connector. Two on-board link ports extend from the ADSP-21160s to front panel I/O. The RS-232 port allows the DSPs to communicate with external serial devices, facilitating remote debugging, command, and control. The I/O expansion connector allows you to attach an optional BittWare expansion card, adding up to two link ports or a serial port and an RS-232 port.

1.1.3 Hammerhead-PMC+ Software Architecture

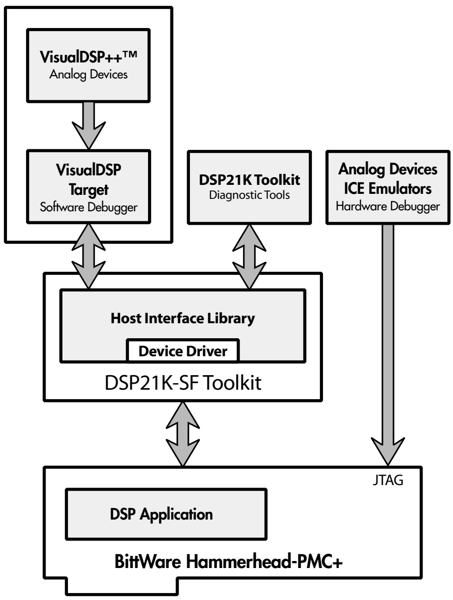
You will need three types of software development tools for the Hammerhead-PMC+: code development tools, debugging tools, and host interface tools. Figure 1–2 is a general block diagram of how the software development tools work together with the Hammerhead-PMC+.

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

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

BittWare's DSP21k-SF Toolkit provides your host interface tools. The DSP21k-SF Toolkit is a complete software development kit that allows you to easily develop application code and integrate the Hammerhead-PMC+ 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 Software Architecture*



1.2 About This User's Guide

1.2.1 Purpose of this Document

This user's guide covers hardware revision 1 of the Hammerhead-PMC+ (Hammerhead-PMC+) board, which supports four ADSP-21160 SHARC processors operating at 80 MHz. The purpose of this document is to provide details about the Hammerhead-PMC+'s major hardware components, to describe how to install and properly operate the Hammerhead-PMC+, 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 Used Throughout This User's Guide

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

- 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-PMC+ for Operation
This chapter describes the tasks that you must perform to prepare the 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-PMC+'s major components and connectors and briefly discusses their function.

Chapter Four: Hammerhead-PMC+ Board Architecture

This chapter discusses the board's architecture, including link ports, serial ports, flags and interrupts, and bus interfaces. It also discusses the architecture of the SharcFIN ASIC.

Chapter Five: Programming Details for the ADSP-21160s

This chapter provides programming details for the DSPs, including how to access the four different types of memory and boot the board by various methods.

Chapter Six: SharcFIN Programming Details

This chapter describes the function of the SharcFIN ASIC. It details the SharcFIN's PCI and SHARC interfaces, describes the memory maps, and explains how to set the SharcFIN's user-configurable registers.

Appendix A: Debugging Your DSP Programs

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

Appendix B: Setting Up the Board 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.

1.3 Other Helpful Documents and Tools

This section gives sources for additional information that applies to the Hammerhead-PMC+, and it lists several third party software development tools that you may find useful.

1.3.1 Documents for Further Reference

The documents in the list below provide additional information about the Hammerhead-PMC+ components and software.

- *ADSP-21160 SHARC User's Guide* – Analog Devices, Inc.
- *Intel 21154 Chip Data Sheet* – Intel
- *SharcFIN ASIC User's Guide* – BittWare, Inc.
- *DSP21k-SF Toolkit User's Guide* (Version 6.0 and greater) – BittWare, Inc.
- *I/O Expansion Card Technical Data Sheet* – BittWare, Inc.

1.3.2 Software DevelopmentTools

VisualDSP++® and BittWare VisualDSP Target

The Hammerhead-PMC+ 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.

Analog Devices In-Circuit Emulators

The ICE in-circuit emulators from Analog Devices provide real-time hardware emulation and debugging. Analog Devices 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-PMC+, 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-PMC+ 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-PMC+ 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* and greater*) User's Guide* from BittWare, Inc. contains complete information about the DSP21k-SF Toolkit.

SpeedDSP Optimized Libraries for SHARC DSPs

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

BittWare's SharcLAB MATLAB Interface

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

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

Chapter 2

Preparing the Hammerhead-PMC+ for Operation

This chapter describes the tasks necessary to prepare the board for installation, such as setting configuration jumpers, connecting external devices, installing the board, and installing development software. It also discusses different methods of resetting the board and how to test it using BittWare's software tools. After reading this chapter, you should be able to:

- Set the board's configuration jumpers
- Mount the Hammerhead-PMC+ on a PMC or PMC+ capable carrier board
- Attach any desired external signals to the board
- Run diagnostic tests on the board to ensure that it is operating properly
- Reset the board by various methods

2.1 Unpacking the Hammerhead-PMC+

Warning!

The Hammerhead-PMC+ 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-PMC+ 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 module from the plastic bag, observing 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.

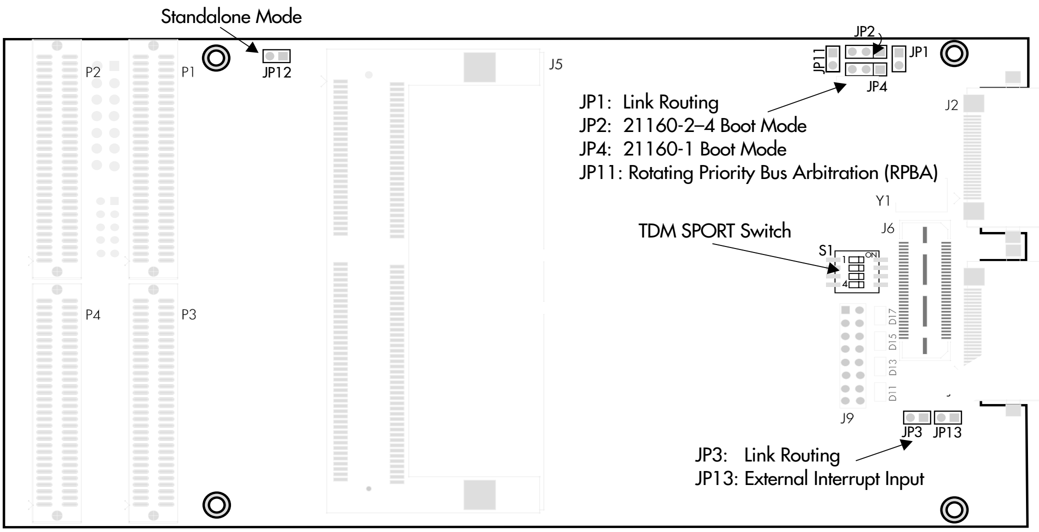
2.2 Configuring the Hammerhead-PMC+

This section describes how to set up the physical features of the Hammerhead-PMC+ board. It gives the location and description of the board's configuration jumpers and explains how to connect external devices to the Hammerhead-PMC+.

2.2.1 Setting the Hammerhead-PMC+ Configuration Jumpers

The Hammerhead-PMC+ has seven configuration jumpers that allow you to control and enable several of the board's features. Before mounting the Hammerhead-PMC+ on a host board, 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-PMC+ Configuration Jumpers (Bottom)*



Setting the Link Routing

Jumpers JP1 and JP3 provide routing for external link ports J2 and J7. The default setting for these jumpers is OUT. Do not change the setting of these jumpers; doing so could damage the board.

Setting the Boot Mode for the Processors

JP2 and JP4 configure the boot mode for the ADSP-21160 processors. The processors can boot in three boot modes:

- host booting
- link booting
- Flash booting

JP4 configures the boot mode for 21160-1, and JP2 configures the boot mode for 21160 2–4.Table 2–1 shows their settings.

Table 2–1 *Settings for the Boot Mode Selection Jumpers (JP2 and JP4)*

Jumper	Jumper Position	Setting	Default
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	
JP4	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	

Setting the Standalone Operation Jumper

Jumper JP12 configures the Hammerhead-PMC+ to operate in standalone mode. The default position for this jumper is OUT, which is the setting for normal operation. When the jumper is IN, the board will operate in standalone mode. For details about setting the board up for standalone operation, refer to Appendix B .

Using the External Interrupt Input

The external interrupt input connector, JP13, is used to send external interrupts directly to the SharcFIN. Do not use a jumper on JP13 as it is a two-pin connector. See section 3.2.10 for details about using JP13.

2.2.2 Configuring the TDM Serial Ports

The TDM serial bus provides a communication route between the ADSP-21160s and synchronous serial devices. One serial bus connects to the expansion card connector (J6), and one connects to the PMC+ interface. The Hammerhead-PMC+ has one TDM configuration switch, S1, that allows you to configure the signals of the external TDM serial ports.

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

Serial Port	Connections
TDM SPORT 0	SPORT0 from 21160-1–4 have TDM serial connection to PMC+ interface A
TDM SPORT 1	SPORT1 from 21160-1–4 have TDM serial connection to rear panel I/O (P3)

Table 2–3 gives the pinout of the switch, and Table 2–4 shows its settings.

Table 2–3 *TDM Serial Port Switch Pinout (S1)*

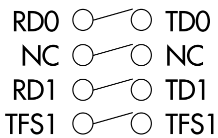
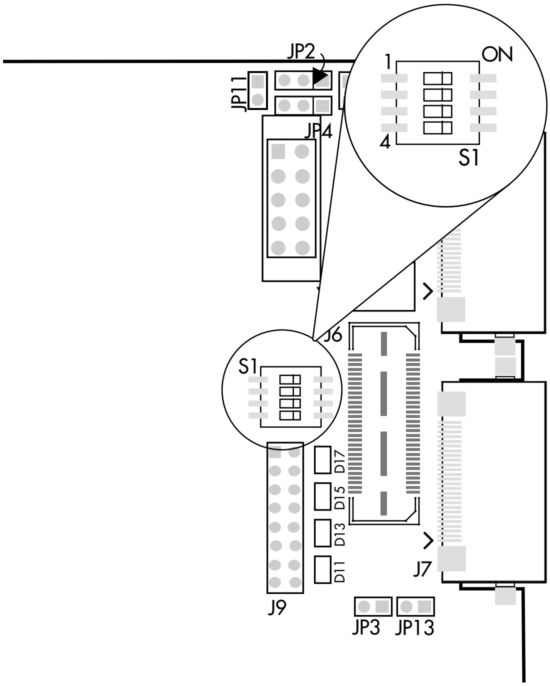


Table 2–4 *TDM Serial Port Switch Settings (S1)*

Switch	Settings
TD/RD	ON: TDM 1-wire OFF: TDM 2-wire, standard mode
TFS	ON: Standard with null-modem cable OFF: Otherwise

Figure 2–2 shows where the TDM serial port configuration switch is located on the Hammerhead-PMC+ board.

Figure 2–2 *Location of the TDM Serial Port Configuration Switch (Bottom)*



2.2.3 Connecting External Devices to the Hammerhead-PMC+

This section explains how to attach external devices to the Hammerhead-PMC+'s external interfaces.

Connecting Link Port Cables to the Front Panel

Four link ports, one per DSP, are available externally on the Hammerhead-PMC+, allowing the DSPs to communicate directly with DSPs on other boards. Link ports 1 and 2 are brought to front panel external connectors J2 and J7, and link ports 3 and 4 are routed to expansion I/O connector J6, where they may be broken out to an optional BittWare expansion I/O card. Section 3.2.2 provides details about the external link connectors, and section 3.2.6 gives an overview of the expansion I/O connector.

BittWare offers link port cables for the front panel connectors; Table 2–5 below gives the part numbers of the link port cables that are compatible with the Hammerhead-PMC+'s front panel connectors.

Table 2–5 *External Link Port Cables Available From BittWare*

Location	Type	Manufacturer	Part Number
Hammerhead-PMC+ (J2, J7)	Coax Ribbon, 12" or 36"	BittWare	CAHC-L26-12 (12") CAHC-L26-36 (36")

Connecting an External Reset Signal to the Reset Connectors

The external reset connector (J4) allows the Hammerhead-PMC+ board to reset or be reset by other system boards. The connector supports an input reset line to allow the Hammerhead-PMC+ to receive reset signals from other boards; it also supports an output reset line to allow the Hammerhead-PMC+ to reset other boards.

To reset the Hammerhead-PMC+ with the external reset connector,

1. Connect a cable (see Figure 3–7 on page 44) from the external reset connector (J4) on the Hammerhead-PMC+ board to another system board.
2. Any reset that occurs on the Hammerhead-PMC+ reset source causes a reset on all Hammerhead-PMC+ reset targets.

