Hammerhead-PC/104-*Plus*

Dual ADSP-21160 32-bit/33 MHz PC/104-*Plus* 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-PC/104-Plus User's Manual

Hardware Revision 1

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

Printed in the USA
October 24, 2001 Edition UG-HHP401-00

Contents

Chapter 1 Introduction

1.1 Overview of the Hammerhead-PC/104-Plus System 2
1.1.1 Hammerhead-PC/104-Plus Features 2
1.1.2 Hammerhead-PC/104-Plus System Architecture 2
1.1.3 Hammerhead-PC/104-Plus Software Architecture 4

1.2 About this User's Guide 6
1.2.1 Purpose of this User's Guide 6
1.2.2 Conventions Used Throughout this User's Guide 6
1.2.3 Chapter Overviews 6

1.3 Other Helpful Documents and Tools 8
1.3.1 Documents for Further Reference 8
1.3.2 Software Development Tools 8

Chapter 2 Getting Your Hammerhead-PC/104-Plus Ready for Operation

2.1 Unpacking the Hammerhead-PC/104-*Plus* *12*

2.2 Configuring the Hammerhead-PC/104-Plus 13
2.2.1 Setting the Hammerhead-PC/104-*Plus* Configuration Jumpers .. 13
2.2.2 Configuring the External Serial Ports 16
2.2.3 Configuring the Board According to Position in the PC/104-Plus Stack 18

2.3 Connecting External Signals to the Board 20
2.3.1 Connecting Serial Port Cables 20
2.3.2 Connecting RS-232 Cables 20
2.3.3 Connecting an External Power Supply 21
2.3.4 Connecting an External Reset Signal to the Reset Connector 21

2.4 Installing the Board and its Software 23
2.4.1 Installing the Analog Devices Software Tools 23
2.4.2 Installing the BittWare DSP21k-SF Toolkit 23
2.4.3 Mounting the Board on a PC/104-Plus Stack 24
2.4.4 Verifying that the Board is Configured Properly 25

2.5 Testing the Board to Make Sure it is Operating Properly 26
2.5.1 Testing the Board with the DSP21k-SF Toolkit Diagnostic Utilities 26
2.5.2 Testing the Board with the Hammerhead-PC/104-*Plus* Example Files 31

2.6 Resetting the Hammerhead-PC/104-*Plus* *32*
2.6.1 Resetting the Board with the Watchdog Timer 32
2.6.2 Resetting the Board via the External Reset Connector 33
2.6.3 Resetting the Board via the PCI Interface 33

Chapter 3 Overview of the Hardware Components

3.1 Function and Location of the Major Components 36
3.1.1 ADSP-21160 SHARC Processors 37
3.1.2 External Memory Banks 38
3.1.3 SharcFIN ASIC 38
3.1.4 On-Board Oscillators 39
3.1.5 Dual RS-232 UART 39
3.1.6 User LEDs 39
3.1.7 Watchdog Timer 39

3.2 Layout and Function of the External Connectors 40
3.2.1 PC/104-*Plus* Connectors 43
3.2.2 External Power Connector 46
3.2.3 External Link Ports 47
3.2.4 External Serial Ports 49
3.2.5 Expansion Connector 51
3.2.6 JTAG Connector 53
3.2.7 SDRAM (SODIMM) 54
3.2.8 Flag I/O Connector 54
3.2.9 RS-232 Connector 55
3.2.10 External Reset Connector 56

3.3 Function and Location of the Configuration Jumpers and Switches 57

Chapter 4 Hammerhead-PC/104-*Plus* Board Architecture

4.1 Overview of the Board Architecture 60

4.2 ADSP-21160 Architecture 62
4.2.1 Resources Available to the ADSP-21160s 62
4.2.2 ADSP-21160 Memory Structure 63

4.2.3 Serial Port Connections 64
4.2.4 Link Port Connections 65
4.2.5 Flag and Interrupt Connections 66
4.2.6 ADSP-21160 Cluster Bus 69

4.3 PCI Interface Architecture 70
4.3.1 SharcFIN Architecture 71
4.3.2 PCI Bus Interface 72
4.3.3 Peripheral Bus Interface 73

Chapter 5 Programming Details for the ADSP-21160s

5.1 Accessing the DSP's Memory 76
5.1.1 Accessing Internal Memory 76
5.1.2 Accessing the DSP's Multiprocessor Memory 76
5.1.3 Accessing External Memory Banks 76
5.1.4 Accessing Unbanked Memory Space 79

5.2 Booting the DSP 80
5.2.1 Booting the Board via Link Port 80
5.2.2 Booting the Board From the Host 80
5.2.3 Booting the Board via the Flash 81

Chapter 6 SharcFIN Programming Details

6.1 Overview of the SharcFIN 84
6.1.1 The Two Sections of the SharcFIN 84
6.1.2 How the SharcFIN Maps to the PCI and ADSP-21160 Buses 84

6.2 Function of the SharcFIN PCI Interface 85
6.2.1 Performing PCI Side DMAs 85
6.2.2 Performing a PCI Side Single Access 86
6.2.3 Performing PCI Side Interrupts 86

6.3 Function of the SharcFIN SHARC Interface 87
6.3.1 ADSP-21160 Bus Interface 87
6.3.2 SDRAM Interface and Control 87
6.3.3 Peripheral Bus Interface (Flash and Dual UART) 88
6.3.4 I^2C Interface 88
6.3.5 Interrupt Multiplexer 89

6.4 Overview of the SharcFIN Memory Map 91
6.4.1 Accessing System Settings and Configuration Registers 92

6.4.2 Accessing the Flash, UART, and Peripheral Bus 93
6.4.3 Accessing Multiprocessor Memory Space 94
6.4.4 Accessing SDRAM 95

6.5 Setting the SharcFIN's SHARC Interface Control Registers..... 97
6.5.1 Address Override Register 99
6.5.2 Status Register 100
6.5.3 Peripheral Bus Configuration Register 101
6.5.4 Watchdog Configuration Register 102
6.5.5 PMC+ Configuration Register 103
6.5.6 Onboard I^2C Control Register 104
6.5.7 PMC+ I^2C Control Register 105
6.5.8 SDRAM Configuration Registers 105
6.5.9 DMA Address Register 107
6.5.10 DMA Configuration Register 107
6.5.11 Interrupt Configuration Registers 110

Appendix A Debugging Your DSP Programs

A.1 Debugging with a Hardware (In-Circuit) Emulator..... 119
A.1.1 Overview of the ICE Emulator 119
A.1.2 Attaching the ICE to the Hammerhead-PC/104-Plus 120
A.1.3 Installing the ICE and Hammerhead-PC/104-Plus In a PC 121
A.1.4 Operating the ICE 121

A.2 Debugging with a Software Emulator..... 122
A.2.1 About the VisualDSP Target 122
A.2.2 Installing the VisualDSP Target 122

Appendix B Setting Up for Standalone Operation

B.1 Developing and Loading a Boot Program..... 124
B.1.1 Developing the Boot Program 124
B.1.2 Loading a Boot Program into the Flash Memory 124
B.1.3 Loading a Link Boot Program 124

B.2 Configuring the Board for Standalone Mode..... 124
B.2.1 Setting the Standalone Mode Jumper 124
B.2.2 Selecting the Boot Mode 125

B.3 Mounting Standoffs on the Board..... 125

B.4 Connecting an External Power Supply 126

B.5 Booting the Board 126

Appendix C Troubleshooting Tips

C.1 Before You Call Technical Support 127

C.2 Contacting Technical Support 128

Figures

Figure 1–1 Block Diagram of the Hammerhead-PC/104-Plus System 3
Figure 1–2 Software Model for the Hammerhead-PC/104-Plus 5
Figure 2–1 Layout of the Hammerhead-PC/104-*Plus* Configuration Jumpers & Switch (Top) 13
Figure 2–2 Layout of the Hammerhead-PC/104-*Plus* Configuration Jumper & Switches (Bottom) 14
Figure 2–3 Standard Serial Port Dip Switch Positions (S2 and S3) 17
Figure 2–4 TDM Serial Port Dip Switch Positions (S1) 18
Figure 2–5 Rotary Switch to Select Position in PC/104 Stack (Top) 19
Figure 2–6 Cable Details for the External Reset Connector J16 (Top) 22
Figure 2–7 Cable Details for the Hammerhead-PC/104-Plus External Reset Connector 22
Figure 2–8 PC/104-*Plus* Stack with Hammerhead-PC/104-*Plus* *25*
Figure 3–1 Layout of the Hammerhead-PC/104-*Plus*'s Major Components (Top) 36
Figure 3–2 Layout of the Hammerhead-PC/104-*Plus*'s Major Components (Bottom) 37
Figure 3–3 Layout of the Hammerhead-PC/104-Plus's External Connectors (Top) 40
Figure 3–4 Layout of the Hammerhead-PC/104-Plus's External Connectors (Bottom) 41
Figure 3–5 Location of the PC/104-*Plus* Connector Pins 43
Figure 3–6 Location of the External Power Connector Pins (Bottom) 46
Figure 3–7 Location of External Link Connectors J2, J7, and J11 (Top) 48
Figure 3–8 Location of External Link Connector J3 (Bottom) 48
Figure 3–9 Location of the External Serial Port Connector Pins (Top) 50
Figure 3–10 Location of the Expansion Bus Connector Pins (Bottom) 51
Figure 3–11 Location of the JTAG Connector Pins (Top) 53
Figure 3–12 Location of the Flag I/O Connector Pins (Top) 54
Figure 3–13 Location of the RS-232 Connector Pins (Top) 55
Figure 3–14 External Reset Connector Pins (Top) 56
Figure 3–15 Location of the Configuration Jumpers and Switches (Top) 58
Figure 3–16 Location of the Configuration Jumpers and Switches (Bottom) 58
Figure 4–1 Block Diagram of the Hammerhead-PC/104*Plus* System 60
Figure 4–2 Block Diagram of Serial Port Connections 64
Figure 4–3 Block Diagram of Link Port Connections 65
Figure 4–4 Block Diagram of Flag and Interrupt Connections 66
Figure 4–5 Block Diagram of the PCI Interface 70
Figure 4–6 Simplified Block Diagram of the SharcFIN Architecture 71
Figure A–1 JTAG Connector 120
Figure B–1 Location of the Standoff Mounting Holes 125

Tables

Table 2–1 Settings for the Boot Mode Selection Jumpers 15
Table 2–2 Selecting the Rotating Priority Bus Mode for the ADSP-21160 (JP7) 16
Table 2–3 Serial Port Dip Switch Connections 16
Table 2–4 Standard Serial Port Dip Switch Settings (S2, S3) 17
Table 2–5 TDM Serial Port Dip Switch Settings (S1) 18
Table 2–6 Rotary Switch Settings 19
Table 2–7 Serial Port Cables Available from BittWare 20
Table 3–1 Overview of the External Connectors 42
Table 3–2 PC/104 Connector Pinout 44
Table 3–3 PC/104-*Plus* Connector Pinout - PCI Interface 45
Table 3–4 External Power Connector Pinout (J1) 46
Table 3–5 Link Port Connector Pinout (J2, J3, J7, and J11) 47
Table 3–6 External Serial Port Connections and Usage 49
Table 3–7 External Serial Port Connector Pinout (J4, J6, J9) 50
Table 3–8 Peripheral Bus Connector Pinout 52
Table 3–9 JTAG Connector Pinout (J12) 53
Table 3–10 Flag I/O Connector Pinout 54
Table 3–11 RS-232 Connector Pinout (J15) 55
Table 3–12 External Reset Connector Pinout (J16) 56
Table 3–13 Hammerhead-PC/104-*Plus* Configuration Jumpers 57
Table 3–14 Hammerhead-PC/104-Plus Serial Port Dip Switches 57
Table 4–1 Resources for 21160-1 62
Table 4–2 Resources for 21160-2 63
Table 4–3 ADSP-21160 Flag Connections 67
Table 4–4 ADSP-21160 Interrupt Connections 67
Table 5–1 Multiprocessor Memory Allocation for DSP 76
Table 5–2 External Memory Bank Allocation 77
Table 5–3 Recommended MSIZE Settings for the Hammerhead-PC/104-Plus 78
Table 6–1 Overview of How the SharcFIN Maps to the PCI and ADSP-21160 Buses 91
Table 6–2 Accessing BAR0–BAR4 From the PCI Side 92
Table 6–3 PCI and ADSP-21160 Addresses for System Settings and Configuration Registers 92
Table 6–4 PCI and ADSP-21160 Addresses for Flash, UART, and Peripheral Bus 93
Table 6–5 PCI and ADSP-21160 Addresses for Multiprocessor Memory Space 94
Table 6–6 PCI and ADSP-21160 Addresses for 64 MB SDRAM 95
Table 6–7 PCI and ADSP-21160 Addresses for 128 MB SDRAM 96
Table 6–8 Memory Map for the SHARC Interface Control Registers 98
Table 6–9 Address Override Register Description 99
Table 6–10 Contents of the Status Register 100
Table 6–11 Contents of the Peripheral Bus Configuration Register 101
Table 6–12 Contents of the Watchdog Configuration Register 102

Table 6–13 Contents of the PMC+ Configuration Register 103
Table 6–14 Contents of the I^2C Control Register 104
Table 6–15 Effects of Values Written to the Clock and Data Bits (B0, B1) 104
Table 6–16 Contents of the SDRAM Size Configuration Register 105
Table 6–17 Contents of the SDRAM Window Register 106
Table 6–18 Contents of the DMA Address Register 107
Table 6–19 Contents of the DMA Configuration Register (Continues on next page) 108
Table 6–20 SharcFIN Interrupt Configuration Registers 110
Table 6–21 Contents of the ADSP-21160 Interrupt Configuration Registers (0x50, 0x52, 0x54, 0x56) (Continues on next page) 111
Table 6–22 Contents of the PCI Interrupt Configuration Register (0x58) (Continues on next page) 113
Table 6–23 Contents of the PMC+ Interrupt Configuration Register (0x5A) (Continues on next page) 115
Table 6–24 Contents of the Flag Status Register 117
Table 6–25 Contents of the Interrupt Status Register 118

Chapter 1
Introduction

The Hammerhead-PC/104-*Plus* is a PC/104-*Plus* format board featuring two ADSP-21160 SHARC® DSPs from Analog Devices, each of which provide 600 MFLOPS peak processing power. Since it is a PC/104-*Plus* format board, it features a 32-bit, 33 MHz PCI interface. The Hammerhead-PC/104-*Plus* supports up to 512 MB of SDRAM and 2MB of Flash memory, and the PCI interface features BittWare's SharcFIN™ ASIC. The SharcFIN flexibly interfaces the ADSP-21160s to the 32-bit, 33 MHz PCI interface, the SDRAM, the Flash memory, and a general-purpose expansion bus. The Hammerhead-PC/104-*Plus* can function as a standalone processor with the on-board boot Flash, or it can boot from the host computer via the PCI interface, or over the external link port.