Connecting an External Power Supply to the Power Connector

The Hammerhead-PMC+ requires a +3.3 and +5V power supply for normal operation. The board's external power connector allows you to connect an external power source for operating the board in standalone mode. The external power connector (J10) supplies +3.3V, +5V, and GND to the Hammerhead-PMC+, but due to height restrictions dictated by the PMC specification, this site may not be populated with a connector. Figure 3–11 on page 50 gives the pinout of the connector and shows where it is located.

To connect an external power source to the Hammerhead-PMC+,

1. Plug a power adapter cable into the Hammerhead-PMC+'s external power connector (J10). Be sure to align pin 1 (GND) on J10 with the GND pin on the cable.
2. Connect the remaining end of the cable to an external power source, such as a switching standalone power supply or the PC's power supply.
3. Apply power to the system.
4. Reset the Hammerhead-PMC+. Section 2.4 explains in more detail how to reset the board.

Connecting an ICE Emulator to the JTAG Header

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

1. Connect the ICE to the Hammerhead-PMC+.
2. Depending on the form factor of the ICE card you are using, either install it in or connect it to your PC.
3. Apply power to the Hammerhead-PMC+
4. Start the emulator software on the PC.

2.3 Installing the Board and Its Software

This section describes how to mount the Hammerhead-PMC+ on a host board and how to install the Analog Devices and BittWare software development tools to interface with the hardware.

2.3.1 Mounting the Board on a PMC or PMC+ Carrier Board

The standard set-up for the Hammerhead-PMC+ is mounted on a PMC or PMC+ capable host board. Its PMC interface features three standard PMC connectors (P1–P3), which provide the 64-bit, 66 MHz PCI interface. An additional connector, the PMC+ connector (P4), supports 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 3.2.11) for our PMC+ extensions. If you are mounting the Hammerhead-PMC+ card on a host board that is not from BittWare and uses the P4 connector, the host board may have incompatibilities with the Hammerhead-PMC+'s PMC+ (P4) connector. Call BittWare technical support for assistance.

To attach the Hammerhead-PMC+ to a PMC or PMC+ host board,

1. Plug the Hammerhead-PMC+'s PMC connectors into the PMC interface on the carrier board, and press the connectors firmly together.
2. Secure the Hammerhead-PMC+ to the mounting holes on the carrier board.

Note

The PMC interface is auto-compatible with 32/33, 32/66, 64/33, and 64/66 PCI interfaces.

Table 2–6 shows the different options for accessing the host via the PMC interface.

Table 2–6 *PMC Interface Access Options*

PMC Connectors	Host Access
P1, P2	32-bit PCI
P1, P2, P3	64-bit PCI
P1, P2, P4	32-bit PCI and PMC+ extensions
P1, P2, P3, P4	64-bit PCI and PMC+ extensions

2.3.2 Installing the Analog Devices DevelopmentTools

The Hammerhead-PMC+ 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-PMC+. This section describes where to find installation instructions for the VisualDSP++ IDE and the BittWare VisualDSP Target.

Installing VisualDSP++

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

Installing BittWare's VisualDSP Target

If you will be using the VisualDSP Target debugger with the Hammerhead-PMC+, 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-PMC+. The *VisualDSP Target User's Guide* gives detailed installation instructions. The Hammerhead-PMC+ is compatible with versions 2.0 and greater of the VisualDSP Target.

2.3.3 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.*

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-PMC+ more quickly and easily. It contains a host interface library of C-callable functions for PC-based programs, diagnostic utilities, demo programs, and a DSP library that provides standard I/O extensions to DSP programs.

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

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

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

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

Installing the DSP21k-SF 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.

Verifying that the Board is Configured Properly

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

2.4 Resetting the Board

This section describes the four methods you can use to reset the Hammerhead-PMC+:

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

2.4.1 Resetting the Board via the PMC+ Interface

The PMC+ interface provides a reset line from the host board to the Hammerhead-PMC+ board. If the Hammerhead-PMC+ is mounted on a BittWare PMC+ capable carrier board, you can reset the Hammerhead-PMC+ via that reset line. Refer to the manual for the BittWare host board for details on issuing a PMC+ reset signal.

2.4.2 Resetting the Board via the Host PCI Interface

A register bit in the SharcFIN ASIC allows you to perform a total reset on the Hammerhead-PMC+. The bit is B0 of the register located at byte offset 0xE8 from the base of Base Address Register 0 (BAR0). For details on this register, refer to the SharcFIN ASIC user's manual.

2.4.3 Resetting the Board via the Watchdog Timer

The Hammerhead-PMC+'s watchdog timer helps to ensure that the Hammerhead-PMC+ 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 SharcFIN 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 (see Table 6–12).

How the Watchdog Timer Functions when Disabled

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

How the Watchdog Timer Functions when Enabled

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

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

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

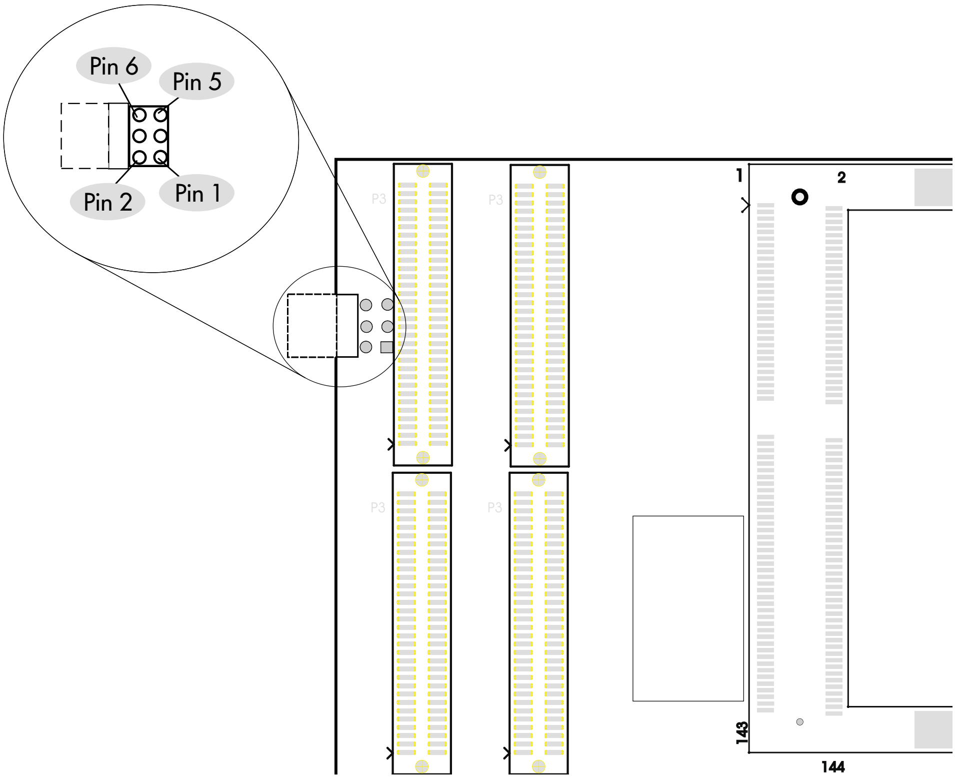
2.4.4 Resetting the Hammerhead-PMC+ with the External Reset Connector

The external reset connector (J4) allows the Hammerhead-PMC+ board to reset or be reset by other system boards. The connector supports an input reset line to allow the Hammerhead-PMC+ to receive reset signals from other boards; it also supports an output reset line to allow the Hammerhead-PMC+ to reset other boards.

To reset the Hammerhead-PMC+ with the external reset connector,

1. Connect a cable (see Figure 3–7 on page 44) from the external reset connector (J4) on the Hammerhead-PMC+ board to another system board.
2. Any reset that occurs on the Hammerhead-PMC+ reset source causes a reset on all Hammerhead-PMC+ reset targets. Figure 2–3 illustrates the signals for the reset source and the reset targets.

Figure 2–3 *External Reset Connector Signal Diagram*



2.5 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 describes two DSP21k-SF Toolkit test utilities and discusses the contents of the example files included with the Hammerhead-PMC+.

2.5.1 Testing the Board with 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 User's Guide* explains DspBad commands in more detail.

1 The processor number is the *device number * 10 + id number*. Refer to the *DSP21k-SF User's Guide* for further explanation.

Testing the Board with Diag21k

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

Step 1: Start Diag21k

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

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

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

   Both of the command line options above tell diag21k to open the first processor on device 0.

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

```
DSP21K Interactive Diagnostic Utility
32-bit version for BittWare boards under Windows 95/98 and
Windows NT/2000. Release 6.30 [ DSP21K-SF, Aug  7 2001
08:24:36 ], Version 4.30 Copyright (c) 1992-2001 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

a. Use the board information command to display information about the Hammerhead-PMC+'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-PMC+'s Memory

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

a. To ensure that none of the processors is executing programs that might change memory while you are testing it, use the following command to reset the board:

diag21k[1]>br

```
Board reset
```

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

diag21k[1]>pc

```
        processor configured
```

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

diag21k[1]>mt aa

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

Step 4: Load and Execute a Program

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

a. The `dsp21ksf\etc` directory contains an example program that calculates the first twenty prime numbers. The source code is in the `examples\Hammerhead-PMC+\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
```