This chapter covers the following topics:

- overviews the architecture of the Hammerhead-PC/104-*Plus* system and its software
- overviews each chapter in this user's guide
- lists documents that provide more information about the Hammerhead-PC/104-*Plus*'s components and software

1.1 Overview of the Hammerhead-PC/104-*Plus* System

This section gives a brief overview of the Hammerhead-PC/104-*Plus* board and describes the software necessary to communicate with the board.

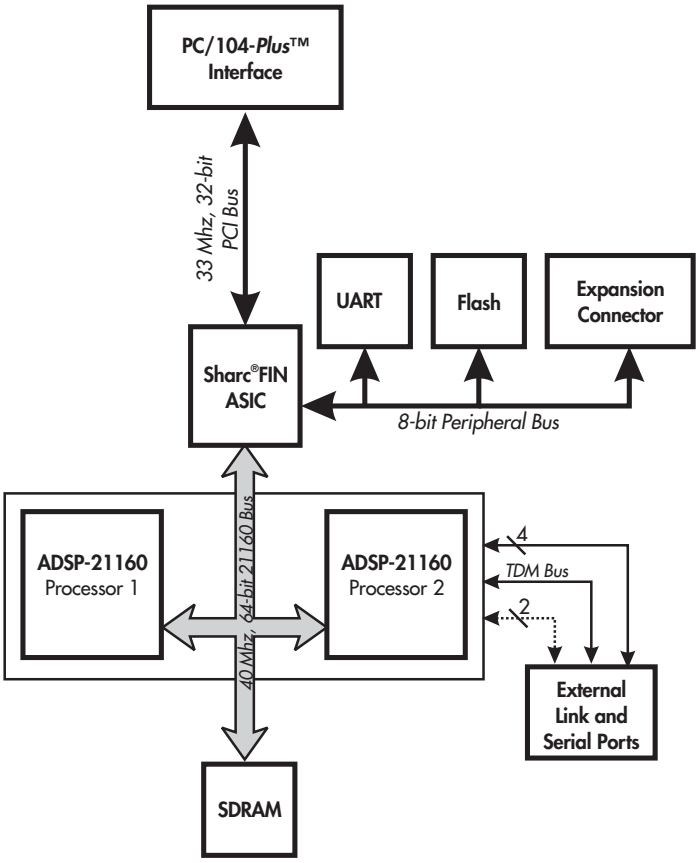
1.1.1 Hammerhead-PC/104-*Plus* Features

- one or two ADSP-21160 SHARC® processors (600 MFLOPS each)
- up to 512 MB SDRAM
- 2 MB of Flash memory with optional boot loading
- four external link ports at 80 MB/s each
- two 40 Mbit/s external serial ports
- one 40 Mb/s external serial TDM bus
- BittWare's SharcFIN ASIC
- 8-bit, 20 MHz peripheral bus
- RS-232 UART
- PCI interface or standalone operation

1.1.2 Hammerhead-PC/104-*Plus* System Architecture

This section gives a basic overview of the Hammerhead-PC/104-*Plus* system and describes how all of its features work together. Figure 1–1 is a detailed block diagram of the Hammerhead-PC/104-*Plus* board and its features.

Figure 1–1 *Block Diagram of the Hammerhead-PC/104-Plus System*



BittWare's Hammerhead-PC/104-*Plus* board features two Analog Devices ADSP-21160 processors, 64-512 MB of SDRAM, 2 MB of Flash memory, four external link ports, two external serial ports, and a serial TDM bus. The board's SharcFIN ASIC interfaces between the processor bus, the PCI bus and the 8-bit peripheral bus.

ADSP-21160 SHARC DSPs

The Hammerhead-PC/104-*Plus* board is configured with two 80 MHz ADSP-21160 processors. The two ADSP-21160 processors share a common 40 MHz, 64-bit cluster bus, which gives them access to the board's SDRAM, the PCI bus interface, and the other SHARC DSP. For additional I/O, each processor also has four flags, three interrupts, six link ports, and two serial ports.

SharcFIN™ ASIC for SHARC DSPs

The Hammerhead-PC/104-*Plus* incorporates BittWare's SharcFIN ASIC for SHARC DSPs. The SharcFIN flexibly interfaces the ADSP-21160 DSPs to the 32-bit, 33 MHz PCI bus, the SDRAM, the Flash memory, and a peripheral bus. It also provides a feature-rich set of DMA functions and interrupt options to support very high-speed, real-time data flow with minimum processor overhead.

I/O Support

The Hammerhead-PC/104-*Plus* offers a variety of user I/O options in addition to its 32-bit, 33 MHz PCI interface: external serial port connections, external link port connections, and an RS-232 interface.

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

The Hammerhead-PC/104-*Plus* features four 80 MB/s external link ports. Two link ports on each DSP connect to the external connectors and four links per DSP are dedicated for interprocessor communication.

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

1.1.3 Hammerhead-PC/104-*Plus* Software Architecture

You will need three types of software development tools for the Hammerhead-PC/104-*Plus*: 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-PC/104-*Plus*.

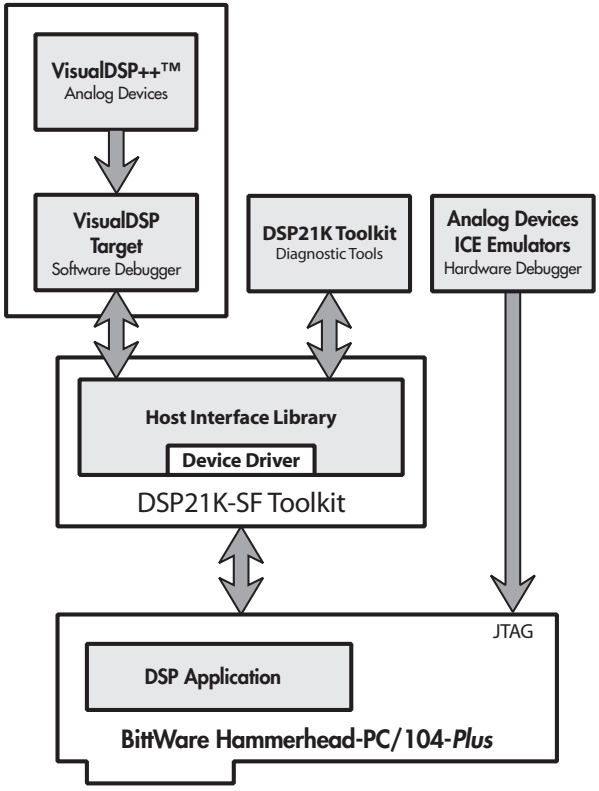
To begin developing code for the Hammerhead-PC/104-*Plus*, use Analog Devices' VisualDSP++® Integrated Development Environment (IDE). VisualDSP++ is an easy-to-use project management environment comprised of an integrated development environment (IDE) and software debugger.

BittWare's DSP21k-SF Toolkit provides your host interface tools. The DSP21k-SF Toolkit allows you to easily develop application code and integrate the Hammerhead-PC/104-*Plus* into your system. The software

tools include a comprehensive host interface library (HIL), a standard I/O library, and diagnostic utilities.

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.

Figure 1–2 *Software Model for the Hammerhead-PC/104-Plus*



1.2 About this User's Guide

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

1.2.1 Purpose of this User's Guide

This user's guide covers hardware revision 0 of the Hammerhead-PC/104-*Plus* board, which supports two ADSP-21160 SHARC processors operating at 80 MHz. The purpose of this document is to provide details about the Hammerhead-PC/104-*Plus*'s major hardware components, to describe how to install and properly operate the Hammerhead-PC/104-*Plus*, 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.

- Since the Hammerhead-PC/104-*Plus* has two processors, we refer to them as 21160-1 and 21160-2.
- All signal names appear in small capitals (RESET).
- Active low signals appear in small capitals with an overline ($\overline{\text{RESET}}$).
- A "0x" prefix designates a number as a hexadecimal number (0x01).
- Commands that the user enters (for programs such as Diag21k or DspBad in the DSP21k-SF Toolkit) appear in the **`Courier bold`** font.
- Filenames and directories appear in the `Courier` font.

1.2.3 Chapter Overviews

Chapter 2: Getting Your Hammerhead-PC/104-Plus Ready for Operation

This chapter describes the tasks that you must perform to prepare your board for installation, install the software for the board, install the board, and test the installation.

Chapter 3: Overview of the Hardware Components

This chapter shows the location of the Hammerhead-PC/104-*Plus*'s major components and connectors and briefly discusses their function.

Chapter 4: Hammerhead-PC/104-Plus Board Architecture

This chapter discusses the board's architecture and its serial ports, link ports, flags and interrupts, and bus interfaces.

Chapter 5: Programming Details for the ADSP-21160s

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

Chapter 6: SharcFIN Programming Details

This chapter provides a brief functional overview of the SharcFIN ASIC and describes how to configure its SHARC interface control registers.

Appendix A: Debugging Your DSP Programs

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

Appendix B: Setting Up for Standalone Operation

This appendix describes how to set up the board and run it in standalone mode. It explains the steps involved in writing a boot program, loading the boot program into Flash memory, supplying power to the board, and booting it in Flash boot mode.

Appendix C: Troubleshooting Tips

This appendix discusses common operating problems and how to contact technical support at BittWare.

1.3 Other Helpful Documents and Tools

This section describes where to look for more information that applies to the Hammerhead-PC/104-*Plus* or its components. It also lists several third party software development tools that you may find useful.

1.3.1 Documents for Further Reference

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

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

DSP21k-SF Toolkit Installation and Reference Guides – BittWare, Inc.

SharcFIN ASIC User's Guide – Bittware, Inc.

1.3.2 Software DevelopmentTools

VisualDSP++ and BittWare VisualDSP Target

The Hammerhead-PC/104-*Plus* is compatible with the VisualDSP++ software 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' SHARC C compiler, C runtime library, assembler, linker, loader, simulator, and splitter. The debugger works together with a software target, an emulator, or a simulator and 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

Analog Devices' ICE in-circuit emulators provide real-time hardware emulation and debugging. With the 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 ICE emulator with the Hammerhead-PC/104-*Plus*, 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-PC/104-*Plus* with the DSP21k-SF Toolkit. Using the Toolkit's C-callable library of routines for DOS, Windows, and Linux programs, you can download and start programs, read from and write to the Hammerhead-PC/104-*Plus's* memory, and control other board functions. Another library gives your DSP programs standard I/O routines such as screen display, keyboard input, and disk file access. The *DSP21k-SF Toolkit User's Guide* from BittWare, Inc. contains complete information about the DSP21k-SFToolkit.

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

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

Chapter 2
Getting Your Hammerhead-PC/104-*Plus* Ready for Operation

This chapter describes how to prepare your board for installation, install the software for the board, install the board, and run diagnostic tests on the board to make sure it is working properly. This chapter does not provide comprehensive instructions for all of the tasks; instead, it provides a sequence of steps for you to follow. In addition to the information in this chapter, you will also need to refer to the documentation for Analog Devices' VisualDSP++, the BittWare DSP21k-SF Toolkit, and the host PC.

To prepare your Hammerhead-PC/104-Plus board for operation,

1. Unpack the Hammerhead-PC/104-*Plus (*section 2.1*)*.
2. Set the board's configuration jumpers (section 2.2.1).
3. Configure the external serial ports (section 2.2.2).
4. Configure the board according to its location in the PC/104 stack (section 2.2.3).
5. Connect any desired external cables to the board (section 2.3).
6. Install the Analog Devices software tools (section 2.4.1).
7. Install BittWare's DSP21k-SF Toolkit (section 2.4.2).
8. Mount the board on a PC/104-*Plus* stack (section 2.4.3).
9. Run diagnostic tests on the board to ensure it is operating properly (section 2.5.1).
10. Run the example software included with the Hammerhead-PC/104-*Plus* (section 2.5.2).

2.1 Unpacking the Hammerhead-PC/104-*Plus*

Warning!

The Hammerhead-PC/104-Plus 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-PC/104-Plus 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-PC/104-*Plus*

This section explains how to set up the physical features of the board to get it ready for installation. It includes instructions for setting configuration jumpers, configuring the board's serial ports, and configuring the board according to its location in the PC/104-*Plus* stack.

2.2.1 Setting the Hammerhead-PC/104-*Plus* Configuration Jumpers

The Hammerhead-PC/104-*Plus* has eight jumpers and four switches that allow you to control and enable several of the board's features. Before mounting the board on a PC/104-*Plus* stack, make sure you have properly configured all of the jumpers and switches. Figure 2–1 shows the location of the jumpers and switch on the top of the board, and Figure 2–2 shows the location of the jumper and switches on the bottom.