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

diag21k[1]>ps

```
processor running
```

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

diag21k[0]>mr li _primes 20

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

Step 5: Test the Remaining Processors

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>
```

2.5.2 Testing the Board with the Hammerhead-PMC+ Example Files

The example software provided with the Hammerhead-PMC+ 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-PMC+ CD-ROM.

3

Chapter 3
Overview of the Hardware Components

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

- SharcFIN ASIC
- ADSP-21160 DSPs
- Flash memory
- SDRAM
- On-board oscillators
- RS-232 UART
- Watchdog timer
- LEDs
- JTAG header
- External power connector
- External reset connector
- Flag I/O connector
- External interrupt input
- Configuration jumpers
- Serial port configuration switch
- I/O expansion connector

3.1 Layout and Function of the Major Components

This section briefly describes the function of each major component on the board and shows where each is located. Figure 3–1 shows the components on the top side of the board, and Figure 3–2 shows the components on the bottom side of the board.

Figure 3–1 *Location of the Hammerhead-PMC+'s Major Components (Top)*

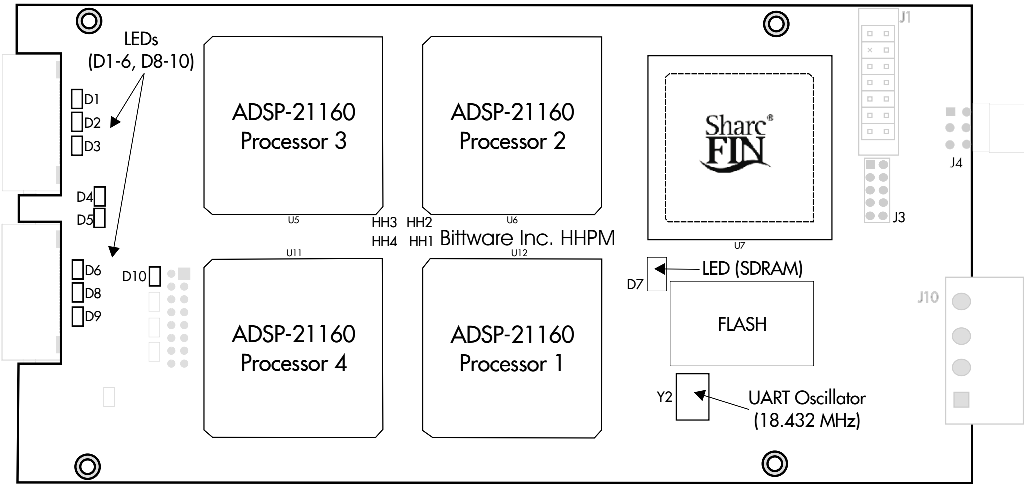
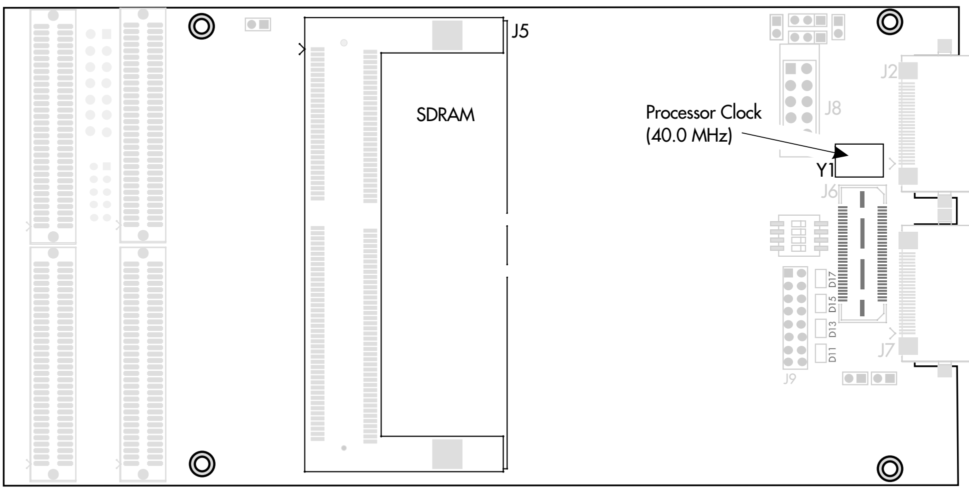


Figure 3–2 *Location of the Hammerhead-PMC+'s Major Components (Bottom)*



3.1.1 ADSP-21160 SHARC Processors

The Hammerhead-PMC+ features four ADSP-21160 SHARC processors from Analog Devices. The Hammerhead-PMC+'s processors have a total of 2400 MFLOPs of processing power and operate at 80 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 SharcFIN ASIC

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

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

3.1.3 Memory

Flash Memory

A 2 MB bank of Flash memory stores boot programs that the processors can load, enabling the Hammerhead-PMC+ 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 it as non-volatile storage space.

SDRAM

The Hammerhead-PMC+ has a standard 144-pin SODIMM for adding a 64, 128, 256, or 512 MB SDRAM module to the board for banked external memory. The SDRAM is available to the ADSP-21160 DSPs at 40 MHz via the ADSP-21160 cluster bus.

3.1.4 On-board Oscillators

The Hammerhead-PMC+ has two on-board oscillators: one for the DSP cluster, and one for the RS-232 UART.

ADSP-21160 Oscillator

A 40 MHz system oscillator chip (Y1) provides the 1× clock for the four ADSP-21160 DSPs on the board. Figure 3–2 shows where the oscillator is located.

UART Oscillator

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

3.1.5 Dual RS-232 UART

The Hammerhead-PMC+ features a dual RS-232 UART that interfaces serial data from the RS-232 ports to the ADSP-21160 DSPs. Figure 3–1 shows where the UART is located, and section 3.2.7 describes the RS-232 connector in more detail.

3.1.6 LEDs

The Hammerhead-PMC+ has nine user LEDs, which you can use to indicate certain conditions in the software or to provide feedback. One LED is connected to each ADSP-21160, each LED corresponding to a different ADSP-21160 flag (see Table 3–1 below). Section 4.2.5 describes the LEDs' connections to the ADSP-21160s.

Table 3–1 *LED Connections*

LED	Connection
D1	21160-4 F3
D2	21160-4 F2
D3	21160-3 F3
D4	21160-3 F2
D5	21160-2 F3
D6	21160-2 F2
D7	SDRAM
D8	21160-1 F3
D9	21160-1 F2

3.2 Layout and Function of the External Connectors

This section briefly describes the function of the external connectors on the board and shows where they are located (see Figure 3–3 and Figure 3–4). It also provides the pinouts for the connectors.

Figure 3–3 *Layout of the Hammerhead-PMC+'s External Connectors (Top)*

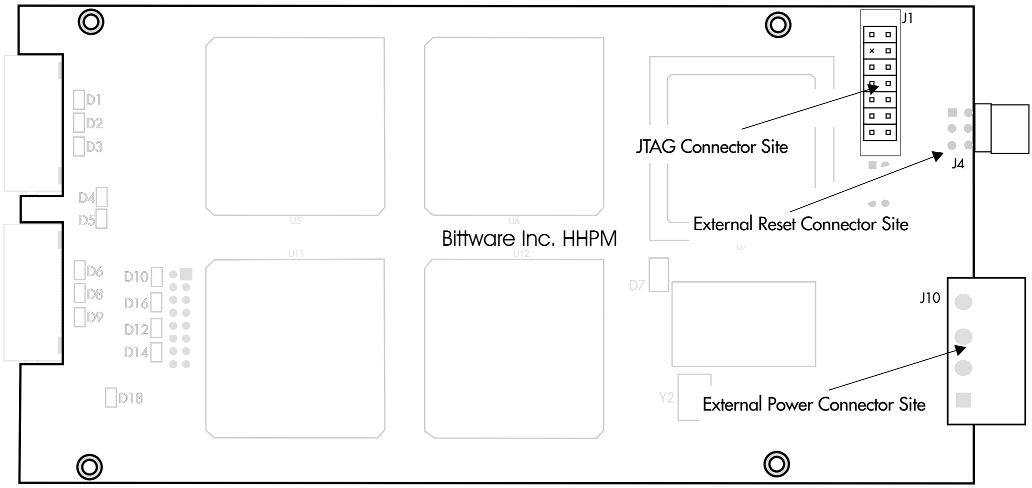


Figure 3–4 *Layout of the Hammerhead-PMC+'s External Connectors (Bottom)*

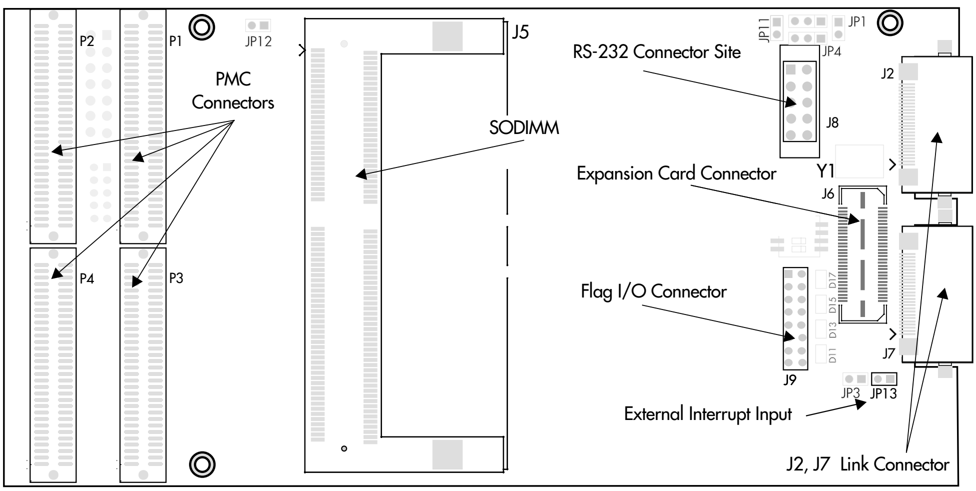


Table 3–2 *Overview of the External Connectors*

Connector	Ref Des	Type	Description
JTAG Header	J1	14-pin	Connection for in-circuit emulator (ICE)
Link Port	J2	10-pin	External link port connector
Test Header	J3	N/A	BittWare use only
External Reset	J4	6-pin	Connection for external reset signals
SODIMM	J5	144-pin	Connection for standard 144-pin SODIMM SDRAM module
I/O Expansion Header	J6	64-pin	Connection for optional BittWare I/O expansion daughter cards
Link Port	J7	10-pin	External link port connector
RS-232 Port	J8	10-pin	External RS-232 serial interface via the UART
Flag I/O Connector	J9	16-pin	Connection for direct signal input/output to the ADSP-21160s' flags
External Power	J10	4-pin	Connections for 3.3V, +5V, and GND external power supply
External Interrupt Input	JP13	2-pin	Provides a connection for an external interrupt into the SharcFIN

3.2.1 JTAG Header

The JTAG header (J1) allows in-circuit emulation with an optional ICE emulator manufactured by Analog Devices. All four ADSP-21160 DSPs are connected to the JTAG connector. Figure 3–5 shows the pins on the JTAG header, and Table 3–3 gives the connector pinout. Appendix A explains how to connect an emulator to the Hammerhead-PMC+.

Figure 3–5 *Location of the JTAG Header Pins (Top)*

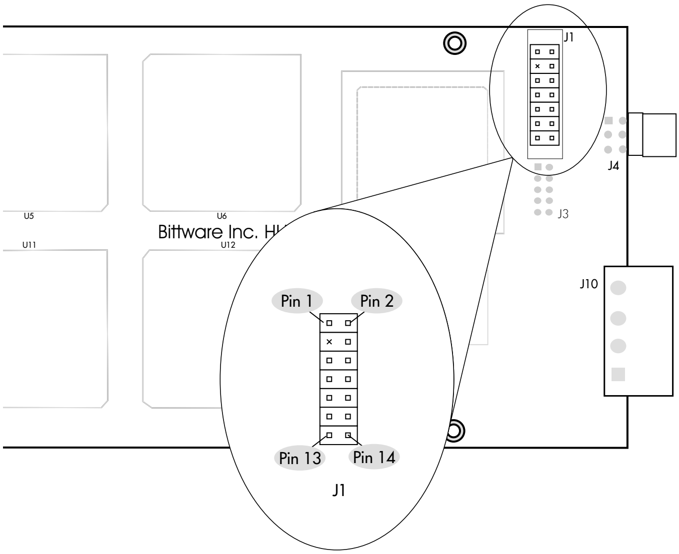


Table 3–3 *JTAG Header Pinout*

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

3.2.2 External Link Ports

The Hammerhead-PMC+ has two external link ports running at 80 MB/s that are accessible from the front panel. Figure 3–6 shows the location of the pins on the external link ports (J2 and J7), and Table 3–4 gives the connectors' pinouts. Section 2.2.3 describes how to attach external signals to the connectors.

Figure 3–6 *Location of the External Link Port Connector Pins (Bottom)*

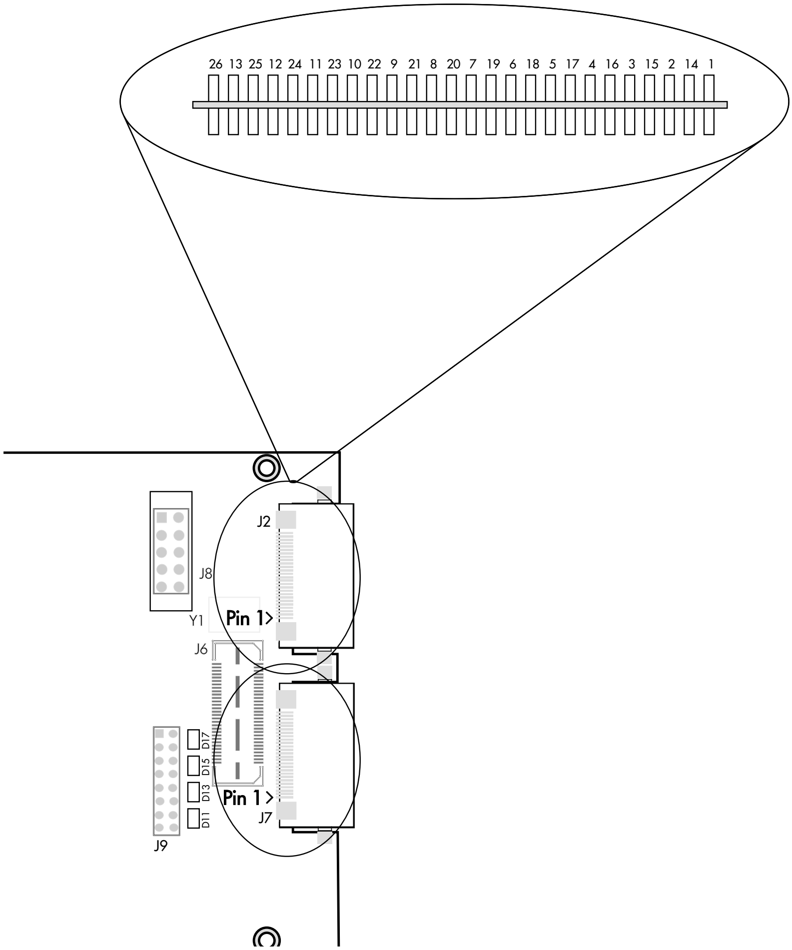


Table 3–4 *External Link Port Connector Pinouts for J2 and J7*

J2		J7	
Pin	**Description**	**Pin**	**Description**
1	USRDEF1	1	USRDEF1
2	LCLK	2	LCLK
3	LACK	3	LACK
4	LDAT[0]	4	LDAT[0]
5	LDAT[1]	5	LDAT[1]
6	LDAT[2]	6	LDAT[2]
7	LDAT[3]	7	LDAT[3]
8	LDAT[4]	8	LDAT[4]
9	LDAT[5]	9	LDAT[5]
10	LDAT[6]	10	LDAT[6]
11	LDAT[7]	11	LDAT[7]
12	NC	12	NC
13	NC	13	NC
14	GND	14	GND
15	GND	15	GND
16	GND	16	GND
17	GND	17	GND
18	GND	18	GND
19	GND	19	GND
20	GND	20	GND
21	GND	21	GND
22	GND	22	GND
23	GND	23	GND
24	GND	24	GND
25	GND	25	GND
26	JP1	26	JP3

3.2.3 External Reset Connector

The Hammerhead-PMC+'s reset connector (J4) allows one Hammerhead-PMC+ board to reset other Hammerhead-PMC+ boards in the same system. 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 in the cluster. The input signal is pulled up with a 10K resistor. If the output signal is driven low, the board will output a reset signal to other boards. When the output is connected to group reset, it can drive a reset signal to up to 250 boards. If the output signal is tied to the board's hardware reset line, it is driven low by either a host board reset or by a watchdog reset.

The section entitled "Resetting the Hammerhead-PMC+ with the External Reset Connector" on page 23 explains how to use the external reset signals, and section 2.4 explains the board's reset events. Figure 3–7 shows where the pins are located, and Table 3–6 gives the connector pinout.

Figure 3–7 *Location of the External Reset Connector Pins (Top)*

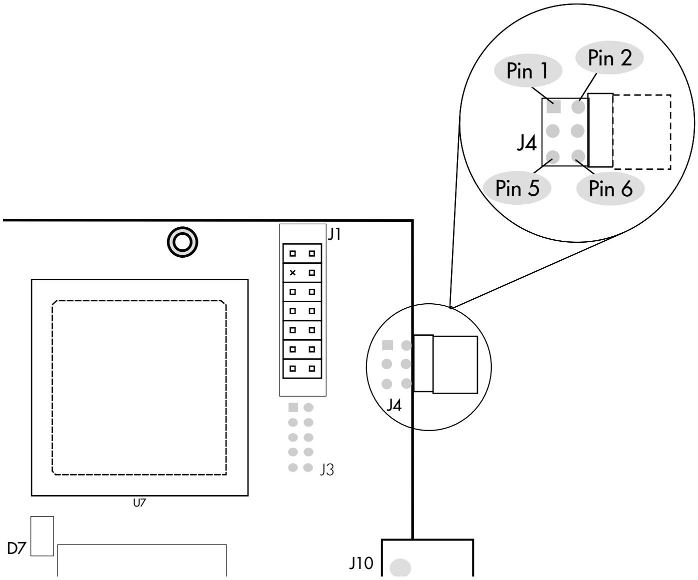


Table 3–5 *External Reset Connector Pinout*

Pin	Signal	Pin	Signal
1	NC	2	$\overline{\text{HA\_EXTGRPRSTIN}}$
3	NC	4	GND
5	GND	6	$\overline{\text{HA\_EXTGRPRSTOUT}}$

3.2.4 SODIMM Connector

The Hammerhead-PMC+ has an industry-standard 144-pin connection for a standard SODIMM module (J5). The SODIMMs are available in 64, 128, 256, and 512 MB modules. Figure 3–4 on page 38 shows the location of the SODIMM connector pins.

3.2.5 External Serial Ports

The serial ports on the four ADSP-21160 processors are routed to TDM serial bus 0 and TDM serial bus 1. Serial port 0 on each DSP connects to TDM serial bus 0, which is brought out to the PMC+ connector (P4). Serial port 1 on each DSP connects to TDM serial bus 1, which is routed to the expansion I/O connector and may be accessed via an optional expansion card. For more information on the expansion card, including serial port connector pinouts, refer to the *I/O Expansion Card Technical Data Sheet* (BittWare, Inc.).

3.2.6 I/O Expansion Connector