Figure 2–1 *Layout of the Hammerhead-PC/104-Plus Configuration Jumpers & Switch (Top)*

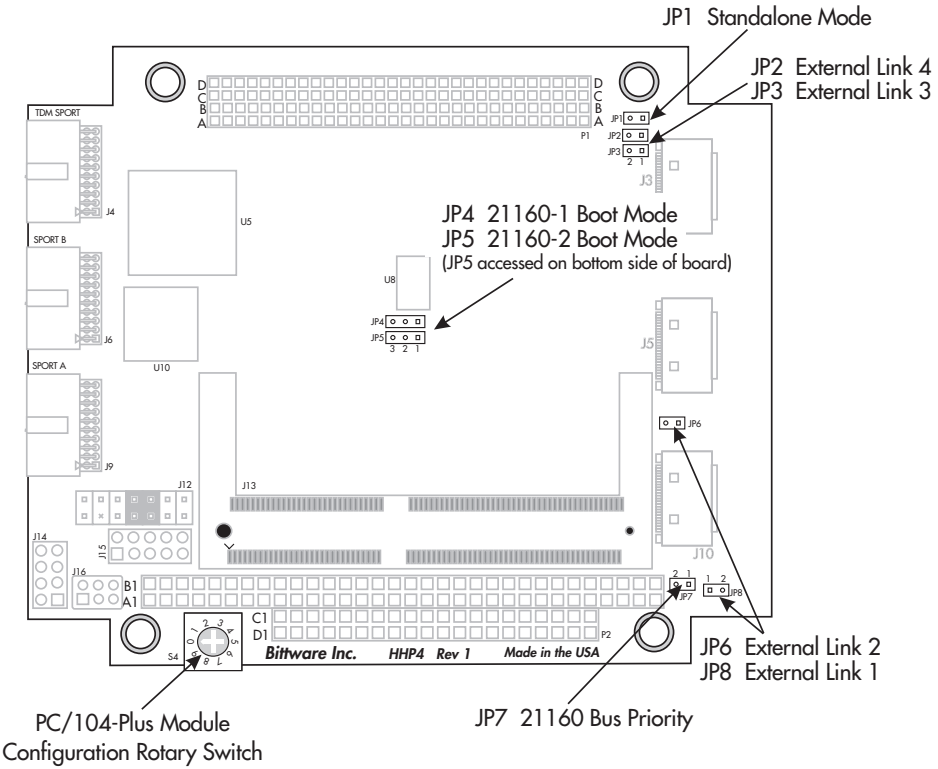
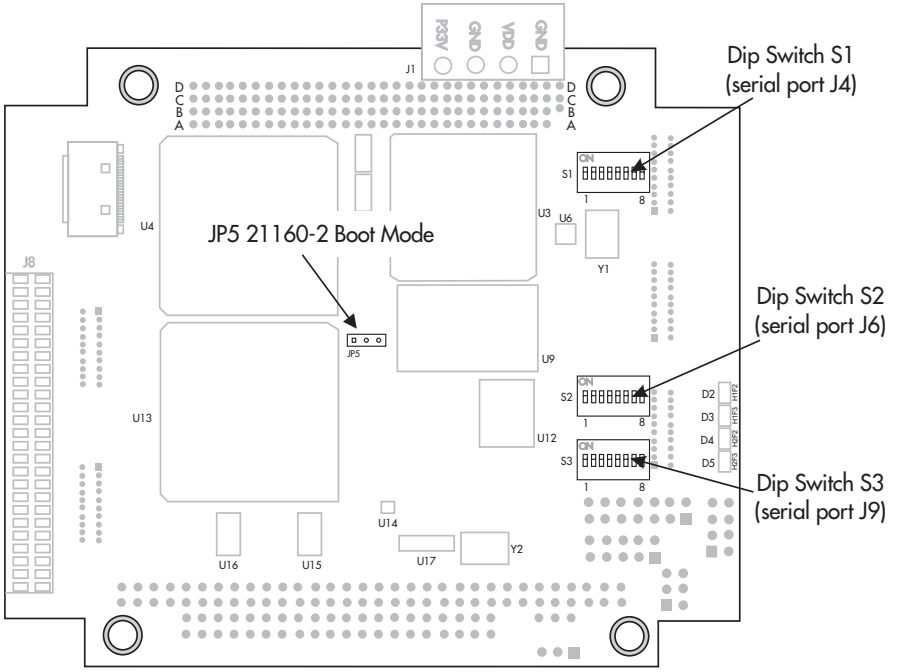


Figure 2–2 *Layout of the Hammerhead-PC/104-Plus Configuration Jumper & Switches (Bottom)*



Configuring the Board for Standalone Mode

Jumper JP1 configures the Hammerhead-PC/104-*Plus* for standalone mode. The jumper must be on for the board to operate properly in standalone mode.

Gating Power to the External Link Port Connectors

Jumpers placed on JP2, JP3, JP6 and JP8 gate power to the external link connectors on the Hammerhead-PC/104-*Plus*. These jumpers are for custom use only. Contact BittWare for more information prior to setting JP2, JP3, JP6 or JP8.

Selecting the Boot Mode

Use jumpers JP4 and JP5 to select the boot mode for the Hammerhead-PC/104-*Plus*'s ADSP-21160 processors. The processors can boot from the host PC via the 32-bit, 33 MHz PCI interface or the link ports, or they can operate in standalone mode by booting from the Flash memory. Table 2–1 shows the settings for the boot mode selection jumpers.

Note

If you have chosen the single-processor configuration of the Hammerhead-PC/104-Plus, jumper JP5 is not used.

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

Jumper	Jumper Position	Setting	Default
JP4	Pins 1–2	21160-1 will boot from on-board Flash	
	Pins 2–3	21160-1 will boot via link booting (via J11, External Link 4)	
	No Jumper	21160-1 will boot from the host computer	✔
JP5*	Pins 1–2	21160-2 will boot from the on-board Flash	
	Pins 2–3	21160-2 will boot via link booting (via 21160-1)	
	No Jumper	21160-2 will boot from the host computer	✔

* Jumper JP5 is labeled on the top side of the Hammerhead-PC/104-Plus, but the jumper is located on the bottom side.

Selecting the ADSP-21160 Bus Arbitration Mode

JP7 configures the rotating priority bus arbitration (RPBA) mode for the 64-bit ADSP-21160 cluster bus. The ADSP-21160 bus has two RPBA modes: fixed priority scheme and rotating priority scheme. The *ADSP-21160 User's Guide* (Analog Devices) explains the RPBA modes in more detail.

The fixed priority scheme for bus arbitration gives priority to the ADSP-21160 processor with the lowest multiprocessor ID. With the fixed priority scheme, 21160-1 would always have priority.

The rotating priority scheme for bus arbitration gives priority to the ADSP-21160 processors on a rotating schedule. For example, 21160-1 would have priority, then 21160-2 would have priority, and so on.

Table 2–2 *Selecting the Rotating Priority Bus Mode for the ADSP-21160 (JP7)*

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

2.2.2 Configuring the External Serial Ports

The Hammerhead-PC/104-*Plus* has three 20-pin right-angle IDC external serial port connectors to provide a communication route between the ADSP-21160s and synchronous serial devices. Each connector has a corresponding dip switch that you can use to configure the connector. Table 2–3 lists the board's serial port switches and Figure 2–2 shows where the dip switches are located on the Hammerhead-PC/104-*Plus* board.

Table 2–3 *Serial Port Dip Switch Connections*

Dip Switch	Serial Port	Description
S1	J4	Connected to both ADSP-21160 processors as a TDM serial port
S2	J6	Connected to 21160-2 as a standard serial port
S3	J9	Connected to 21160-1 as a standard serial port

Figure 2–3 illustrates the standard serial port switch settings for S2 and S3, and Table 2–4 details the settings. Figure 2–4 shows the TDM serial port switch settings for S1, and Table 2–4 details the settings.

Figure 2–3 *Standard Serial Port Dip Switch Positions (S2 and S3)*

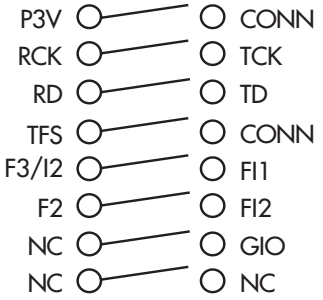


Table 2–4 *Standard Serial Port Dip Switch Settings (S2, S3)*

Switch Pins	Signal	Settings
1	P3V	ON: If using active external adapter that needs power OFF: Otherwise
2	TCK/RCK	ON: TDM mode OFF: Standard mode
3	TD/RD	ON: Connects TD and RD signals together on external connector for TDM 1-wire mode. Also connects TD and RD signals on ADSP-21160 OFF: 2-wire TDM, standard mode
4	TFS	ON: Standard with null-modem cable OFF: Otherwise
5	I2/FI1	ON: If SharcFIN is not driving IRQ2, FLAG3 will drive IRQ2 OFF: Otherwise
6	F2/FI2	ON: S2 connects ADSP-21160-2 FLAG2 to serial port 1 FLAG2 S3 connects ADSP-21160-1 FLAG2 to serial port 1 FLAG2 OFF: No connect
7	GI0	Depends on implementation
8	NC	Not connected

Figure 2–4 *TDM Serial Port Dip Switch Positions (S1)*

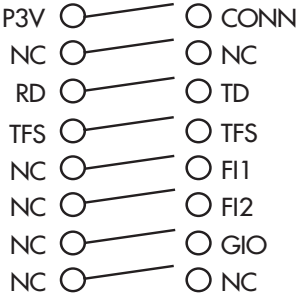


Table 2–5 *TDM Serial Port Dip Switch Settings (S1)*

Switch Pins	Signal	Settings
1	P3V	ON: If using active external adapter that needs power OFF: Otherwise
2	NC	Not connected
3	RD/TD	ON: Connects TD0 and RD0 signals together on external connector for TDM 1-wire mode. Also connects TD and RD signals on ADSP-21160. OFF: TDM 2-wire, standard mode
4	TFS	ON: Standard with null-modem cable OFF: Otherwise
5	F1	ON: TDM FLAG1 OFF: No connect
6	F2	ON: TDM FLAG2 OFF: No connect
7	GI0	Depends on implementation
8	NC	Not connected

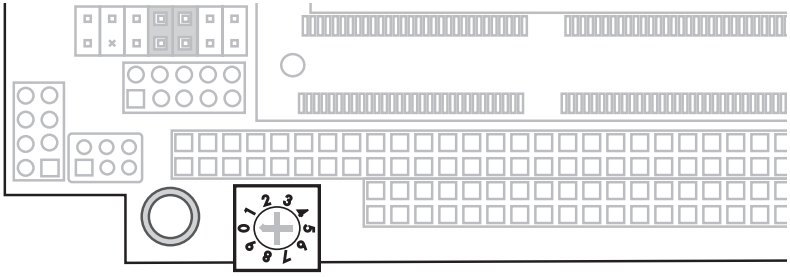
2.2.3 Configuring the Board According to Position in the PC/104-*Plus* Stack

In addition to the host CPU (carrier board), a PC/104-*Plus* stack can contain up to four PC/104-*Plus* modules. With each module added to the stack, the distance between the host CPU and the DSPs on the top module becomes greater, and clock signals can be delayed. To correct the delay in clock signals between the host CPU and the top PC/104-*Plus* modules, the Hammerhead-

PC/104-*Plus* has a rotary switch that configures the board according to its position in the stack.

The rotary switch has four settings, one for each potential position in the PC/104-*Plus* stack. For example, if the board was first in the stack, you would set the rotary switch to position 0; if the board was second in the stack, you would set the switch to position 1, and so on. Table 2–7 shows the rotary switch and its settings, and Figure 2–8 shows where the switch is located on the board.

Figure 2–5 *Rotary Switch to Select Position in PC/104 Stack (Top)*



The clocks are tuned on the host CPU so that the length of CLK3 trace is 0.662" less than CLK2, CLK2 trace is 0.662" less than CLK1, and CLK1 trace is 0.662" less than CLK0. Therefore, the first module in the stack must select CLK0 (the longest trace), the second must select CLK1, and so on. These settings provide basically no clock skew between modules. Table 2–6 gives the switch settings and signals used for each module in the stack.

Table 2–6 *Rotary Switch Settings*

Switch Position	Module Slot	$\overline{\text{REQ}}$	$\overline{\text{GNT}}$	CLK	ID Address	$\overline{\text{INT0}}$	$\overline{\text{INT1}}$	$\overline{\text{INT2}}$	$\overline{\text{INT3}}$
0 or 4	1	$\overline{\text{REQ0}}$	$\overline{\text{GNT0}}$	CLK0	AD20	$\overline{\text{INTA}}$	$\overline{\text{INTB}}$	$\overline{\text{INTC}}$	$\overline{\text{INTD}}$
1 or 5	2	$\overline{\text{REQ1}}$	$\overline{\text{GNT1}}$	CLK1	AD21	$\overline{\text{INTB}}$	$\overline{\text{INTC}}$	$\overline{\text{INTD}}$	$\overline{\text{INTA}}$
2 or 6	3	$\overline{\text{REQ2}}^*$	$\overline{\text{GNT2}}^*$	CLK2	AD22	$\overline{\text{INTC}}$	$\overline{\text{INTD}}$	$\overline{\text{INTA}}$	$\overline{\text{INTB}}$
3 or 7	4	$\overline{\text{REQ2}}^*$	$\overline{\text{GNT2}}^*$	CLK3	AD23	$\overline{\text{INTD}}$	$\overline{\text{INTA}}$	$\overline{\text{INTB}}$	$\overline{\text{INTC}}$

* Because module slots 3 and 4 share $\overline{\text{REQ2}}$ and $\overline{\text{GNT2}}$, they cannot both be bus master devices.

2.3 Connecting External Signals to the Board

This section describes how to connect external signals to the board to connect it to external devices.

2.3.1 Connecting Serial Port Cables

The Hammerhead-PC/104-*Plus* has three external serial port connectors (J4, J6, J9), which provide a communication route between the ADSP-21160 processors and other synchronous serial devices. Section 3.2.4 shows the location and pinout of the connectors. If you choose to connect the Hammerhead-PC/104-*Plus* to another serial device, the cables in Table 2–7 are available from BittWare.

Table 2–7 *Serial Port Cables Available from BittWare*

Description	BittWare Part Number
Ribbon high-density, straight-through 20-pin 8-inch cable for standard serial port on rear panel	CARH-S20-08
Ribbon high-density, straight-through 20-pin 16-inch cable for standard serial port on rear panel	CARH-S20-16
Ribbon high-density, Null 20-pin 8-inch cable for TDM serial port on rear panel	CARH-N20-08
Ribbon high-density, Null 20-pin 16-inch cable for TDM serial port on rear panel	CARH-N20-16

2.3.2 Connecting RS-232 Cables

To connect the Hammerhead-PC/104-*Plus* to your PC via the RS-232 interface, attach a 10-pin DB9-to-AMP cable to the Hammerhead-PC/104-*Plus*'s RS-232 port (J15). The cable provides a straight-through connection from the Hammerhead-PC/104-*Plus*'s UART to the PC. Since the connector's pinout (see Table 3–11) is data communication equipment (DCE), you can connect it directly to equipment configured as data terminal equipment (DTE), such as a PC, without a null-modem cable.

BittWare offers a host serial interface cable that connects the RS-232 connectors directly to a standard PC's DB9 RS-232 Com port. To connect the Hammerhead-PC/104-*Plus* to a PC with a host serial interface cable, follow the steps below.

1. If your PC's RS-232 port is DB25, you'll need to attach a DB25-to-DB9 adapter onto the DB9 end of a DB9-to-AMP 10-pin cable (available from BittWare, part number CARL-U10-06, or you can create one of your own).
2. Connect the end of the cable with the AMP 10-pin connector to J15 on the Hammerhead-PC/104-*Plus*. Be sure to line up the marked side with pin 1 on J15. Figure 3–13 shows where pin 1 is located.
3. Making sure that the PC's power is off, connect the serial interface cable to the PC.

2.3.3 Connecting an External Power Supply

When operating in standalone mode, the Hammerhead-PC/104-*Plus* requires a +3.3V and +5V power source. The external power connector (J1) supplies +3.3V, +5V and GND to the Hammerhead-PC/104-*Plus*. Section 3.2.2 gives the pinout of the connector and shows where it is located.

To connect an external power source to the Hammerhead-PC/104-Plus,

1. Plug a power adapter cable into the Hammerhead-PC/104-*Plus*'s external power connector (J1). Be sure to align pin 1 (GND) on J1 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-PC/104-*Plus* (see section 2.6).

2.3.4 Connecting an External Reset Signal to the Reset Connector

The external reset connector (J16) allows the Hammerhead-PC/104-*Plus* board to reset or be reset by other system boards. The connectors support an input reset line to allow the Hammerhead-PC/104-*Plus* to receive reset signals from other boards; it also supports an output reset line to allow the Hammerhead-PC/104-*Plus* to reset other boards. Figure 2–7 shows the cable details for the external reset circuit.

To reset the Hammerhead-PC/104-Plus with the external reset connector,

1. Connect a cable (see Figure 2–7) from the external reset connector (J16) on the Hammerhead-PC/104-*Plus* board to another system board.
2. Any reset that occurs on the Hammerhead-PC/104-*Plus* reset source causes a reset on all Hammerhead-PC/104-*Plus* reset targets.

Figure 2–6 *Cable Details for the External Reset Connector J16 (Top)*

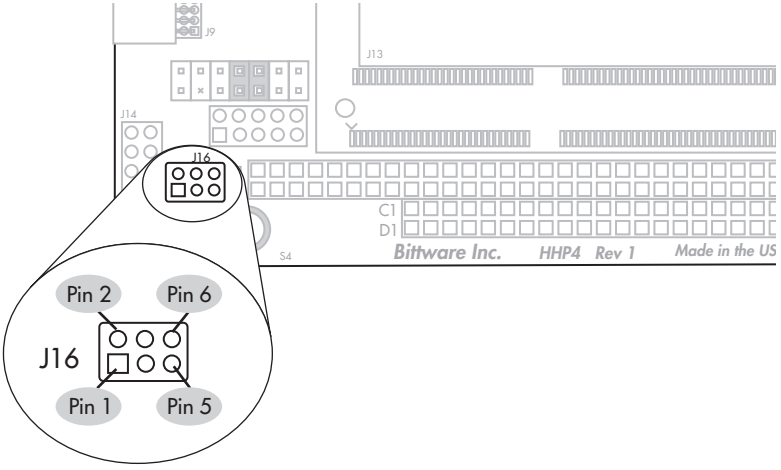
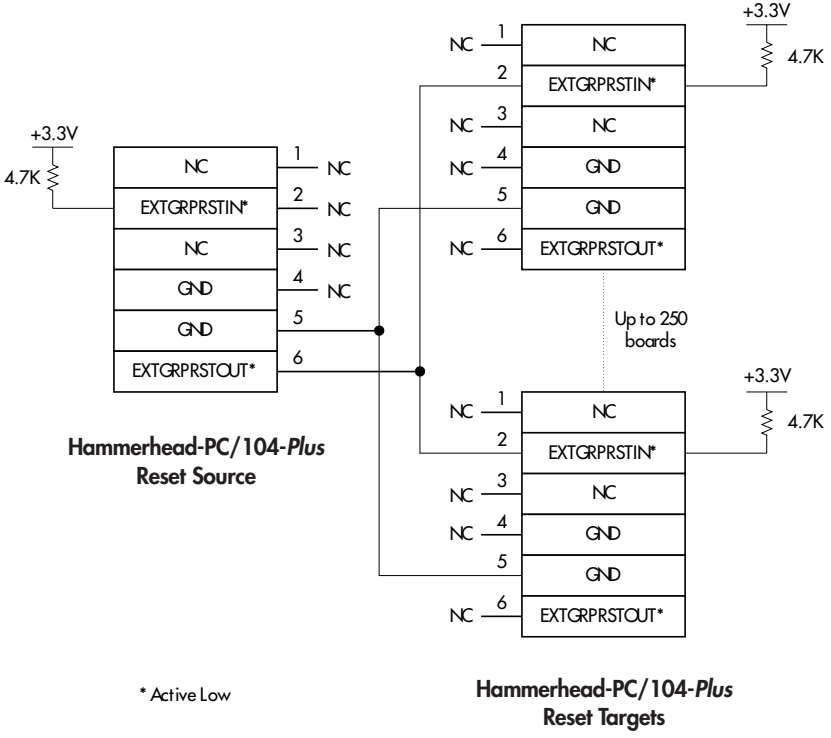


Figure 2–7 *Cable Details for the Hammerhead-PC/104-Plus External Reset Connector*



2.4 Installing the Board and its Software

This section describes the steps necessary to install the software development tools and install the board in the PC/104-*Plus* stack.

2.4.1 Installing the Analog Devices Software Tools

The Hammerhead-PC/104-*Plus* is compatible with the VisualDSP++® software development toolset from Analog Devices. VisualDSP++ is an easy-to-use project management environment comprised of an integrated development environment (IDE) and debugger. The VisualDSP++ IDE includes access to Analog Devices' SHARC C compiler, C runtime library, assembler, linker, loader, simulator, and splitter.

BittWare's VisualDSP Target allows you to use the VisualDSP++ debugger with your BittWare board. It works with the VisualDSP++ debugger to allow direct communication with the DSPs on the Hammerhead-PC/104-*Plus*. This section describes where to find installation instructions for the VisualDSP++ IDE and the BittWare VisualDSP Target.

Installing the VisualDSP++® IDE and Debugger

To install VisualDSP++, refer to the VisualDSP++ user documentation.

Installing the BittWare VisualDSP Target

If you will be using the VisualDSP++ debugger with the Hammerhead-PC/104-*Plus*, 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-PC/104-*Plus*. The *VisualDSP Target User's Guide* gives detailed installation instructions. You may also use the Analog Devices emulator in place of VisualDSP Target. Refer to Appendix A for installation instructions for the emulator.

2.4.2 Installing the BittWare DSP21k-SF Toolkit

This section gives a basic overview of installing the BittWare DSP21k-SF Toolkit; it does not give detailed instructions. 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-PC/104-*Plus* 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 programs that allow 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.

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 DSP's ability to load and run a program.

Installing the DSP21k-SF Toolkit Libraries and Utilities

To install the DSP21k-SF libraries and utilities, run the DSP21k-SF Toolkit install program. The *DSP21k-SF Toolkit User's Guide* explains the procedure in more detail.

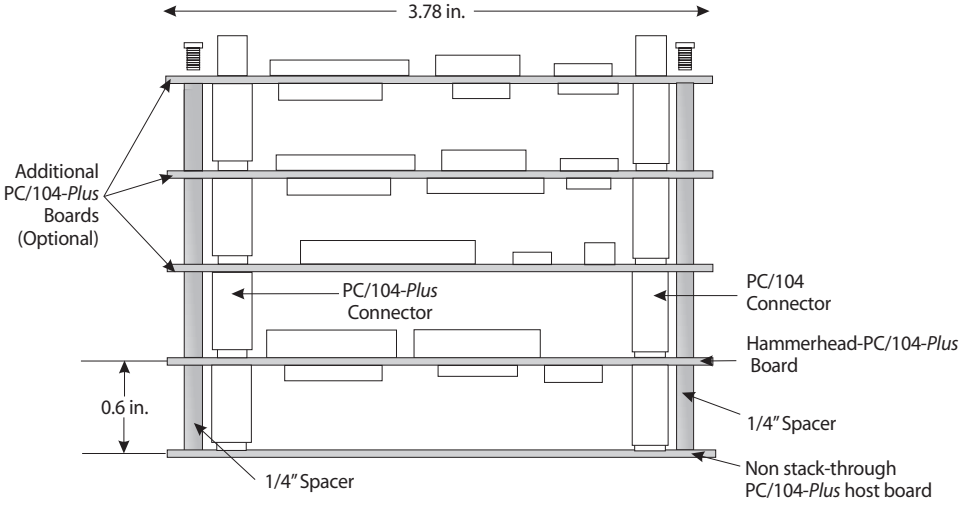
2.4.3 Mounting the Board on a PC/104-*Plus* Stack

The Hammerhead-PC/104-*Plus* plugs onto a PC/104-*Plus* stack with the ADSP-21160 processors facing the bottom of the stack.

Note

PC/104-Plus is the PCI bus, PC/104 is the ISA bus. The Hammerhead-PC/104-Plus uses the PC/104-Plus bus and passes through the PC/104 bus.

Figure 2–8 *PC/104-Plus Stack with Hammerhead-PC/104-Plus*



2.4.4 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-PC/104-*Plus* board. The *DSP21k-SF Toolkit User's Guide* explains how to run the BittWare Configuration Manager.

2.5 Testing the Board to Make Sure it is Operating Properly

This section discusses running diagnostic tests on the board after you install it to make sure it is operating properly. It runs through examples of two DSP21k-SF Toolkit diagnostic utilities and discusses the contents of the example files included with the Hammerhead-PC/104-*Plus*.

2.5.1 Testing the Board with the DSP21k-SF Toolkit Diagnostic Utilities

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

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

Testing the Board with DspBad

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

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

or

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

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

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

Testing the Board with Diag21k

The example below shows you basic Diag21k commands to test the Hammerhead-PC/104-*Plus*'s memory and load and run a DSP program. Be sure to follow the example steps below in the order in which they appear. The *DSP21k-SF Toolkit User's Manual* 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

Opened 2 DSPs.

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

Step 2: Display Board Information

Use the board information command to display information about the Hammerhead-PC/104-*Plus*'s processors.

```
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*
```

Step 3: Test the Hammerhead-PC/104-Plus's Memory

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.

Now that you have found the memory bank settings, you can test all of the Hammerhead-PC/104-*Plus*'s memory with the following commands.

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

```
diag21k[1]>br

        Board reset
```

b. Next, 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-PC/104-*Plus* board, and you are ready to load a program and execute it.

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

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

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

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

```
diag21k[1]>ps

        processor running
```

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

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

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

Step 5: Test the Remaining Processor

a. To test the second ADSP-21160 processor, select it with the DSP select command.

```
diag21k[1]>ds 2

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

b. With this 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[1]>q

        exiting...resetting processor(s)

        C:\DSP21KSF\BIN>
```

2.5.2 Testing the Board with the Hammerhead-PC/104-*Plus* Example Files

The example software provided with the Hammerhead-PC/104-*Plus* 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-PC/104-*Plus* CD-ROM.

2.6 Resetting the Hammerhead-PC/104-*Plus*

You can use the following three methods to reset the Hammerhead-PC/104-*Plus*:

- Enable the watchdog timer
- Attach an external reset switch to the board
- Send a reset signal via the PCI interface

2.6.1 Resetting the Board with the Watchdog Timer

The watchdog timer helps to ensure that the Hammerhead-PC/104-*Plus* 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 and section 6.5.4).

How the Watchdog Timer Functions When Disabled

The watchdog is disabled after the board is reset. When the watchdog is disabled, the SharcFIN 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. Section 6.5.4 explains how to disable the watchdog timer.

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 timeout time, and the next four bits determine which flag the watchdog will respond to (see Table 6–12 in 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 the chosen FLAG1 signal within the given time frame. The Watchdog Configuration

register allows you to select the watchdog's timeout time (seeTable 6–12). If the watchdog timer elapses, it will generate a system reset and the normal boot process will begin.

2.6.2 Resetting the Board via the External Reset Connector

The external reset connector (J16) allows the Hammerhead-PC/104-*Plus* board to reset or be reset by other system boards. The connector supports an input reset line to allow the Hammerhead-PC/104-*Plus* to receive reset signals from other boards; it also supports an output reset line to allow the Hammerhead-PC/104-*Plus* to reset other boards.

To reset the Hammerhead-PC/104-Plus with the external reset connector,

1. Connect a cable from the external reset connector (J16) on the Hammerhead-PC/104-*Plus* board to another system board. Refer to Figure 2–6 for cable details.
2. Any reset that occurs on the Hammerhead-PC/104-*Plus* reset source causes a reset on all Hammerhead-PC/104-*Plus* reset targets.

2.6.3 Resetting the Board via the PCI Interface

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

Chapter 3
Overview of the Hardware Components

This chapter gives a brief description of each of the Hammerhead-PC/104-*Plus*'s major hardware components, connectors, and jumpers. 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:

- ADSP-21160 SHARC processors
- Flash memory
- SDRAM
- SharcFIN ASIC
- on-board oscillators
- UART
- user LEDs
- watchdog timer
- PC/104-*Plus* module configuration rotary switch
- PC/104-*Plus* connector
- JTAG connector
- external serial ports
- external link ports
- expansion connector
- buffered inverted flag outputs
- external power connector
- external reset connector
- RS-232 connector
- configuration jumpers
- serial port configuration switches

3.1 Function and Location of the Major Components

Figure 3–1 highlights the major components on the top side of the Hammerhead-PC/104-*Plus*, and Figure 3–2 highlights the components on the bottom. The sections that follow describe the features labeled in the diagrams.