The Hammerhead-PMC+ features a 64-pin connector (J6) that allows you to attach one of BittWare's optional expansion cards. The cards are available in two configurations. One features two link port connectors, and the other features an external serial port connector and an RS-232 connector. Figure 3–8 shows the location of the expansion connector pins, and Table 3–6 gives their pinout. The *I/O Expansion Card Technical Data Sheet* (BittWare, Inc.) describes the expansion cards and explains how to attach one to the Hammerhead-PMC+.

Figure 3–8 *Location of the I/O Expansion Connector Pins (Bottom)*

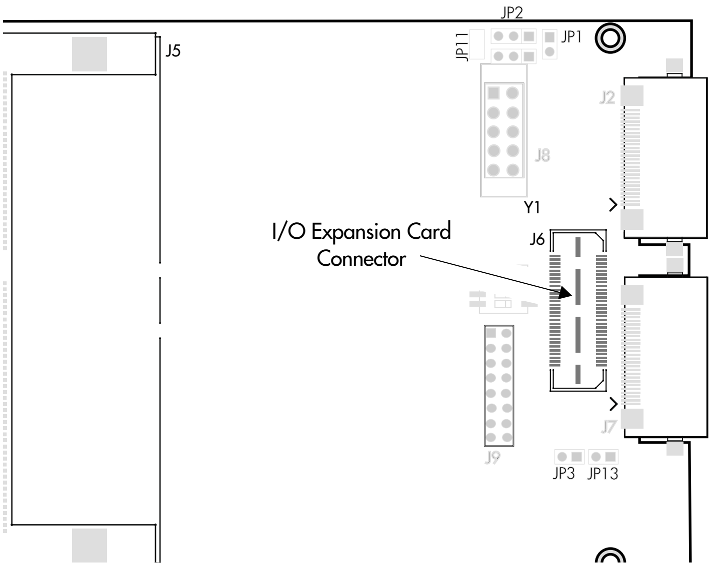


Table 3–6 *I/O Expansion Connector Pinout*

Pin	Signal	Pin	Signal
1	VDD	2	VDD
3	GND	4	GND
5	EA_L4CLK	6	EA-L3CLK
7	GND	8	GND
9	EA_L4ACK	10	EA_L3ACK
11	GND	12	GND
13	EA_L4DAT0	14	EA_L3DAT0
15	GND	16	GND
17	EA_L4DAT1	18	EA_L3DAT1
19	GND	20	GND
21	EA_L4DAT2	21	EA_L3DAT2
23	GND	24	GND
25	EA_L4DAT3	26	EA_L3DAT3
27	GND	28	GND
29	EA_L4DAT4	30	EA_L3DAT4
31	GND	32	GND
33	EA_L4DAT5	34	EA_L3DAT5
35	GND	36	GND
37	EA_L4DAT6	38	EA_L3DAT6
39	GND	40	GND
41	EA_L4DAT7	42	EA_L3DAT7
43	GND	44	GND
45	XHA_TDMTFS1	46	HA_TDMTD1
47	GND	48	GND
49	HA_TDMTRC1	50	XA_TDMRD1
51	GND	52	GND
53	HA_TDMRFS1	54	HA_TXD1
55	GND	56	HA_RXD1
57	$\overline{\text{HA\_CTS1}}$	58	$\overline{\text{HA\_RTS1}}$
59	P33V	60	P33V
61	GND	62	GND
63	GND	64	GND

3.2.7 RS-232 Connector

The Hammerhead-PMC+ is configured with a dual UART, which transports serial data between the host and the board's ADSP-21160 processors. The SharcFIN ASIC provides a memory-mapped register set that allows the DSP cluster to interface to a PC via its RS-232 port. There are two UART channels on the Hammerhead-PMC+: one channel is routed to the board's RS-232 connector site (J8), but because of height restrictions dictated by the PMC specification, this site is not populated with a connector except by special order. The other channel can be accessed via an optional BittWare expansion I/O card, which is available with an RS-232 connector. Figure 3–9 shows where the connector pad is located on the Hammerhead-PMC+, and the *I/O Expansion Card Technical Data Sheet* (BittWare, Inc.) gives the location and pinout of the expansion card's RS-232 connector.

Figure 3–9 *Location of the RS-232 Connector Site (Bottom)*

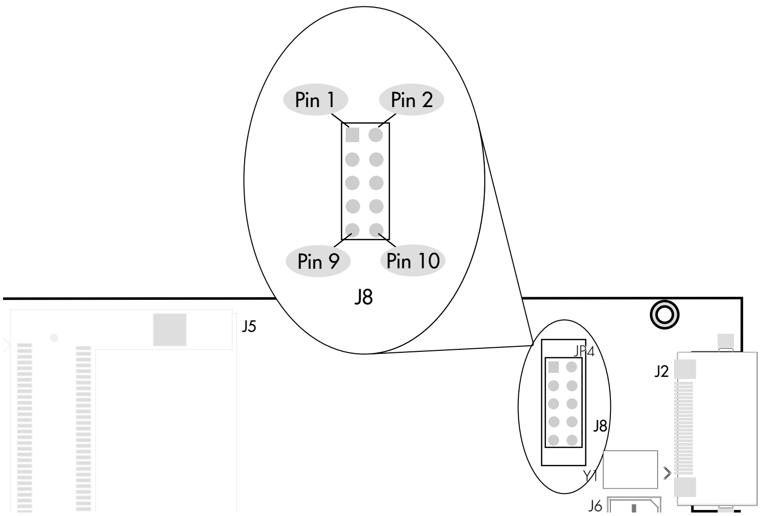


Table 3–7 *RS-232 Connector Pinout*

Pin	Signal	Pin	Signal
1	–	2	–
3	HA_TXD0	4	HX_CTS0
5	HA_RXD0	6	HA_RTS0
7	–	8	NC
9	GND	10	NC

3.2.8 Flag I/O Connector

The 16-pin flag I/O site (J9) allows access to the FLAG2 and FLAG3 signals on each ADSP-21160 DSP. This access to the ADSP-21160s' flags allows you to input and output signals directly to the processors. Because the connector is directly routed to the ADSP-21160s, the circuit is diode protected to GND and 3.3V to shield the processors from voltage overload. Figure 3–10 shows where the connector is located, and Table 3–8 gives the connector signals.

Figure 3–10 *Location of the Flag I/O Connector (Bottom)*

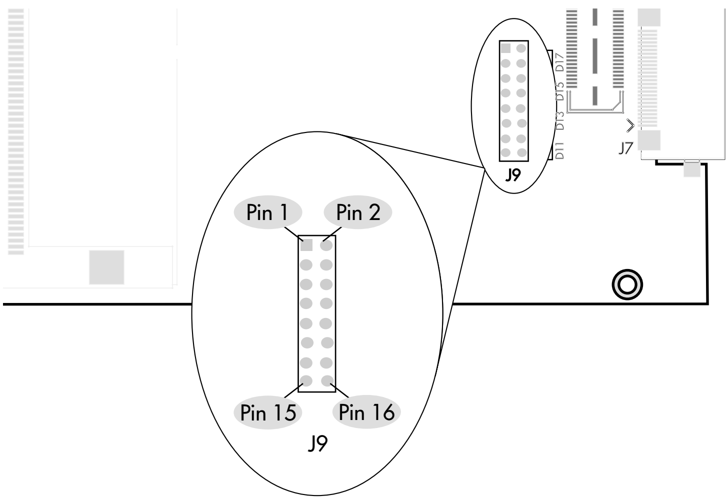


Table 3–8 *Flag I/O Pinout (J9)*

Pin	Signal	Description	Pin	Signal	Description
1	HA1_F2	21160-1 FLAG2	2	GND	Ground
3	HA1_F3	21160-1 FLAG3	4	GND	Ground
5	HA2_F2	21160-2 FLAG2	6	GND	Ground
7	HA2_F3	21160-2 FLAG3	8	GND	Ground
9	HA3_F2	21160-3 FLAG2	10	GND	Ground
11	HA3_F3	21160-3 FLAG3	12	GND	Ground
13	HA4_F2	21160-4 FLAG2	14	GND	Ground
15	HA4_F3	21160-4 FLAG3	16	GND	Ground

3.2.9 External Power Connector

The Hammerhead-PMC+ has an external power connector (J10) 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, and GND to the Hammerhead-PMC+. Figure 3–11 shows the location of the pins on the external power connector (J10), andTable 3–9 gives the connector pinout. Section 2.2.3 explains how to connect an external power supply to the connector.

Figure 3–11 *Location of the External Power Connector Pins (Top)*

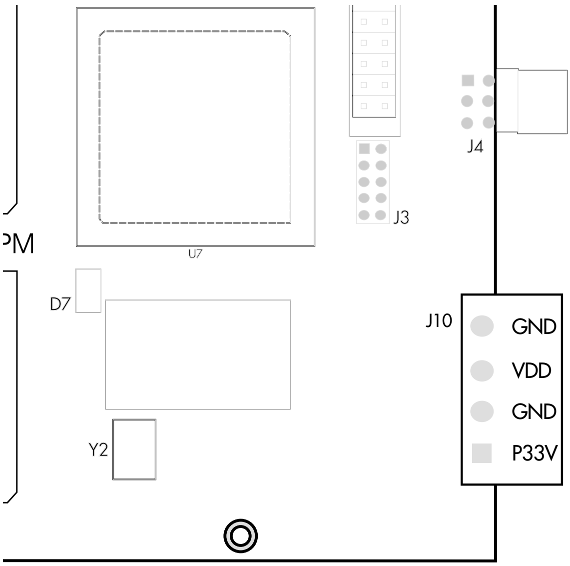


Table 3–9 *External Power Connector Pinout*

Pin	Signal
1	GND
2	VDD
3	GND
4	P33V

3.2.10 External Interrupt Input Connector

Connector JP13 provides a direct path for external interrupts to the SharcFIN. To input your external interrupts, attach a ground wire to pin 2 of JP13 and a signal wire to pin 1. Since the connector is directly routed to the SharcFIN, the circuit is diode protected to GND and 3.3V to shield the ASIC from voltage overload.

3.2.11 PMC+ Interface

The PMC+ interface allows you to mount the Hammerhead-PMC+ on a standard PMC carrier board or on one of BittWare's PMC+ carrier boards. It consists of four 64-pin connectors. Three connectors are standard 64-pin PMC connectors that provide the 64-bit, 66 MHz PCI interface. The fourth connector is a 64-pin PMC+ connector that provides BittWare's PMC+ I/O extensions, which include a serial TDM bus, four link ports, an I^2C interface, and a reset line. Table 3– 2 gives the pinout for the PMC+ interface, and Figure 3–4 shows where the connectors are located.

Warning!

BittWare uses P4[a] of the PMC connectors for our PMC+ extensions. If you are mounting the Hammerhead-PMC+ card on a host board that is not from BittWare and uses the P4 connector, the host board may have incompatibilities with the Hammerhead-PMC+'s PMC+ (P4) connector. Call BittWare technical support for assistance.

a P4 is the fourth (user-definable) PMC connector as defined in the *IEEE P1386.1 Standard Physical and Environmental Layers for PCI Mezzanine Cards: PMC* (PMC specification). It is referred to as Pn4/Jn4 in that document.

Note

The PMC+ extensions are available only when the Hammerhead-PMC+ is attached to a BittWare PMC+ carrier board.

Table 3–10 *PMC+ Interface Pinout (P1-P4)*

P1 (PMC)				P2 (PMC)			
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

P3 (PMC)

Pin	Signal	Pin	Signal
1	PCI-RSVD	2	GND
3	GND	4	$\overline{C/BE[7]}$
5	$\overline{C/BE[6]}$	6	$\overline{C/BE[5]}$
7	$\overline{C/BE[4]}$	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

P4 (PMC+)

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

3.3 Layout and Function of the Configuration Jumpers and Switches

This section shows where the Hammerhead-PMC+'s configuration switch and jumpers are located (Figure 3–12) and gives a short description of each (Table 3–11).

Figure 3–12 *Location of the Configuration Jumpers and Switch (Bottom)*

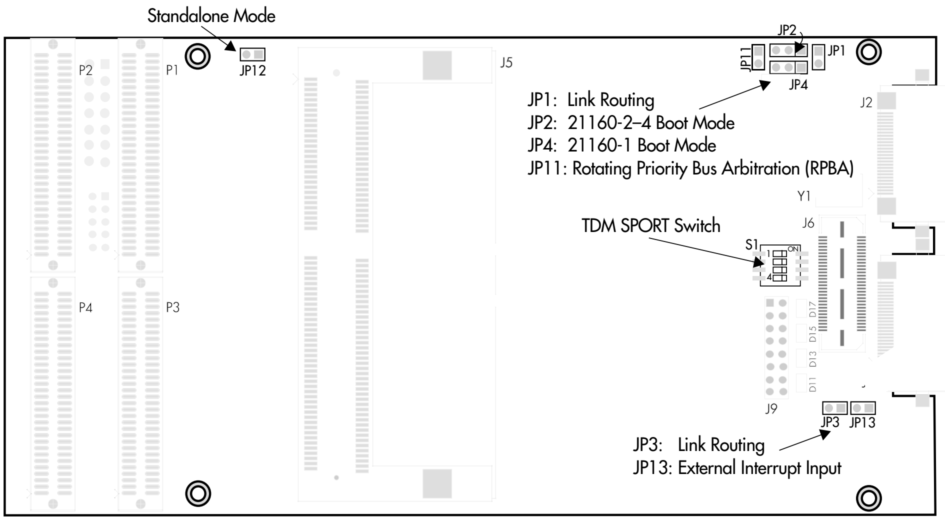


Table 3–11 *Overview of the Configuration Jumpers*

Jumper	Name	Description
JP1	External Link Connector Routing	Custom usage only. Please contact BittWare for details before changing setting.
JP2	21160-2-4 Boot Mode	Sets 21160-2–4 to boot from host computer, on-board Flash, or remote processor via link port
JP3	External Link Connector Routing	Custom usage only. Please contact BittWare for details before changing setting.
JP4	21160-1 Boot Mode	Sets 21160-1 to boot from host computer, on-board Flash, or remote processor via link port
JP11	21160 Bus Priority	Selects "fixed" or "rotating" bus priority scheme for ADSP-21160 processors
JP12	Standalone Mode	Configures the SharcFIN to operate in stand-alone mode, or independently of the PCI bus
JP13[a]	External Interrupt Input	Provides a connection for an external interrupt into the SharcFIN

a JP13 is actually a connector and is described in section 3.2.8.

Table 3–12 *Overview of Configuration Switch S1*

Switch	Name	Description
S1	TDM Switch	Configures the signals of the external TDM port

Chapter 4

Hammerhead-PMC+ Board Architecture

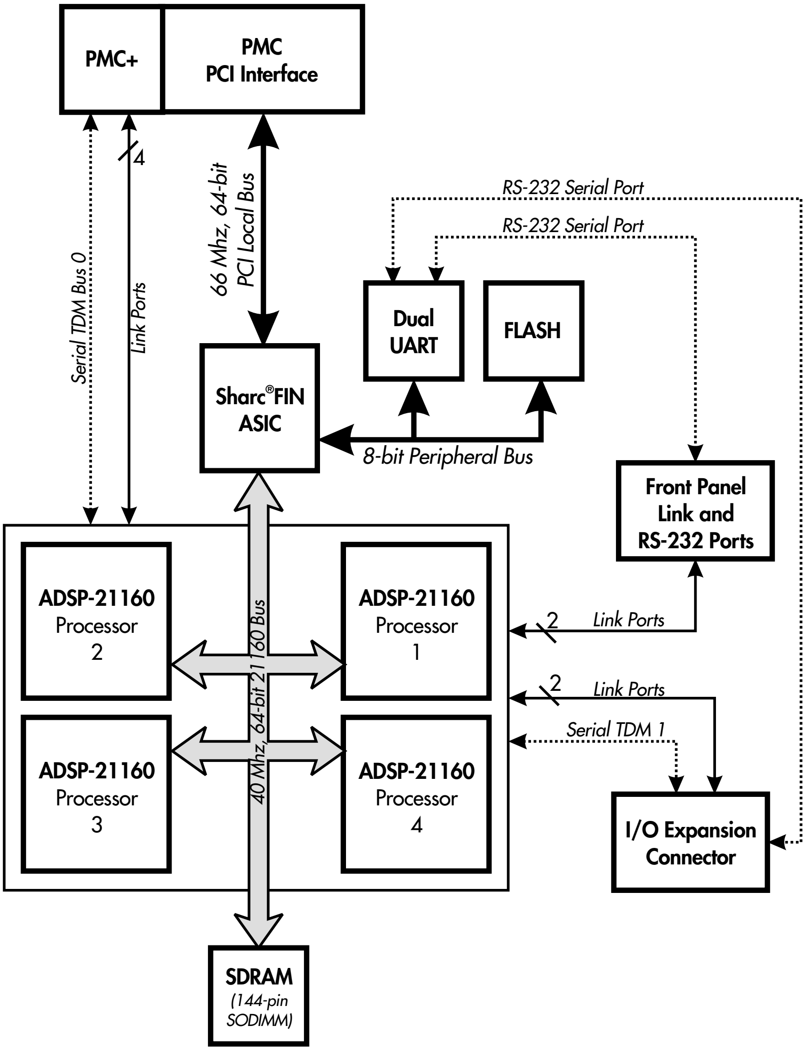
This chapter discusses the architecture of the board and describes how the ADSP-21160 DSPs communicate with other DSPs, with the host, and with other I/O peripherals on and off the board. This chapter covers the following topics:

- How the DSPs access internal and external memory
- The connections of the DSPs' serial ports
- The connections of the DSPs' link ports
- The connections of the DSPs' flags and interrupts
- The connections to the DSPs' 64-bit cluster bus
- The structure of the PCI interface (including the SharcFIN ASIC, PCI bus, and peripheral bus)
- The connections available via the PMC+ interface

4.1 Overview of the Board Architecture

This section briefly describes how data flows through the Hammerhead-PMC+ board. The sections that follow discuss the board's architecture in more detail.

Figure 4–1 *Block Diagram of the Hammerhead-PMC+ System Architecture*