Figure 3–1 *Layout of the Hammerhead-PC/104-Plus's Major Components (Top)*

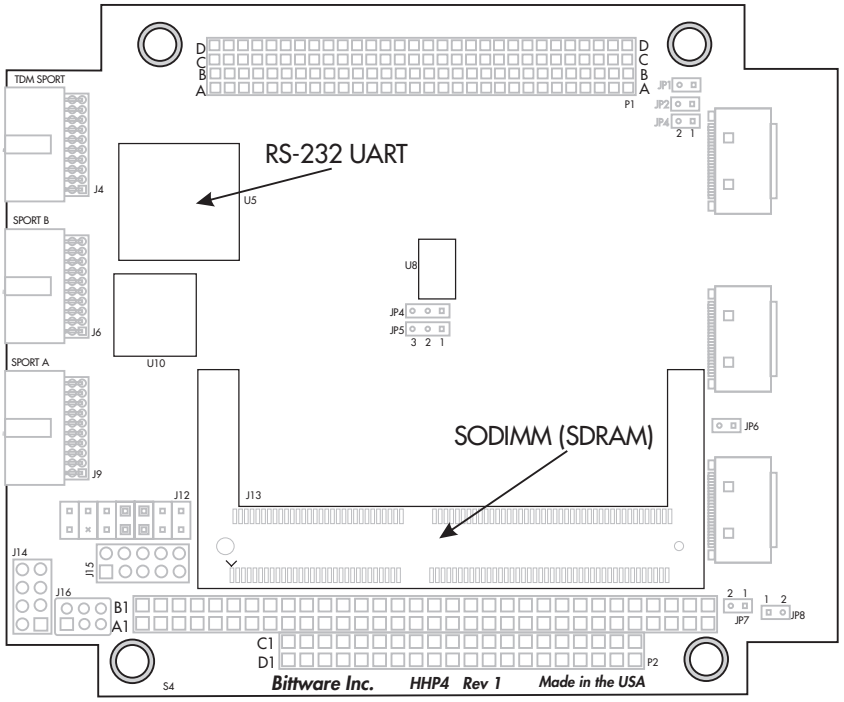
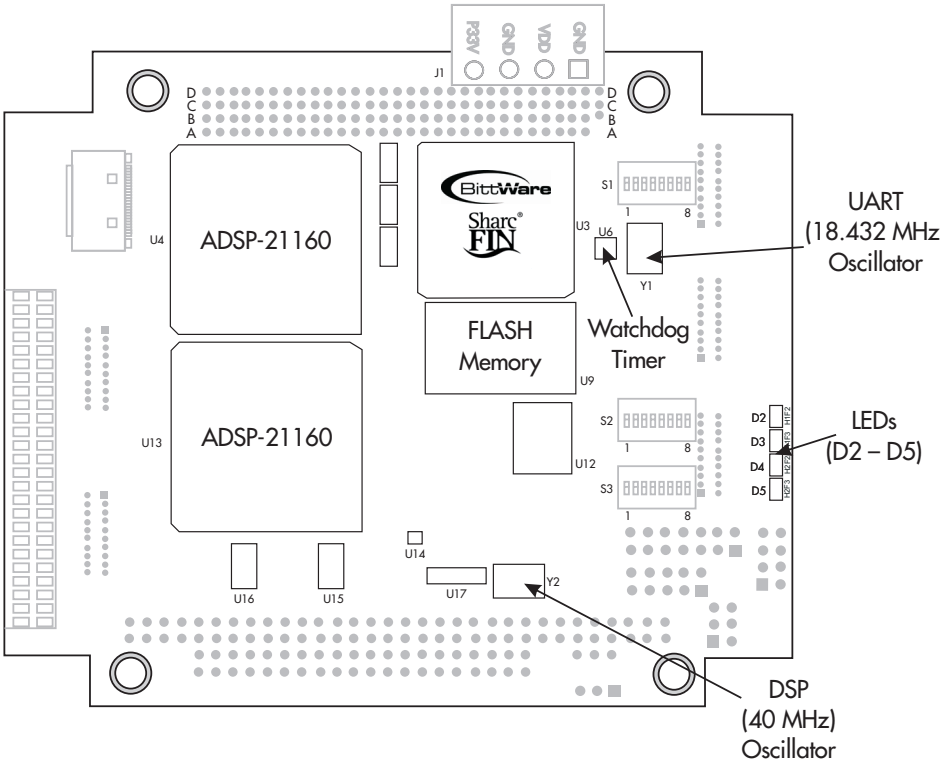


Figure 3–2 *Layout of the Hammerhead-PC/104-Plus's Major Components (Bottom)*



3.1.1 ADSP-21160 SHARC Processors

The Hammerhead-PC/104-*Plus* features two ADSP-21160 SHARC digital signal processors from Analog Devices. The DSPs feature 600 MFLOPS of processing power and operate at 80 MHz. Each processor supports two I^2S serial ports, 14 DMA channels, four flags, three interrupts, six link ports, and 4 Mbits of dual-ported on-chip SRAM.

The two processors share a common processor bus, which gives them direct access to the SDRAM and the SharcFIN. The SharcFIN provides access to the PCI interface, the Flash, and the UART.

3.1.2 External Memory Banks

Flash Memory

The board's 2 MB (2M × 8) bank of Flash memory has two uses. It stores programs that boot the ADSP-21160 processors, allowing the Hammerhead-PC/104-*Plus* to boot without a host computer (see Appendix B). It also functions as non-volatile storage space, allowing the ADSP-21160s to read, write, and erase its contents.

SDRAM

The Hammerhead-PC/104-*Plus* 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 ADSP-21160s can access the SDRAM at rates up to 40 MHz (see section 4.2).

3.1.3 SharcFIN ASIC

BittWare's SharcFIN ASIC flexibly interfaces the ADSP-21160 DSPs to a wide range of the Hammerhead-PC/104-*Plus*'s interfaces, including the 32/33 MHz PCI bus (Rev. 2.2 compliant), SDRAM, UART, I^2C serial ports, Flash, and a general purpose expansion 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:

- 32-bit, 33 MHz PCI Rev. 2.2 compliant interface (528 MB/s burst)
- Connected to the 32-bit, 40 MHz ADSP-21160 processor bus
- Connected to the peripheral bus
  - 8 bits wide @ 20 MHz
  - Accessible from the ADSP-21160 cluster bus and the PCI bus
  - Flash interface for SHARC boot and non-volatile data storage
- Four independent DMA controllers with support for fly-by and chaining (two transmit, two receive)
- 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 processor bus
- 16-byte configurable PCI mailbox registers
- I_2O™ V1.5 compliant

- Programmable interrupt multiplexer: 10 inputs, 7 outputs (1 of each dedicated to PCI)
- Standard UART and I^2C

3.1.4 On-Board Oscillators

DSP Clock

A 40 MHz system oscillator chip provides the clock for the ADSP-21160 processors. Figure 3–2 shows where it is located.

UART Clock

An 18.432 MHz oscillator chip provides the clock for the UART. Figure 3–2 shows where it is located.

3.1.5 Dual RS-232 UART

The dual RS-232 UART allows the ADSP-21160 processors to communicate with external devices, such as the host computer, via the Hammerhead-PC/104-*Plus*'s RS-232 ports. The UART is accessible via the 8-bit, 20 MHz peripheral bus.

3.1.6 User LEDs

You can use the four LEDs on the Hammerhead-PC/104-*Plus* board to indicate certain conditions in the software or to provide feedback. Each LED has a corresponding flag pin that controls it on one of the ADSP-21160 processors. Section 4.2.5 shows which LED is connected to which flag.

3.1.7 Watchdog Timer

The watchdog timer helps to ensure that the Hammerhead-PC/104-*Plus* is operating properly. It is useful for standalone applications that need to restart when certain errors occur or a program crashes. Section 6.5.4 explains how to enable and disable the watchdog timer.

3.2 Layout and Function of the External Connectors

Figure 3–3 highlights the external connectors on the top side of the Hammerhead-PC/104-*Plus*, and Figure 3–4 highlights the external connectors on the bottom. Table 3–1 gives a brief description of each connector.

Figure 3–3 *Layout of the Hammerhead-PC/104-Plus's External Connectors (Top)*

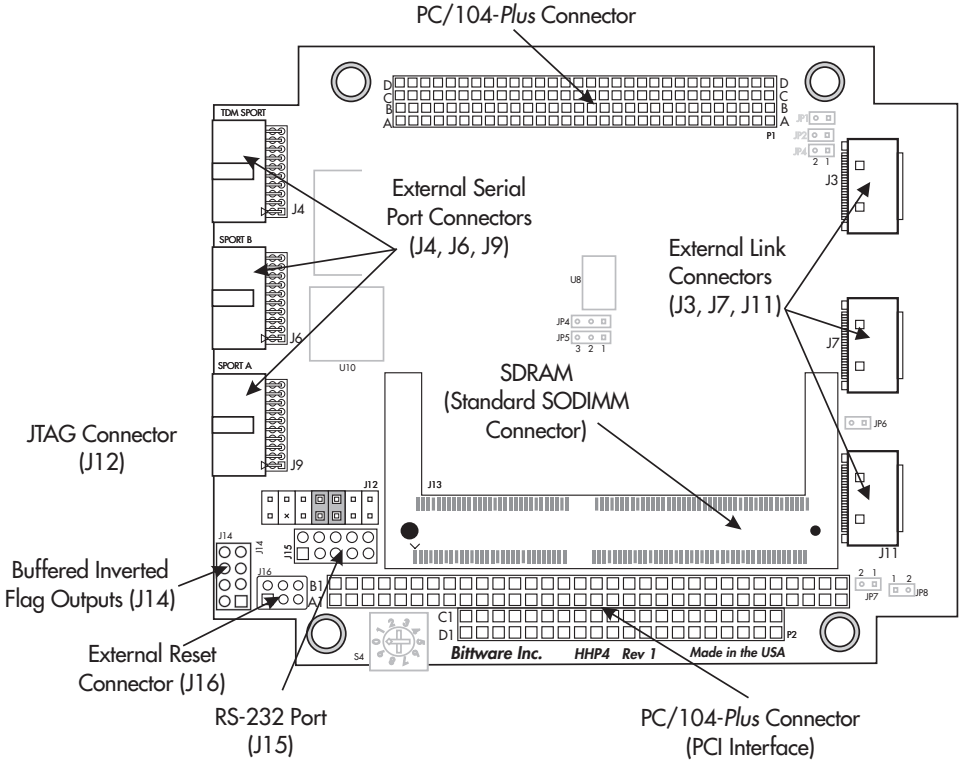


Figure 3–4 *Layout of the Hammerhead-PC/104-Plus's External Connectors (Bottom)*

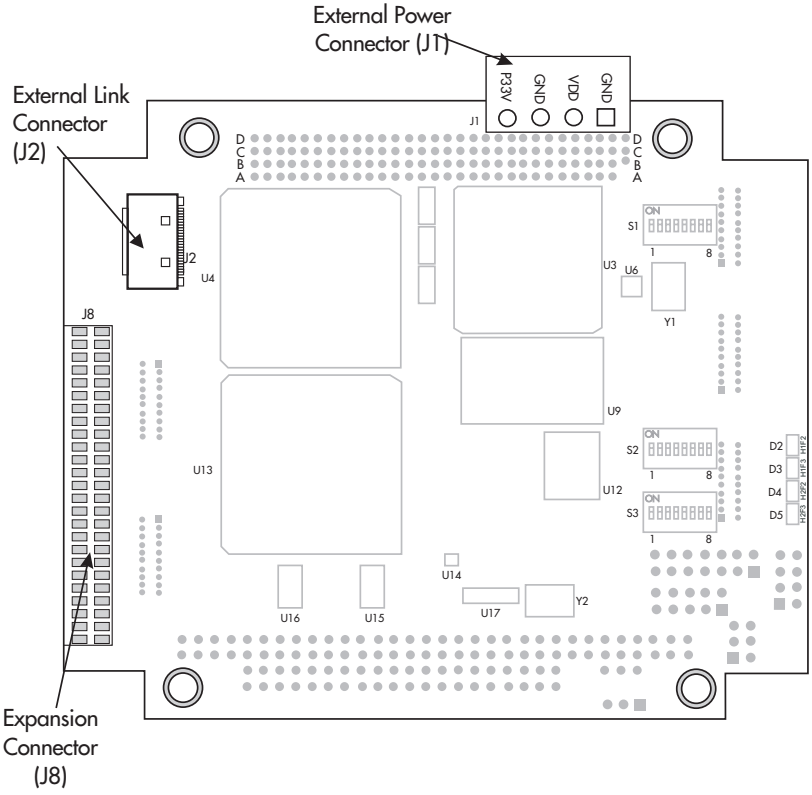


Table 3–1 *Overview of the External Connectors*

Connector	Number	Type	Description
PC/104	P1	PC/104	PC/104 stack-through connector
PC/104-*Plus*	P2	PC/104-*Plus*	PC/104-*Plus* stack-through connector with PCI bus interface
External Power	J1	4-pin	Connection for 3.3V and +5V external power supply
External Link Ports	J2, J3, J7, J11	26-pin	80 MB/s communication route between external devices and ADSP-21160s
External Serial Ports	J4, J6, J9	20-pin IDC	Communication route between ADSP-21160s and synchronous serial devices
Expansion Connector	J8	50-pin	Connection site for custom peripheral on 8-bit, 20 MHz expansion bus
JTAG	J12	14-pin	Connection for ICE in-circuit emulator
SODIMM (SDRAM)	J13	144-pin	Connection for standard 144-pin SODIMM SDRAM modules
Flag I/O	J14	16-pin	Access to FLAG2 and FLAG3 on each processor
RS-232	J15	10-pin	External serial interface to UART
External Reset	J16	6-pin	Connection for external reset signals

3.2.1 PC/104-*Plus* Connectors

The PC/104-*Plus* interface consists of two connectors: a 120-pin (4×30) connector that provides the PCI bus interface, and a standard 104-pin PC/104 connector that provides the PC/104 bus interface. The connectors are stack-through, allowing multiple PC/104-*Plus* modules in a stack. Figure 3–5 below gives the location of the PC/104-*Plus* connector pins, and Table 3–2 gives the connector pinouts. Table 3–3 gives the pinouts for the standard PC/104 connector.

Figure 3–5 *Location of the PC/104-Plus Connector Pins*

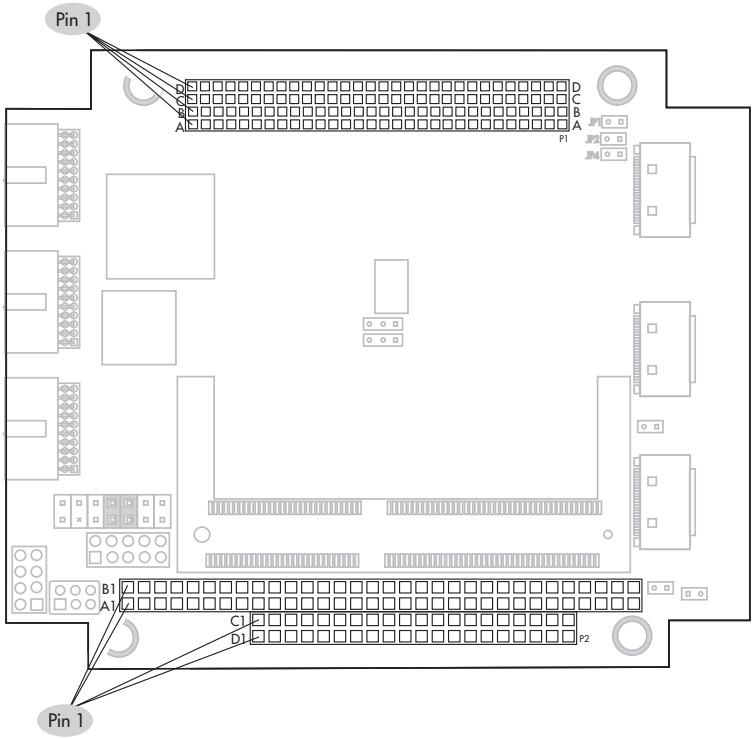


Table 3–2 *PC/104 Connector Pinout*

P1:A		P1:B		P1:C		P1:D	
A1	GND	B1	Reserved	C1	+5V	D1	AD00
A2	VI/O	B2	AD02	C2	AD01	D2	+5V
A3	AD05	B3	GND	C3	AD04	D3	AD03
A4	$\overline{\text{CBE0}}$	B4	AD07	C4	GND	D4	AD06
A5	GND	B5	AD09	C5	AD08	D5	GND
A6	AD11	B6	VI/O	C6	AD10	D6	M66EN
A7	AD14	B7	AD13	C7	GND	D7	AD12
A8	+3.3V	B8	$\overline{\text{CBE1}}$	C8	AD15	D8	+3.3V
A9	$\overline{\text{SERR}}$	B9	GND	C9	$\overline{\text{SB0}}$	D9	PAR
A10	GND	B10	$\overline{\text{PERR}}$	C10	+3.3V	D10	SDONE
A11	$\overline{\text{STOP}}$	B11	+3.3V	C11	$\overline{\text{LOCK}}$	D11	GND
A12	+3.3V	B12	$\overline{\text{TRDY}}$	C12	GND	D12	$\overline{\text{DEVSEL}}$
A13	$\overline{\text{FRAME}}$	B13	GND	C13	$\overline{\text{IRDY}}$	D13	+3.3V
A14	GND	B14	AD16	C14	+3.3V	D14	$\overline{\text{CBE2}}$
A15	AD18	B15	+3.3V	C15	AD17	D15	GND
A16	AD21	B16	AD20	C16	GND	D16	AD19
A17	+3.3V	B17	AD23	C17	AD22	D17	+3.3V
A18	IDSEL0	B18	GND	C18	IDSEL1	D18	IDSEL2
A19	AD24	B19	$\overline{\text{CBE3}}$	C19	VI/O	D19	IDSEL3
A20	GND	B20	AD26	C20	AD25	D20	GND
A21	AD29	B21	+5V	C21	AD28	D21	AD27
A22	+5V	B22	AD30	C22	GND	D22	AD31
A23	$\overline{\text{REQ0}}$	B23	GND	C23	$\overline{\text{REQ1}}$	D23	VI/O
A24	GND	B24	$\overline{\text{REQ2}}$	C24	+5V	D24	$\overline{\text{GNT0}}$
A25	$\overline{\text{GNT1}}$	B25	VI/O	C25	$\overline{\text{GNT2}}$	D25	GND
A26	+5V	B26	CLK0	C26	GND	D26	CLK1
A27	CLK2	B27	+5V	C27	CLK3	D27	GND
A28	GND	B28	$\overline{\text{INTD}}$	C28	+5V	D28	$\overline{\text{RST}}$
A29	+12V	B29	$\overline{\text{INTA}}$	C29	$\overline{\text{INTB}}$	D29	$\overline{\text{INTC}}$
A30	-12V	B30	Reserved	C30	Reserved	D30	3.3VKEY

Table 3–3 *PC/104-Plus Connector Pinout -* ***PCI Interface***

P2:A		P2:B		P2:C		P2:D	
A1	IOCHCHK	B1	GND	C0	GND	D0	GND
A2	SD7	B2	RESET	C1	SBHE	D1	$\overline{\text{MEMCS16}}$
A3	SD6	B3	VDD	C2	LA23	D2	$\overline{\text{IOCS16}}$
A4	SD5	B4	IRQ9	C3	LA22	D3	IRQ10
A5	SD4	B5	N5V	C4	LA21	D4	IRQ11
A6	SD3	B6	DRQ2	C5	LA20	D5	IRQ12
A7	SD2	B7	N12V	C6	LA19	D6	IRQ15
A8	SD1	B8	ENDXFR	C7	LA18	D7	IRQ14
A9	SD0	B9	P12V	C8	LA17	D8	$\overline{\text{DACK0}}$
A10	IOCHRDY	B10	(KEY)	C9	$\overline{\text{MEMR}}$	D9	$\overline{\text{DRQ0}}$
A11	AEN	B11	$\overline{\text{SMEMW}}$	C10	$\overline{\text{MEMW}}$	D10	$\overline{\text{DACK5}}$
A12	SA19	B12	$\overline{\text{SMEMR}}$	C11	SD8	D11	DRQ5
A13	SA18	B13	$\overline{\text{IOW}}$	C12	SD9	D12	$\overline{\text{DACK6}}$
A14	SA17	B14	$\overline{\text{IOR}}$	C13	SD10	D13	DRQ6
A15	SA16	B15	$\overline{\text{DACK3}}$	C14	SD11	D14	$\overline{\text{DACK7}}$
A16	SA15	B16	$\overline{\text{DRQ3}}$	C15	SD12	D15	DRQ7
A17	SA14	B17	$\overline{\text{DACK1}}$	C16	SD13	D16	VDD
A18	SA13	B18	$\overline{\text{DRQ1}}$	C17	SD14	D17	$\overline{\text{MASTER}}$
A19	SA12	B19	$\overline{\text{REFRESH}}$	C18	SD15	D18	GND
A20	SA11	B20	SYSCLK	C19	KEY	D19	GND
A21	SA10	B21	IRQ7				
A22	SA9	B22	IRQ6				
A23	SA8	B23	IRQ5				
A24	SA7	B24	IRQ4				
A25	SA6	B25	IRQ3				
A26	SA5	B26	DACK2				
A27	SA4	B27	TC				
A28	SA3	B28	BALE				
A29	SA2	B29	VDD				
A30	SA1	B30	OSC				
A31	SA0	B31	GND				
A32	GND	B32	GND				

3.2.2 External Power Connector

When operating in standalone mode, the Hammerhead-PC/104-*Plus* can receive power from an external power supply via the 4-pin external power connector. The external power connector (J1) supplies +3.3V, +5 VDD, and GND to the board. Figure 3–6 below shows the location of the external power connector pins, and Table 3–4 gives the pinout.

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

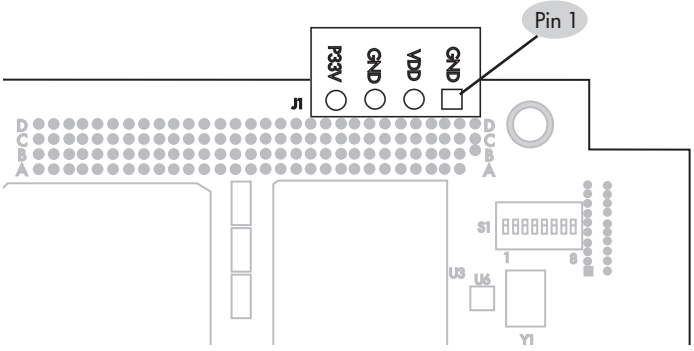


Table 3–4 *External Power Connector Pinout (J1)*

Pin	Signal
1	GND
2	VDD
3	GND
4	3.3V

3.2.3 External Link Ports

The Hammerhead-PC/104-*Plus* has four 26-pin external link port connectors: J2, J3, J7, and J11. The external link ports allow you to connect the Hammerhead-PC/104-*Plus* directly to other boards.Table 3–5 gives the connector pinout; Figure 3–7 shows the location of the link pins on the top side of the board, and Figure 3–8 shows the link pins on the bottom.

Table 3–5 *Link Port Connector Pinout (J2, J3, J7, and J11)*

J2		J3		J7		J11	
Pin	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**	**Pin**	**Signal**
1	NC	1	NC	1	NC	1	NC
2	EA_L4CLK	2	EA_L3CLK	2	EA_L2CLK	2	EA_L1CLK
3	EA_L4ACK	3	EA_L3ACK	3	EA_L2ACK	3	EA_L1ACK
4	EA_L4DAT0	4	EA_L3DAT0	4	EA_L2DAT0	4	EA_L1DAT0
5	EA_L4DAT1	5	EA_L3DAT1	5	EA_L2DAT1	5	EA_L1DAT1
6	EA_L4DAT2	6	EA_L3DAT2	6	EA_L2DAT2	6	EA_L1DAT2
7	EA_L4DAT3	7	EA_L3DAT3	7	EA_L2DAT3	7	EA_L1DAT3
8	EA_L4DAT4	8	EA_L3DAT4	8	EA_L2DAT4	8	EA_L1DAT4
9	EA_L4DAT5	9	EA_L3DAT5	9	EA_L2DAT5	9	EA_L1DAT5
10	EA_L4DAT6	10	EA_L3DAT6	10	EA_L2DAT6	10	EA_L1DAT6
11	EA_L4DAT7	11	EA_L3DAT7	11	EA_L2DAT7	11	EA_L1DAT7
12	NC	12	NC	12	NC	12	NC
13	NC	13	NC	13	NC	13	NC
14	GND	14	GND	14	GND	14	GND
15	GND	15	GND	15	GND	15	GND
16	GND	16	GND	16	GND	16	GND
17	GND	17	GND	17	GND	17	GND
18	GND	18	GND	18	GND	18	GND
19	GND	19	GND	19	GND	19	GND
20	GND	20	GND	20	GND	20	GND
21	GND	21	GND	21	GND	21	GND
22	GND	22	GND	22	GND	22	GND
23	GND	23	GND	23	GND	23	GND
24	GND	24	GND	24	GND	24	GND
25	GND	25	GND	25	GND	25	GND
26	JP2	26	JP4	26	JP7	26	JP9

Figure 3–7 *Location of External Link Connectors J2, J7, and J11 (Top)*

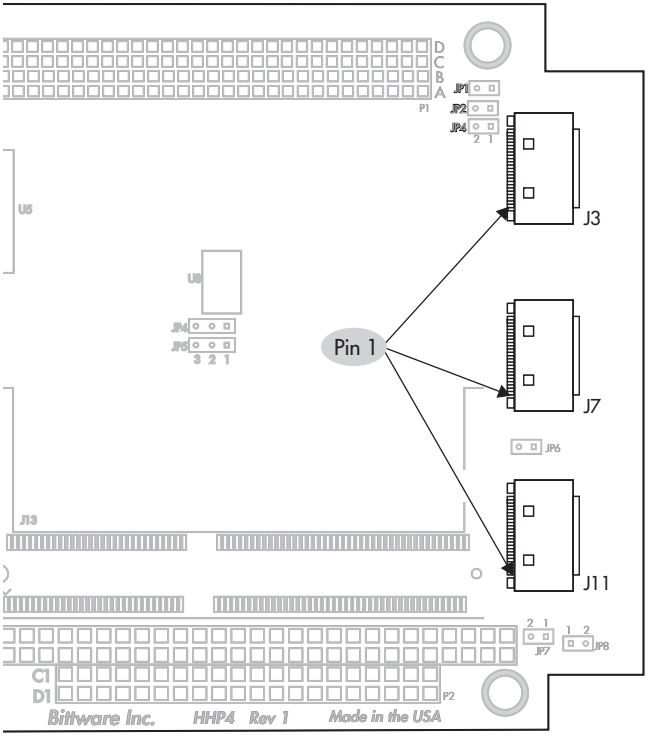
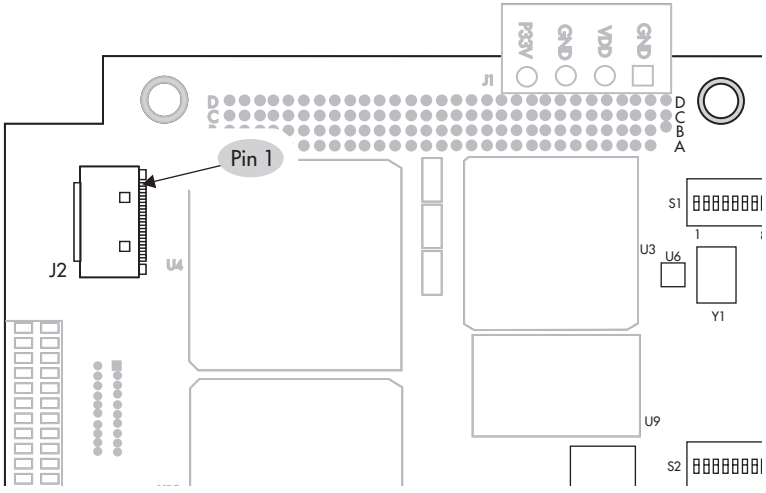


Figure 3–8 *Location of External Link Connector J3 (Bottom)*



3.2.4 External Serial Ports

Three 20-pin, 50mm right-angle IDC external serial port connectors (J4, J6, J9) provide a communication route between the ADSP-21160s and synchronous serial devices at rates up to 40 Mbits/s. The serial ports on the Hammerhead-PC/104-*Plus* conform to BittWare's "*Universal Serial Port Specification*" (available from BittWare upon request).