The Hammerhead-PMC+ features four ADSP-21160 DSPs that support a 64-bit 66 MHz PCI bus, a 64-bit 40 MHz ADSP-21160 cluster bus, a bank of up to 512 MB of SDRAM, a 2 MB bank of Flash memory, a dual UART, a PMC+ interface, and a SharcFIN ASIC.

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

For I/O options, the board features external link port connectors, an RS-232 port, a PMC interface, and an expansion card connector. Two external link ports are available on the front panel. The RS-232 port connects to the UART to allow serial communication with the ADSP-21160 DSPs. The PMC+ interface extends off of the PCI bus, providing access to standard PMC carrier boards or to BittWare's PMC+ carrier boards. When the Hammerhead-PMC+ is attached to a BittWare PMC+ capable carrier board, the PMC+ interface provides two link ports, a TDM serial bus, an I^2C interface, interrupts, and a reset line directly to the DSPs. The I/O expansion connector allows you to attach one of BittWare's optional expansion cards, which adds either two link ports or a serial port and an RS-232 port to the Hammerhead-PMC+.

4.2 ADSP-21160 Architecture

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

4.2.1 Resources Available to the ADSP-21160s

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

Table 4–1 *Resources for 21160-1 (Quad-Processor Configuration)*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM Bus 0 (PMC+)	SharcFIN[a]	SharcFIN	External J7
1	Flash, UART	TDM Bus 1 (J6)[b]	21160-3 F1[c] 21160-4 F1	SharcFIN	21160-4 L2
2	SharcFIN		21160-2 F3 Flag I/O (J9)	LED D9 Flag I/O (J9)	21160-2 L4
3				21160-2 I2 LED D8 Flag I/O (J9)	21160-2 L5
4					PMC+ L1
5					21160-4 L3

a IRQ0, Flag0, and Flag1 on each processor connect to the SharcFIN interrupt multiplexer. From the SharcFIN, you can route them to different locations on the board.

b This serial port connection is available only if an I/O expansion card configured with link ports is attached to the Hammerhead-PMC+.

c IRQ1 on each DSP is hardwired within the SharcFIN.

Table 4–2 *Resources for 21160-2 (Quad-Processor Configuration)*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM Bus 0 (PMC+)	SharcFIN[a]	SharcFIN	PMC+ L2
1	Flash, UART	TDM Bus 1 (J6)[b]	21160-3 F1[c] 21160-4 F1	SharcFIN	External J2
2	SharcFIN		21160-1 F3 Flag I/O (J9)	LED D6 Flag I/O (J9)	21160-3 L4
3				21160-1 I2 LED D5 Flag I/O (J9)	21160-3 L5
4					21160-1 L2
5					21160-1 L3

a IRQ0, Flag0, and Flag1 on each processor connect to the SharcFIN interrupt multiplexer. From the SharcFIN, you can route them to different locations on the board.

b This serial port connection is available only if an I/O expansion card configured with link ports is attached to the Hammerhead-PMC+.

c IRQ1 on each DSP is hardwired within the SharcFIN.

Table 4–3 *Resources for 21160-3 (Quad-Processor Configuration)*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM Bus 0 (PMC+)	SharcFIN[a]	SharcFIN	PMC+ L3
1	Flash, UART	TDM Bus 1 (J6)[b]	21160-1 F1[c] 21160-2 F1	SharcFIN	External J6[†]
2	SharcFIN		21160-4 F3 Flag I/O (J9)	LED D4 Flag I/O (J9)	21160-4 L4
3				21160-4 I2 LED D3 Flag I/O (J9)	21160-4 L5
4					21160-2 L2
5					21160-2 L3

a IRQ0, Flag0, and Flag1 on each processor connect to the SharcFIN interrupt multiplexer. From the SharcFIN, you can route them to different locations on the board.

b This serial port connection is available only if an I/O expansion card configured with link ports is attached to the Hammerhead-PMC+.

c IRQ1 on each DSP is hardwired within the SharcFIN.

Table 4–4 *Resources for 21160-4 (Quad-Processor Configuration)*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM Bus 0 (PMC+)	SharcFIN[a]	SharcFIN	PMC+ L4
1	Flash, UART	TDM Bus 1 (J6)[b]	21160-1 F1[c] 21160-2 F1	SharcFIN	External J6[†]
2	SharcFIN		21160-3 F3 Flag I/O (J9)	LED D2 Flag I/O (J9)	21160-1 L1
3				21160-3 I2 LED D1 Flag I/O (J9)	21160-1 L5
4					21160-3 L2
5					21160-3 L3

a IRQ0, Flag0, and Flag1 on each processor connect to the SharcFIN interrupt multiplexer. From the SharcFIN, you can route them to different locations on the board.

b This serial port connection is available only if an I/O expansion card configured with link ports is attached to the Hammerhead-PMC+.

c IRQ1 on each DSP is hardwired within the SharcFIN.

4.2.2 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.

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 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-PMC+ has one cluster of four DSPs; all four DSPs in the cluster share a common bus (the ADSP-21160 cluster bus), and each can view the on-chip SRAM of the other three DSPs in its cluster.

External Memory

External memory space consists of other devices that share the ADSP-21160's 64-bit cluster bus. The external memory space is divided into four banked sections of memory.

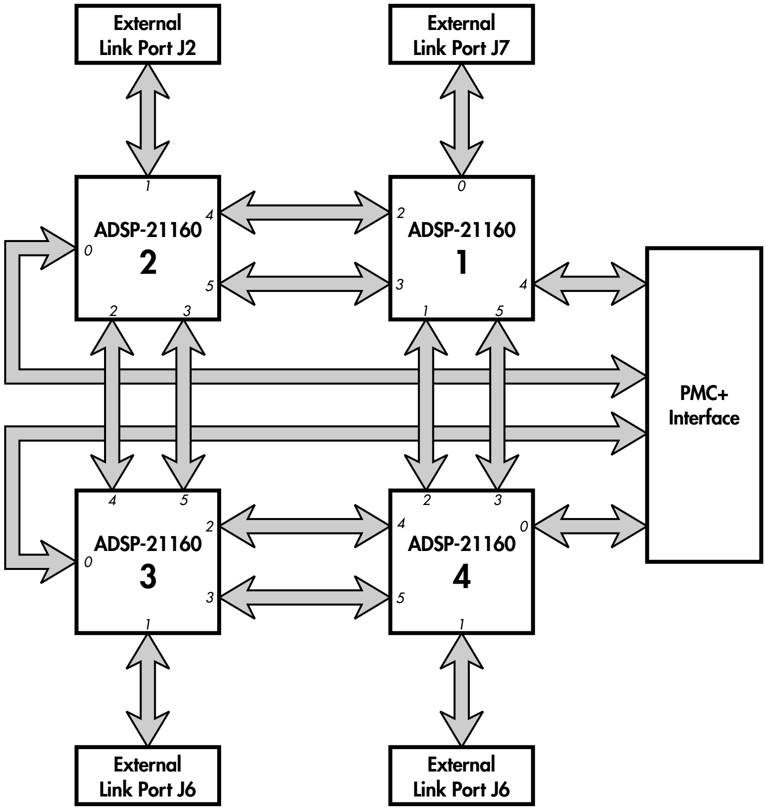
Each ADSP-21160 DSP has four memory select lines, MS0–MS3, which allow it to access the external memory banks located on the 64-bit ADSP-21160 cluster bus and on the 8-bit peripheral bus. Refer to section 5.1 for more details on the ADSP-21160 memory configuration.

4.2.3 Link Port Connections

Each ADSP-21160 DSP has six 8-bit, 80 Mbyte/s link ports to allow high-speed communication between the on-board DSPs and DSPs on other boards. The link ports are also available for link booting. Figure shows the link port connections on the Hammerhead-PMC+.

- Four link ports on each processor are dedicated for interprocessor communication.
- One link port from each processor extends to the PMC+ interface.
- One link port from 21160-3 and 21160-4 extend to an optional external link port connector; the processors can boot from a remote processor via these link ports (see section 5.2.1).

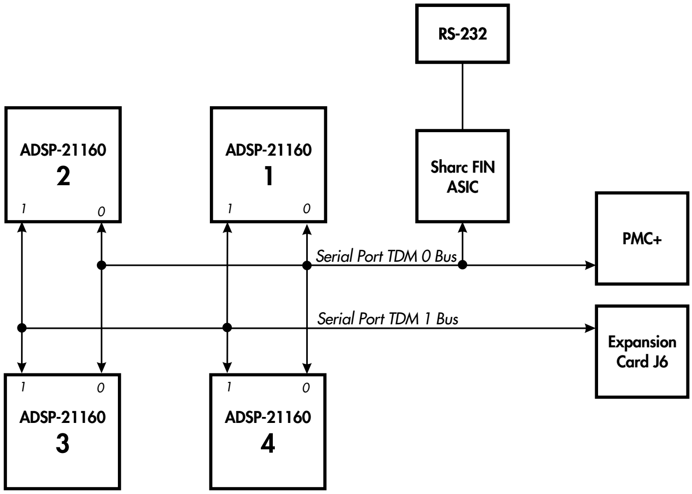
Figure 4–2 Block Diagram of the Link Port Connections



4.2.4 Serial Port Connections

Each ADSP-21160 DSP has two 40 Mbit/s serial ports, both of which connect to external TDM buses. Serial port 0 from each DSP connects to the PMC+ interface, and serial port 1 from each DSP connects to the expansion card connector on the front panel. Figure 4–3 illustrates the serial port connections for the Hammerhead-PMC+.

Figure 4–3 *Block Diagram of the Serial Port Connections*



Serial Port Connections to Serial TDM Bus 0

Serial port 0 from each processor is connected to the serial port TDM 0 bus, which is brought out to the PMC+ connector.

Serial Port Connections to Serial TDM Bus 1

Serial port 1 from each processor is connected to the serial port TDM 1 bus, which is brought out to the expansion card connector. Refer to section 3.2.6 for details on the expansion card configuration options.

4.2.5 Flag and Interrupt Connections

Each ADSP-21160 DSP has four flags and three interrupts. Two flags and one interrupt from each DSP connect to the SharcFIN ASIC; using registers in the SharcFIN, you can configure the routing of those flags and interrupts. The remaining flags from each DSP connect to the other DSPs in the cluster, to LEDs and to a flag I/O connector. The remaining interrupts connect via the SharcFIN to the other DSPs in the cluster. Figure 4–4 illustrates the flag and interrupt connections on the Hammerhead-PMC+. Table 4–5 gives the flag connections, and Table 4–6 lists the interrupt connections for the DSPs.

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

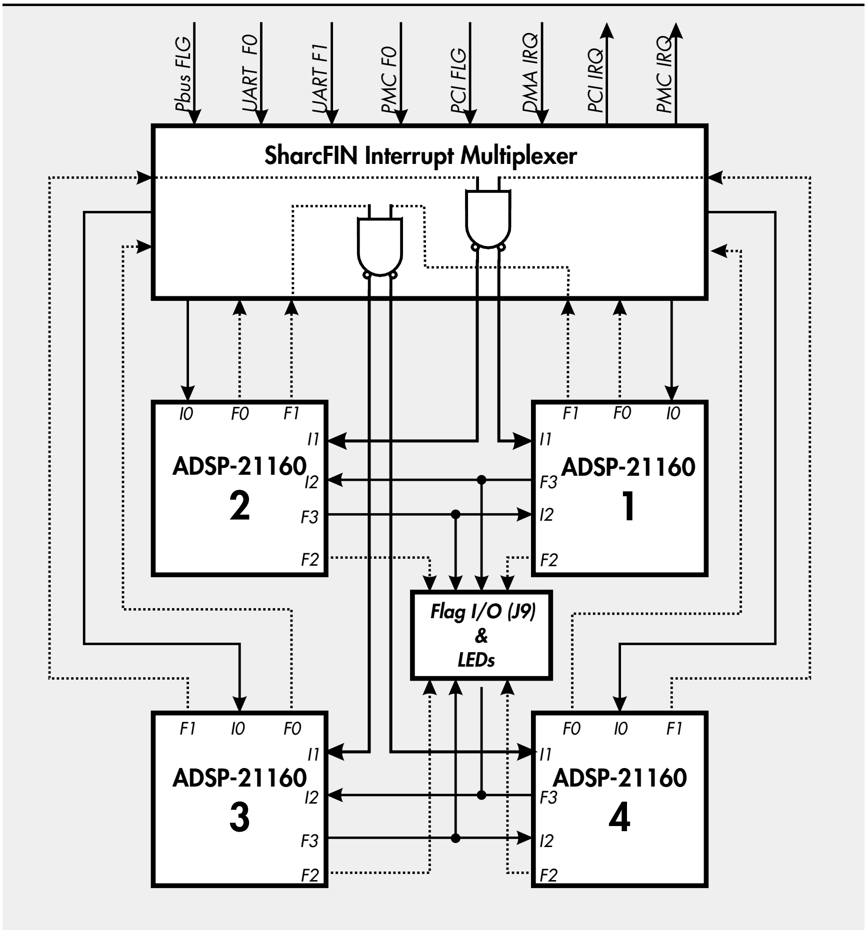


Table 4–5 *ADSP-21160 Flag Connections*

	21160-1	21160-2	21160-3	21160-4
Flag0	SharcFIN	SharcFIN	SharcFIN	SharcFIN
Flag1	SharcFIN	SharcFIN	SharcFIN	SharcFIN
Flag2	LED D9 Flag I/O J9	LED D6 Flag I/O J9	LED D4 Flag I/O J9	LED D2 Flag I/O J9
Flag3	21160-2 I2 LED D8 Flag I/O J9	21160-1 I2 LED D5 Flag I/O J9	21160–4 I2 LED D3 Flag I/O J9	21160-3 I2 LED D1 Flag I/O J9

Table 4–6 *ADSP-21160 Interrupt Connections*

	21160-1	21160-2	21160-3	21160-4
IRQ0	SharcFIN	SharcFIN	SharcFIN	SharcFIN
IRQ1[a]	21160-3 F1 21160-4 F1	21160-3 F1 21160-4 F1	21160-1 F1 21160-2 F1	21160-1 F1 21160-2 F2
IRQ2	21160-2 F3 Flag I/O J9	21160-1 F3 Flag I/O J9	21160-4 F3 Flag I/O J9	21160-3 F3 Flag I/O J9

a IRQ1 is hardwired within the SharcFIN.

4.2.6 ADSP-21160 Cluster Bus

The Hammerhead-PMC+ ADSP-21160 cluster bus is a 40 MHz, 64-bit bus that connects the four ADSP-21160 processors in the cluster and a bank of up to 512 MB SDRAM. They are connected to the PCI interface through the SharcFIN ASIC. 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.