External Serial Port Connections and Usage

Each serial port has a corresponding dip switch that allows you to configure its signal connections.Table 3–6 below lists the connections for each serial port and the dip switches that correspond to each serial port.

Table 3–6 *External Serial Port Connections and Usage*

Serial Port	Dip Switch	Description
J4	S1	Connected to both ADSP-21160 processors as a TDM serial port
J6	S2	Connected to 21160-2 as a standard serial port
J9	S3	Connected to 21160-1 as a standard serial port

Serial Port Connector Pinout

Figure 3–9 shows where the pins are located on the serial port connectors, and Table 3–7 gives their pinouts. The signals in the pinout tables come directly from the ADSP-21160 pins they are associated with, and they exhibit the same timing characteristics described in the *ADSP-21160 User's Manual* (Analog Devices, Inc.). The signals are series terminated with 82Ω resistors.

Figure 3–9 *Location of the External Serial Port Connector Pins (Top)*

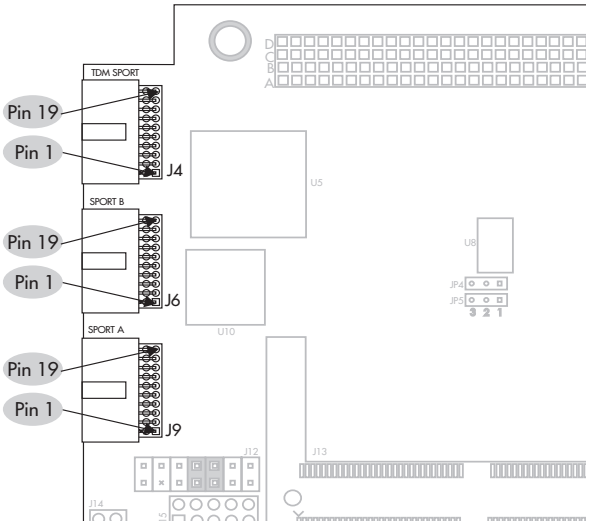


Table 3–7 *External Serial Port Connector Pinout (J4, J6, J9)*

Pin	Signal	Pin	Signal
1	P33V	2	GND
3	RCK	4	GND
5	RFS	6	GND
7	RD	8	GND
9	RD2/FI1	10	GND
11	GND	12	TD2/FI2
13	GND	14	TD
15	GND	16	TFS
17	GND	18	TCK
19	GND	20	GIO

3.2.5 Expansion Connector

The Hammerhead-PC/104-*Plus* features an expansion connector site that provides access to the board's 8-bit, 20 MHz peripheral bus. This connector is not populated except by special order, but Table 3–10 shows the location of the pins, and Table 3–8 gives the pinout for the connector site. Contact BittWare for more information before using the expansion connector.

Figure 3–10 *Location of the Expansion Bus Connector Pins (Bottom)*

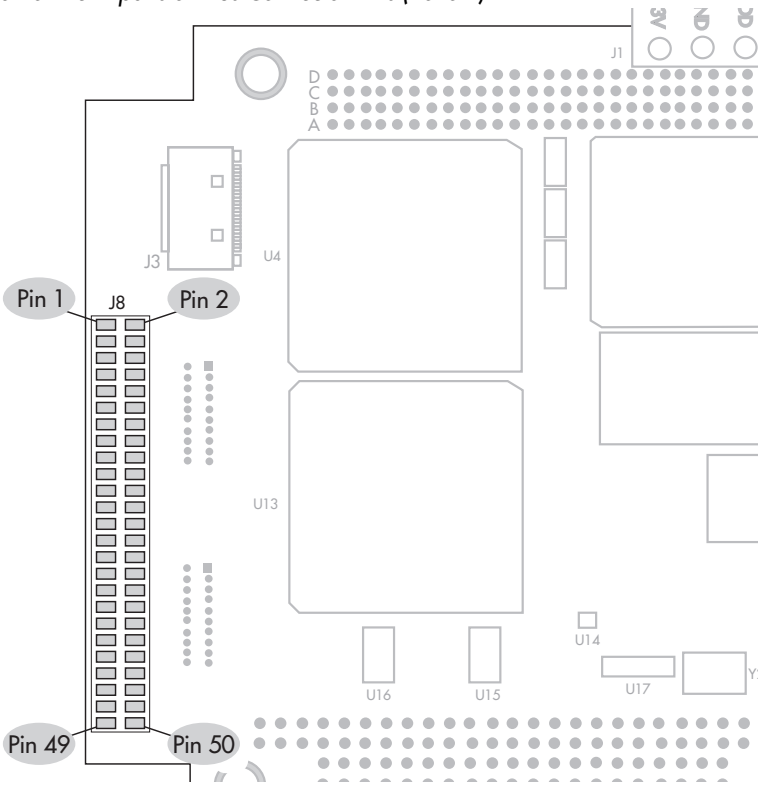


Table 3–8 *Peripheral Bus Connector Pinout*

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

3.2.6 JTAG Connector

The JTAG connector (J12) is a 14-pin connector that allows you to attach an optional Analog Devices ICE emulator for program debugging. Appendix A overviews installing and using in-circuit emulators with the Hammerhead-PC/104-*Plus*. Refer to Figure 3–11 below for the location of the JTAG connector pins and to Table 3–9 for the connector pinout.

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

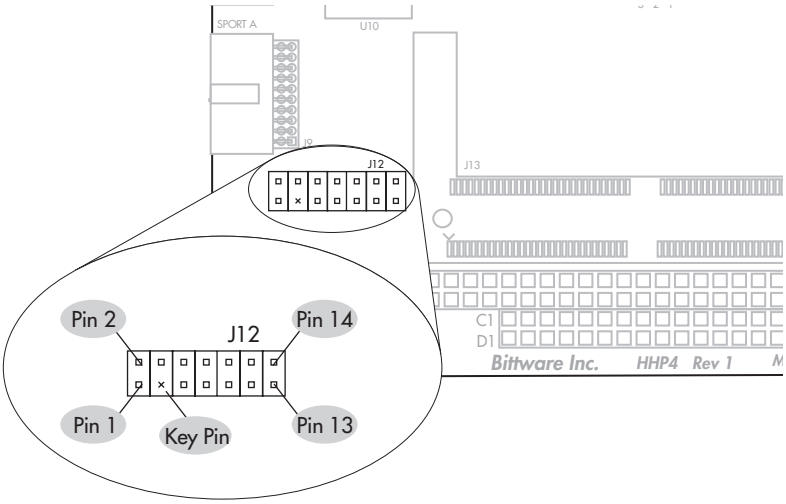


Table 3–9 *JTAG Connector Pinout (J12)*

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

3.2.7 SDRAM (SODIMM)

The Hammerhead-PC/104-*Plus* has an industry-standard 144-pin connection for a standard SODIMM module. The SODIMM modules are available in 64, 128, 256, and 512 MB modules. Figure 3–3 shows where the connector pins are located on the board.

3.2.8 Flag I/O Connector

The 8-pin flag I/O connector (J14) allows access to the FLAG2, FLAG3, and IRQ2 signals on each ADSP-21160 DSP. 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 3–12 shows where the pins are located on the connector, and Table 3–10 gives the connector pinout.

Figure 3–12 *Location of the Flag I/O Connector Pins (Top)*

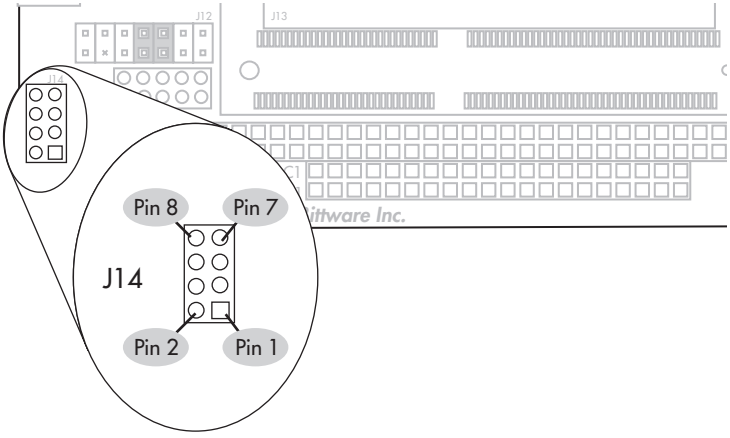


Table 3–10 *Flag I/O Connector Pinout*

Pin	Signal	Description	Pin	Signal
1	HA1_F2	21160-1 FLAG2	2	GND
3	HA1_F3 HA2_I2	21160-1 FLAG3 21160-2 IRQ2	4	GND
5	HA2_F2	21160-2 FLAG2	6	GND
7	HA2_F3 HA1_I2	21160-2 FLAG3 21160-1 IRQ2	8	GND

3.2.9 RS-232 Connector

The Hammerhead-PC/104-*Plus* is configured with a 10-pin RS-232 connector (J15). The connector transports serial data between the host and the UART, which interfaces the data to the ADSP-21160 processors. Figure 3–13 shows the location of the RS-232 connector pins, and Table 3–11 gives the connector pinout. An RS-232 cable is available from BittWare, part number CARL-U10-06, or you may provide your own.

Figure 3–13 *Location of the RS-232 Connector Pins (Top)*

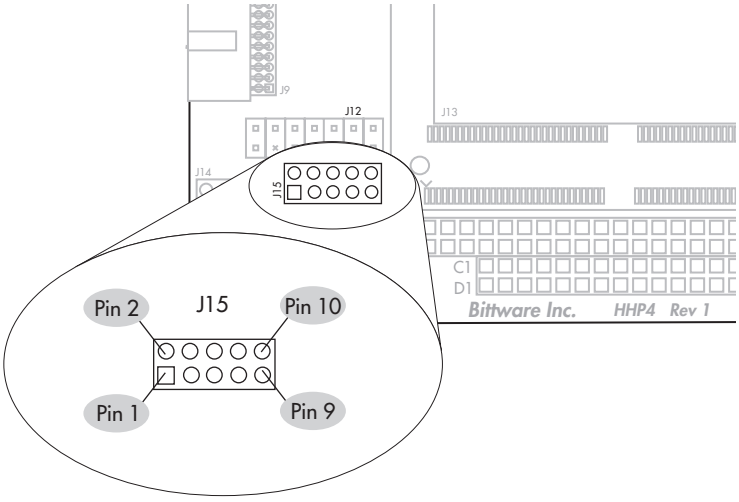


Table 3–11 *RS-232 Connector Pinout (J15)*

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

* Pins 1, 2, and 7 are connected together on the board.

3.2.10 External Reset Connector

The external reset connector (J16) allows the Hammerhead-PC/104-*Plus* board to reset or be reset by other system boards. The connector supports an input reset line to allow the Hammerhead-PC/104-*Plus* to receive reset signals from other boards; it also supports an output reset line to allow the Hammerhead-PC/104-*Plus* to reset other boards. Figure 3–14 shows the location of the external reset connector pins.

Figure 3–14 *External Reset Connector Pins (Top)*

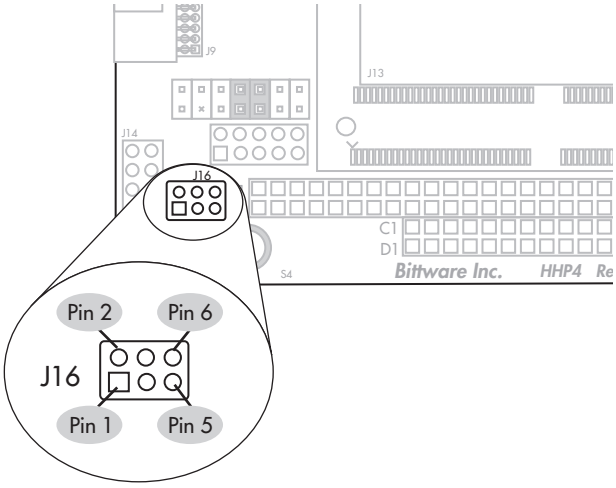


Table 3–12 *External Reset Connector Pinout (J16)*

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

3.3 Function and Location of the Configuration Jumpers and Switches

The Hammerhead-PC/104-*Plus* has nine configuration jumpers and four dip switches that allow you to control and enable certain features on the board. Before installing and operating the Hammerhead-PC/104-*Plus*, make sure to properly set all of the configuration jumpers. Table 3–13 below gives an overview of the jumpers, Table 3–14 gives an overview of the dip switches, and section 2.2 describes their settings in detail. Figure 3–15 and Figure 3–16 show the locations of the configuration jumpers and switches.

Table 3–13 *Hammerhead-PC/104-Plus Configuration Jumpers*

Jumper	Name	Type	Description
JP1	Standalone mode	2-pin	Jumper must be on to operate board in standalone mode
JP2, JP3, JP6, JP8	External link port power	2-pin	Gates power to the external link port connectors*
JP4	21160-1 boot mode jumper	3-pin	Sets 21160-1 to boot from host computer, on-board Flash, or remote processor via link port
JP5	21160-2 boot mode jumper	3-pin	Sets 21160-2 to boot from host computer, on-board Flash, or remote processor via link port
JP7	21160 rotating priority bus arbitration	2-pin	Determines fixed or rotating priority for the ADSP-21160 cluster bus

* These jumpers are for custom use only. Contact BittWare for more information prior to setting JP2, JP3, JP6 or JP8.