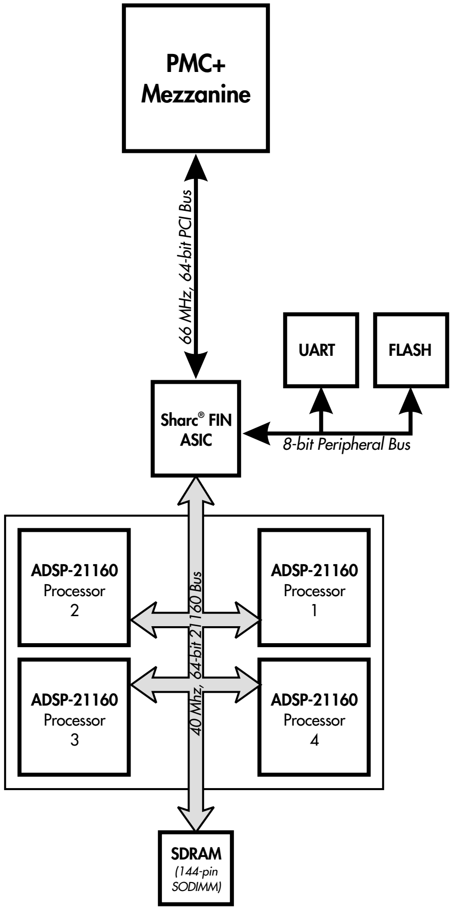
The ADSP-21160 cluster bus has access to the peripheral bus via

- BMS
- MS1

4.3 PCI Interface Architecture

The Hammerhead-PMC+ PCI interface consists of a 64-bit, 66 MHz PCI bus, which connects to the ADSP-21160 DSPs via the SharcFIN ASIC. Figure 4–5 is a block diagram of the PCI interface.

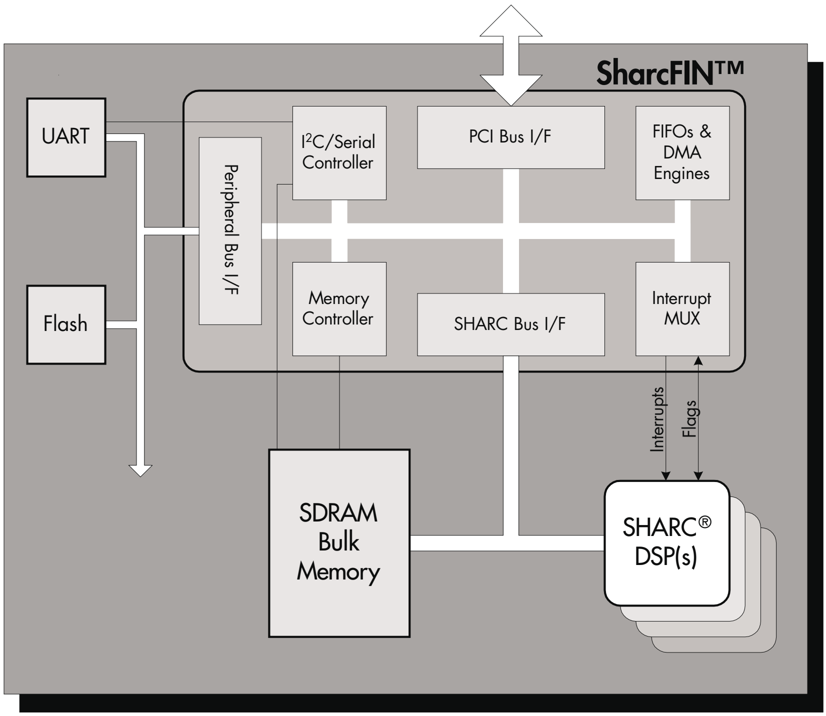
Figure 4–5 *Block Diagram of the PCI Interface Architecture*



4.3.1 Overview of the SharcFIN Architecture

The Hammerhead-PMC+ features BittWare's SharcFIN ASIC, which interfaces the ADSP-21160 DSPs to the PCI bus, SDRAM, and devices on the peripheral bus. It also provides an interrupt multiplexer to allow you to configure the interrupt connections on the board. This section provides an overview of the SharcFIN architecture. Figure 4–6 below is a simplified block diagram of the SharcFIN architecture as it is implemented on the Hammerhead-PMC+ board.

Figure 4–6 *Simplified Block Diagram of the SharcFIN Architecture*



Interface to ADSP-21160 Cluster Bus

The first function of the SharcFIN is to interface to the ADSP-21160 cluster bus. The SharcFIN provides a 64-bit interface to the ADSP-21160 cluster bus; it also integrates a full-featured SDRAM controller, which allows the ADSP-

21160s to access SDRAM using burst mode access at sustained data rates of 40 MB/sec.

Interface to PCI

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

Interface to Peripheral Bus

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

I^2C Serial Controller

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

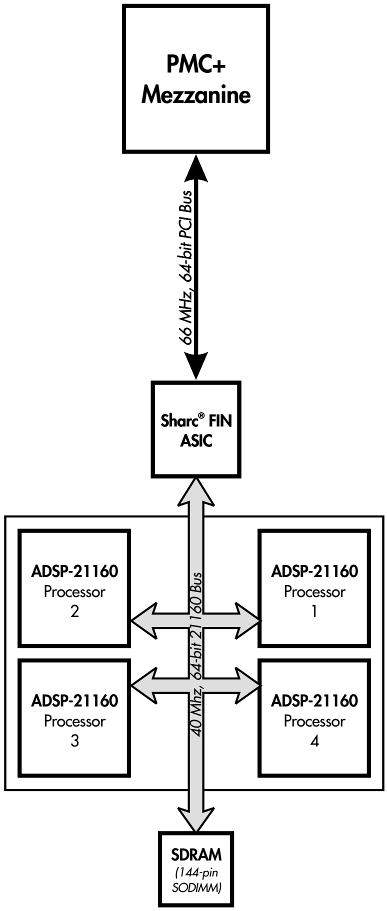
Interrupt Multiplexer

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

4.3.2 PCI Bus Interface

The Hammerhead-PMC+'s 64-bit 66 MHz PCI bus interfaces to the ADSP-21160 DSPs via the PCI-to-DSP bridge. The PCI-to-DSP interface allows the ADSP-21160 processors to communicate with the PCI bus, and vice-versa. It consists of the SharcFIN ASIC (PCI-to-DSP bridge) and the PCI bus. Figure 4–7 illustrates the PCI-to-DSP interface.

Figure 4–7 *Block Diagram of the PCI-to-DSP Interface*



4.3.3 Peripheral Bus

A 20 MHz, 8-bit peripheral bus extends off of the SharcFIN ASIC. The peripheral bus connects to low-speed peripherals, which include the Flash memory, the 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 interfaces to the SharcFIN (see section 4.3.1), which connects it to the ADSP-21160 DSPs and the PCI interface. The peripheral bus operates at 3.3 volts but can tolerate inputs of 5 volts.

4.4 PMC+ Interface Architecture

This section explains the connections available via the PMC interface, which include a 64-bit 66 MHz PCI interface, link ports, a TDM serial bus, a reset line, and an I^2C interface.

The PMC interface features four connectors. The first three connectors (P1–P3) provide the 64-bit, 66 MHz PCI interface. The fourth connector (P4) provides BittWare's PMC+ extensions.

4.4.1 PMC-to-PCI Interface

The first three connectors from each PMC interface provide the 64-bit, 66 MHz PCI interface. The PMC connects directly to the SharcFIN ASIC.

4.4.2 PMC+ Extensions

The fourth connector from the PMC interface provides BittWare's PMC+ extensions, which include link ports, a TDM serial bus, a reset line, and an I^2C interface. Please note that the PMC+ extensions are only available when the Hammerhead-PMC+ is attached to a BittWare PMC+ capable carrier board.

Table 4–7 *ADSP-21160 Link and Serial Port Connections to PMC Interfaces*

Board Configuration	PMC Link 1	PMC Link 2	PMC Link 3	PMC Link 4	Serial TDM 0
Quad	21160-1 L4	21160-2 L0	21160-3 L0	21160-4 L0	SPORT 0[a]
Dual	21160-1 L4	21160-2 L0	21160-2 L2	21160-1 L5	SPORT 0*
Single	21160-1 L4	21160-1 L3		21160-1 L5	SPORT 0*

a Includes all processors.

5

Chapter 5

Programming Details for the ADSP-21160s

This chapter supplies the information you will need when you are writing programs that allow the DSPs and the host to communicate. We do not provide the programs themselves; rather, we give programming details specific to the Hammerhead-PMC+. 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 SharcFIN ASIC.

5.1 Accessing the DSPs' Memory

5.1.1 Accessing Internal Memory

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

5.1.2 Accessing the DSPs' Multiprocessor Memory

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

Table 5–1 *Multiprocessor Memory Allocation for DSPs*

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

5.1.3 Accessing External Memory Banks

External memory space refers to the off-chip memory or memory-mapped peripherals that are attached to the ADSP-21160 cluster buses. On the Hammerhead-PMC+, these devices include the SDRAM, the Flash, the SharcFIN ASIC, and the UART.

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

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

Table 5–2 *External Memory Bank Allocation*

External Memory Bank	Memory Select Line	Description	Wait State	Wait Mode
0	MS0	SDRAM	1	2
1	MS1	Flash, Dual RS-232 UART	7	0
2	MS2	SharcFIN configuration registers	1	2
3	MS3	Unused	7	0

Setting the Size of the External Memory Banks

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

You can use the BittWare Configuration Manager (see "Verifying that the Board is Configured Properly" on page 21) to set the MSIZE bits. The default setting for the MSIZE should be equal to the size of the largest external memory device, which is the SDRAM. Table 5–3 lists the recommended settings for MSIZE and shows how MSIZE affects the bank addresses. Note that programming the MSIZE bits may affect where other resources available to the ADSP-21160 processor are located.

Table 5–3 Recommended MSIZE Settings for the Hammerhead-PMC+

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

Accessing the SDRAM

The Hammerhead-PMC+ supports a bank of up to 512 MB of SDRAM which is located on the 64-bit ADSP-21160 cluster bus. The DSPs can access the SDRAM via MS0.

Accessing the Flash, UART, and Peripheral Bus

MS1 provides access to the peripheral bus and all devices located on it, including the Flash memory and the dual UART.

Accessing the SharcFIN ASIC

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

5.1.4 Accessing Unbanked Memory Space

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

5.2 Booting the DSPs

This section explains the three booting options for the Hammerhead-PMC+: link port, Flash, and PCI.

5.2.1 Booting the Board via Link Port

In link port booting, the DSP gets boot data from another DSP's link port or from a 4- or 8-bit wide external device [1] after system power-up. 21160-1 is connected to external link port J7. After booting via link port, 21160-1 will boot the remaining DSPs on the board.

To boot the Hammerhead-PMC+ via link port,

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

For additional information on link booting, refer to the *ADSP-21160 User's Manual* (Analog Devices).

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

5.2.2 Booting the Board From the Host

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

To boot the Hammerhead-PMC+ from the host using HIL functions or Diag21k commands,

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

For additional information on these functions, refer to the DSP21k-SF Toolkit documentation.

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

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

For more information on booting the DSPs from the host without the HIL, refer to the *ADSP-21160 User's Manual* (Analog Devices).

5.2.3 Booting the Board via the Flash

The Flash memory allows you to boot the Hammerhead-PMC+ in standalone mode, without a host PC.

Developing the Boot Program

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

Loading a Boot Program into the Flash Memory

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

6

Chapter 6

SharcFIN Programming Details

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

6.1 Overview of the SharcFIN

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

6.1.1 The Two Sections of the SharcFIN

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

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

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

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

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

6.2 Function of the SharcFIN PCI Interface

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

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

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

6.2.1 Performing PCI Side DMAs

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

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

6.2.2 Performing a PCI Side Single Access

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

6.2.3 Performing PCI Side Interrupts

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

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

Note

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

6.3 Function of the SharcFIN SHARC Interface

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

6.3.1 ADSP-21160 Bus Interface

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

6.3.2 SDRAM Interface and Control

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

Accessing SDRAM from the PCI Side

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

The SDRAM window has the following two limitations:

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

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

SDRAM Timing from the ADSP-21160

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

Using DMA-Based SDRAM Accesses

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

6.3.3 Peripheral Bus Interface (Flash and Dual UART)

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

6.3.4 I^{2}C Interface

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

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

6.3.5 Interrupt Multiplexer

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

How the Interrupt Multiplexer Functions

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

Note

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

Creating PCI Side Interrupts

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

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

6.4 Overview of the SharcFIN Memory Map

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

Note

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

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

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

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

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

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

* Word accesses will produce erroneous data.

† Byte writes will corrupt the rest of the word.

6.4.1 Accessing System Settings and Configuration Registers

BAR0 = MS2 = system settings and configuration registers

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

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

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

6.4.2 Accessing the Flash, UART, and Peripheral Bus

BAR1 = MS1 = Flash/UART/Peripheral bus

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

Warning

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

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

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

6.4.3 Accessing Multiprocessor Memory Space

BAR 2 = MMS = flat map of Multiprocessor Memory Space

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

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

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

6.4.4 Accessing SDRAM

BAR 4 = MS0 = SDRAM[1]

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

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

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

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

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

1. Some caveats apply.

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

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

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

6.5 Setting the SharcFIN's SHARC Interface Control Registers

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

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

Note

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

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

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

* Write Once Read Many

6.5.1 Address Override Register

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

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

Table 6–9 *Address Override Register Description*

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

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

6.5.2 Status Register

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

Table 6–10 *Contents of the Status Register*

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

6.5.3 Peripheral Bus Configuration Register

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

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

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

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

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

6.5.4 Watchdog Configuration Register

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

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

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

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

6.5.5 PMC+ Configuration Register

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

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

Bit	Name	Type	Reset Value	Function
0	PMC Flg/Int En	R/W	0	Configures the PMC+ interface's bus mode lines to be used as flag interrupts.* 1 = The PMC+ interface's bus mode lines will be used as flag interrupts 0 = The PMC+ interface's bus mode lines will be used as bus mode lines
3:1	BusMode2, BusMode3, BusMode4	R/W	BusMode2: 1 BusMode1: 0 BusMode3: 0	Tells the PMC site whether or not it should drive BusMode1 to indicate its presence. When the flag interrupts are disabled (by *PMC Flg/Int En* bit), the BusMode lines work according to the PMC specification. Bits 1–3 are Bus Mode lines 2–4. B1 = 1 Bus Mode line 2 B2 = 0 Bus Mode line 3 B3 = 0 Bus Mode line 4
4	BusMode 1	R/O		Bus Mode line 1 is an input†. It indicates that a PMC card is present on the board.‡ 0 = PMC board is present
5	BusMode2	R/O		Current status of BusMode2 line
6	BusMode3	R/O		Current status of BusMode3 line
7	BusMode4	R/O		Current status of BusMode4 line

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

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

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

6.5.6 Onboard I²C Control Register

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

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

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

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

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

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

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

6.5.7 PMC+ I^2C Control Register

The PMC+ I^2C Control register (offset 0x47) 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.

6.5.8 SDRAM Configuration Registers

The SharcFIN contains two registers that configure the SDRAM:

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

SDRAM Size Configuration Register

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

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

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

SDRAM Window Register

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

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

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

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

6.5.9 DMA Address Register

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

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

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

6.5.10 DMA Configuration Register

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

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

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

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

* X = Unknown

6.5.11 Interrupt Configuration Registers

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

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

Table 6–20 *SharcFIN Interrupt Configuration Registers*

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

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

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

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

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

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

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

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

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

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

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

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

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

Flag and Interrupt Status Registers

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

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

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

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

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

Appendix A
Debugging Your DSP Programs

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

A.1 Debugging with a Hardware Emulator

This section discusses attaching an in-circuit emulator (ICE) from Analog Devices to the Hammerhead-PMC+ board. To use an emulator with the Hammerhead-PMC+, follow the steps below:

1. Connect the probe on the ICE card to the Hammerhead-PMC+.
2. Depending on the form factor of the ICE card you are using, either install it in or connect it to your PC.
3. Connect the Hammerhead-PMC+ to the PC.
4. Apply power to the Hammerhead-PMC+.
5. Start the emulator software on the PC.

A.1.1 Overview of the ICE Emulator

The Hammerhead-PMC+ is compatible with Analog Devices's in-circuit emulator, which is a separate ISA bus, PCI bus, Ethernet, or USB card that connects to the Hammerhead-PMC+'s JTAG connector and installs in or connected to a PC. The ICE provides a controlled environment for observing, debugging, and testing real-time activities in a target hardware environment by connecting directly to the target processor through its JTAG interface.

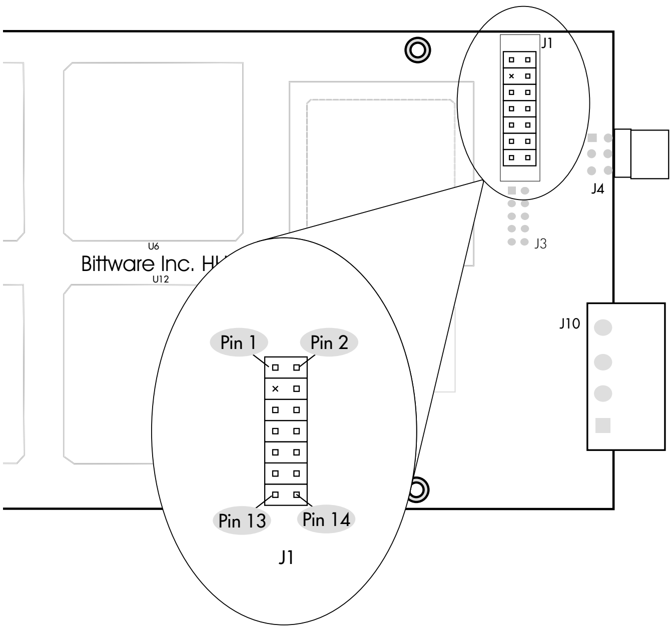
When the ICE is attached to the Hammerhead-PMC+, the Hammerhead-PMC+ becomes the target system for the emulator, allowing you to operate it completely from the emulator's user interface. A powerful tool for debugging programs running on the ADSP-21065L processors, the emulator monitors system behavior 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 Emulator to the Hammerhead-PMC+

To attach the ICE emulator to the Hammerhead-PMC+, follow the instructions below.

1. Locate the JTAG connector on the Hammerhead-PMC+ (see Figure A–1).
2. A cable extends from the ICE card to a probe that connects to the JTAG connector on the Hammerhead-PMC+. 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 Locations*



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 3–3 on page 40 shows the connector pinout.

A.1.3 Installing the Emulator in a PC

Once you have connected the ICE to the Hammerhead-PMC+, install the ICE in or connect it to your PC (depending on the form factor you have chosen). The *ICE Hardware User's Guide* from Analog Devices explains how to install the ICE emulator.

A.1.4 Connecting the Hammerhead-PMC+ to the Host

After installing the ICE you will need to mount the Hammerhead-PMC+ onto a PMC carrier board (see section 2.3.1).

A.1.5 Operating the ICE Emulator

To start operating the ICE emulator with the Hammerhead-PMC+,

1. Apply power to the Hammerhead-PMC+.

> **Note**
>
> *As long as the emulator software is not running, you can safely attach and remove the ICE probe while the Hammerhead-PMC+ is running.*

2. Start the emulator software on your PC. To download and run programs, follow the instructions in the ICE documentation. The emulator software allows you to select which processor to debug.

A.2 Debugging with a Software Emulator

BittWare's VisualDSP Target is a software plug-in for Analog Devices' VisualDSP++ that works with the VisualDSP++ debugger to allow you to debug your DSP programs directly on your BittWare board without using a hardware 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 the Board for Standalone Operation

The Hammerhead-PMC+ can boot via link ports or from a boot program stored in its Flash memory (see section 2.2.1), which allows it to operate in standalone mode, free from a host computer. This section lists the steps necessary to prepare your Hammerhead-PMC+ to operate in standalone mode.

1. While in development mode, develop a boot loader and a standalone operating program for the DSPs (see B.1.1). In lieu of a host, the DSP program code should start with configuring the PCI interface.
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 14).
4. Set the Standalone Mode jumpers (see “Setting the Standalone Operation Jumper” on page 14).
5. Mount standoffs on the board (see B.5).
6. Apply power to the Hammerhead-PMC+ (see B.6).
7. Boot the board in Flash or Link boot mode.

Note

If you are not planning to operate the Hammerhead-PMC+ in standalone mode, follow the set up instructions in Chapter 2.

B.1 Developing and Loading a Boot Program

B.1.1 Developing the Boot Program

Use the Analog Devices software development tools and BittWare's host interface tools (DSP21k-SF Toolkit) to develop a boot program for the ADSP-21160 processors. The example software included with the Hammerhead-PMC+ 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-PMC+ to load the Flash memory with a boot program. 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-PMC+ board.

B.1.3 Loading a Link Boot Program

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

B.2 Setting the Boot Mode Jumpers

The Hammerhead-PMC+ has two configuration jumpers (JP2 and JP4) that configure its boot mode. See "Setting the Boot Mode for the Processors" on page 14 for instructions on setting the jumpers.

B.3 Setting the Standalone Mode Jumper

The Hammerhead-PMC+ has one configuration jumper, JP12, that must be set in order for the board to operate in standalone mode. This jumpers must be ON for the board to operate properly in standalone mode. See "Setting the Standalone Operation Jumper" on page 14 for more detail.

B.4 Booting the Board via Flash or Link

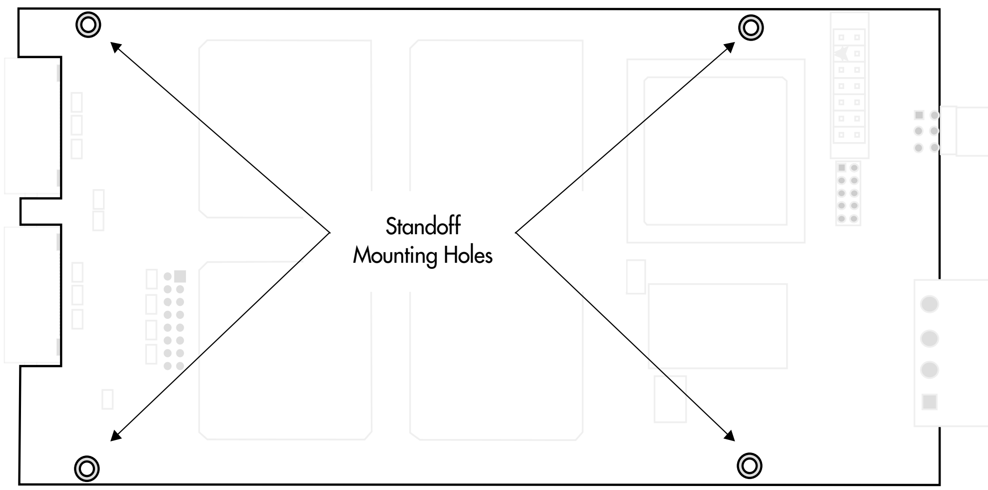
As described in sections 5.2.1 and 5.2.3, you may boot the board via either the Flash or link port. Follow the instructions outlined in these sections to boot the board in standalone mode.

B.5 Mounting Standoffs on the Hammerhead-PMC+

If desired, mount standoffs on the Hammerhead-PMC+ to provide adequate clearance between the work surface and the Hammerhead-PMC+'s components.

1. Place ¼" standoffs in the standoff mounting holes on the Hammerhead-PMC+. 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*



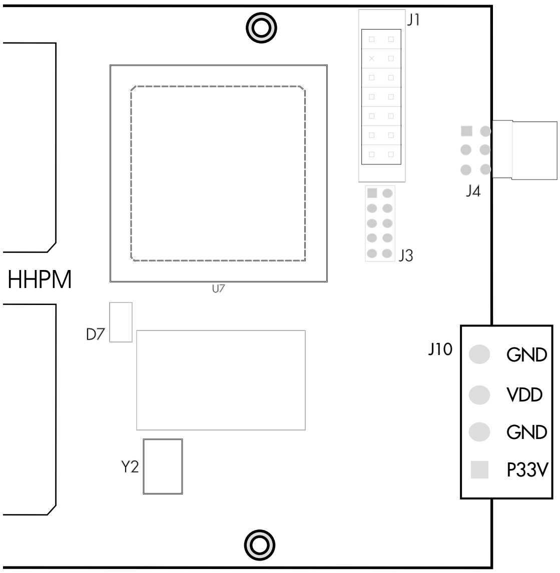
B.6 Supplying Power to the Board

The Hammerhead-PMC+ requires a +3.3 power supply for normal operation. The external power connector (J10) supplies +3.3V and +5V to the Hammerhead-PMC+. Table 3–9 on page 50 gives the pinout of the connector.

To connect an external power source to the Hammerhead-PMC+,

1. Plug a power adapter cable into the Hammerhead-PMC+'s external power connector (J10). Be sure to align pin 1 (GND) on the cable with the GND pin on J10.
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-PMC+. Section 2.4 explains in more detail how to reset the board.

Figure B–2 *Location of the External Power Connector (Top)*



Appendix C
Troubleshooting Tips

This section lists the information you will need to have ready before calling technical support at BittWare. It also provides the telephone number and E-mail address for BittWare 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 these tests 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 95/98, Windows 95B (OSR2), Windows NT Version 3.51, Windows NT Version 4.0, Windows 2000, or Linux

    - The release number of your DSP21k Toolkit (enter `diag21k -v` at a DOS prompt)
- If you could be at the PC that is experiencing problems when making the call, we would better be able to start investigating the problem.

C.2 Contacting Technical Support

You can reach technical support at BittWare, Inc. using one of the following methods:

- Phone (8:30am – 5:30pm 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.