Table 3–14 *Hammerhead-PC/104-Plus Serial Port Dip Switches*

Dip Switch	Name	Type	Description
S1	Serial port dip switch for J4	8-pos dip	Configures serial port J4 as a TDM serial connection to both ADSP-21160 DSPs
S2	Serial port dip switch for J6	8-pos dip	Configures serial port J6 as a standard serial connection to 21160-2 (SPORT B)
S3	Serial port dip switch for J9	8-pos dip	Configures serial port J9 as a standard serial connection to 21160-1 (SPORT A)
S4	PC/104-*Plus* module configuration rotary switch	10-pos rotary	Configures the board according to its position in the PC/104-*Plus* stack

Figure 3–15 *Location of the Configuration Jumpers and Switches (Top)*

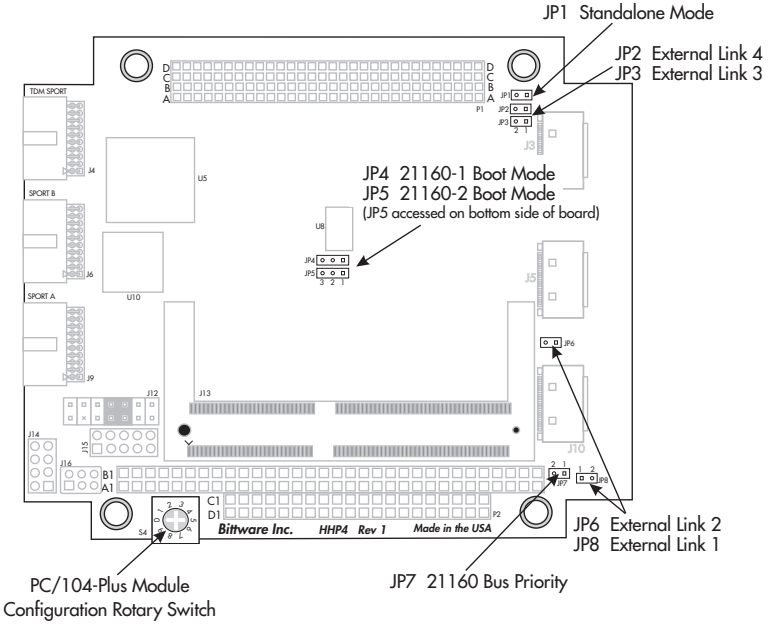
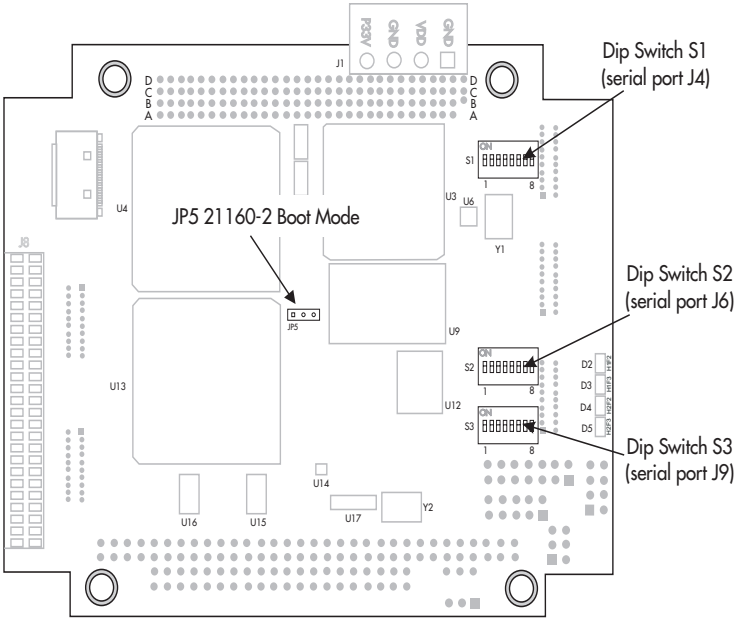


Figure 3–16 *Location of the Configuration Jumpers and Switches (Bottom)*



4

Chapter 4
Hammerhead-PC/104-*Plus* Board Architecture

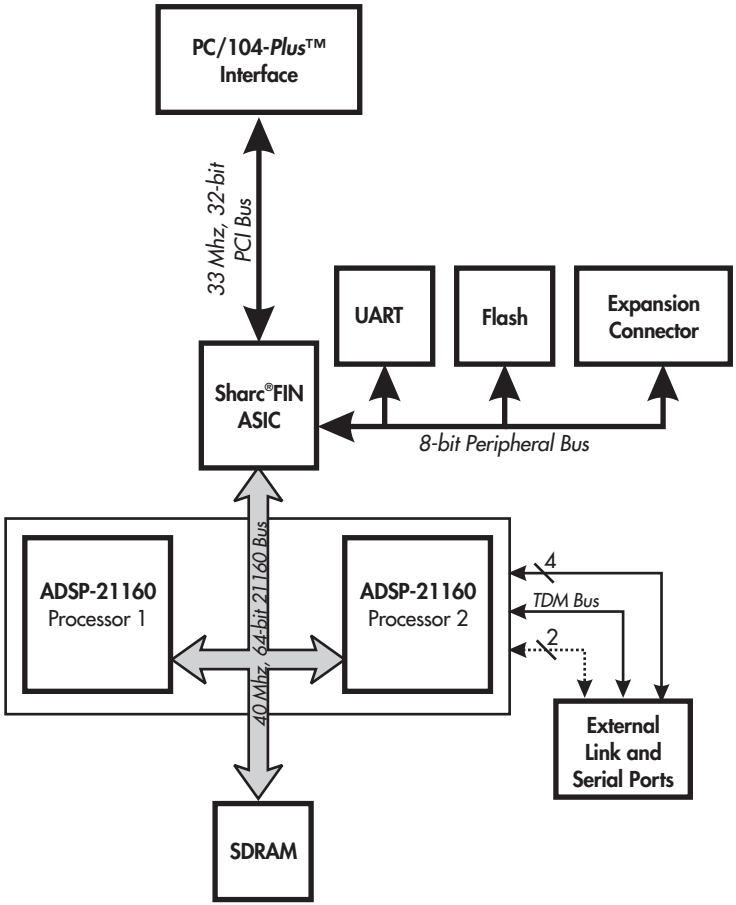
This chapter discusses the architecture of the board, describing how the ADSP-21160 DSPs communicate with other DSPs, with the host, and with other I/O peripherals. This chapter covers the following topics:

- 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' 32-bit cluster bus
- the structure of the PCI interface (including the SharcFIN ASIC, PCI bus, and peripheral bus)

4.1 Overview of the Board Architecture

This section briefly describes how data flows through the Hammerhead-PC/104-*Plus* board. The sections that follow discuss the board's architecture in more detail.

Figure 4–1 *Block Diagram of the Hammerhead-PC/104-Plus System*



The Hammerhead-PC/104-*Plus* features two ADSP-21160 DSPs operating at 80 MHz. The processors communicate via the 64-bit, 40 MHz processor bus and have access to a bank of up to 512 MB SDRAM, the SharcFIN ASIC, 2 MB bank of Flash memory, an RS-232 UART, and an expansion connector.

There are three buses on the Hammerhead-PC/104-*Plus*: the ADSP-21160 cluster bus, the PCI bus, and the peripheral bus. The 64-bit, 40 MHz ADSP-21160 cluster bus connects the ADSP-21160 DSPs and provides access to the bank of SDRAM. It also connects to the SharcFIN ASIC, which provides a bridge between the DSPs and the 32-bit, 33 MHz PCI bus. A peripheral bus extends off the SharcFIN, providing access to the UART, Flash memory, and the expansion connector.

For I/O options, the board features three serial port connectors, four link connectors, an RS-232 port, and an expansion connector. Two of the serial connectors are standard serial ports, the third is a serial TDM port. The RS-232 port connects to the UART to allow serial communication with the ADSP-21160 DSPs, and the expansion connector provides access to the 8-bit, 20 MHz peripheral bus.

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: memory banks, flags and interrupts, serial ports, and link ports. The following tables summarize how the DSPs' resources are used on the Hammerhead-PC/104-*Plus*. The row labeled "MS" refers to the DSPs' external memory select lines (MS0–MS3).

Table 4–1 *Resources for 21160-1*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM Bus (J4)	SharcFIN*	SharcFIN	21160-2 L0
1	Flash	External (J9)	21160-2 F1†	SharcFIN	External J2
2	SharcFIN		21160-2 F3 Flag I/O (J14)	SharcFIN (as HA3_F0)‡ SPORT A J9 LED D3	21160-2 L4
3	UART			SharcFIN (as HA3_F1) 21160-2 I2 LED D5 Flag I/O (J14)	21160-2 L5
4					External J11
5					21160-2 L2

* IRQ0, FLAG0, FLAG1, FLAG2, and FLAG3 on each processor connect to the SharcFIN ASIC interrupt multiplexer. From the SharcFIN, you can route them to different locations on the board.

† IRQ1 is hardwired within the SharCFIN.

‡ See the Note on page 97 for a complete description of these flag connections.

Table 4–2 *Resources for 21160-2*

	MS	Serial Port	Interrupt	Flag	Link Port
0	SDRAM	TDM Bus (J4)	SharcFIN*	SharcFIN	21160-1 L0
1	Flash	External (J6)	21160-1 F1†	SharcFIN	External J7
2	SharcFIN		21160-1 F3 Flag I/O (J14)	SharcFIN (as HA4_F0)‡ SPORT B J6 LED D4	21160-1 L5
3	UART			SharcFIN (as HA4_F1) 21160-1 I2 Flag I/O (J14) LED D2	External J3
4					21160-1 L2
5					21160-1 L3

* IRQ0, FLAG0, FLAG1, FLAG2, and FLAG3 on each processor connect to the SharcFIN ASIC interrupt multiplexer. From the SharcFIN, you can route them to different locations on the board.

† IRQ1 is hardwired within the SharCFIN.

‡ See the Note on page 97 for a complete description of these flag connections.

4.2.2 ADSP-21160 Memory Structure

This section describes the memory structure of the ADSP-21160 DSPs. The processors can access their own internal memory, the internal memory of other processors in the same cluster, and external memory devices. The sections below describe each type of memory.

Internal Memory

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

Multiprocessor Memory

Multiprocessor memory space (MMS) is the on-chip SRAM of the other ADSP-21160 DSPs in the same cluster. A cluster is up to six ADSP-21160 DSPs that share a common processor bus, and any DSP that is connected to the processor bus shares the MMS. The Hammerhead-PC/104-*Plus* has two DSPs on board that share a common bus (the ADSP-21160 cluster bus); each processor can view the other DSP's on-chip SRAM.

External Memory

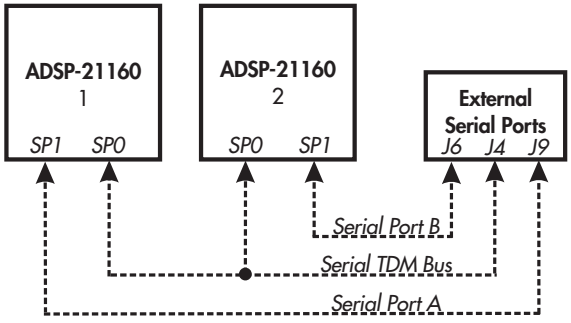
External memory space consists of other devices that share the ADSP-21160's 64-bit cluster bus. The external memory space is divided into four banked sections of memory. Each ADSP-21160 DSP has four memory select lines, MS0–MS3, which allow it to access the external memory banks located on the 64-bit ADSP-21160 cluster bus and on the 8-bit peripheral bus.

4.2.3 Serial Port Connections

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

The serial ports operate at 1/2 the full clock rate of the processor, or 40 MHz. The serial ports support independent transmit and receive functions and can automatically transfer serial port data to and from on-chip memory using DMA block transfers. Figure 4–2 below shows the serial port connections on the Hammerhead-PC/104-*Plus* board.

Figure 4–2 *Block Diagram of Serial Port Connections*



Serial Port Connections to the Serial TDM Bus

SP0 on both ADSP-21160 processors is connected to external connector J4 as a TDM serial port. Switch S1 configures the signals on the external connector. Section 2.2.2 explains how to set the dip switch (see Figure 2–4 and Table 2–4).

Serial Port Connections to the External Serial Ports

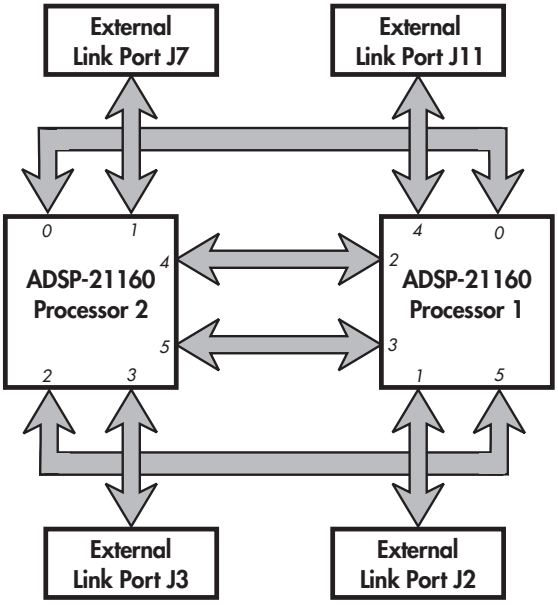
Serial port 1 on 21160-1 is connected to external connector J9 as a standard serial port. Switch S3 configures the external connector. Section 2.2.2 explains how to set the switch (see Figure 2–3 andTable 2–4).

Serial port 1 on 21160-2 is connected to external connector J6 as a standard serial port. Switch S2 configures this serial port. Section 2.2.2 explains how to set the switch (see Figure 2–3 and Table 2–4).

4.2.4 Link Port Connections

Each ADSP-21160 DSP has six 8-bit, 80 Mbyte/sec link ports, which allow it to communicate with the other DSP on the Hammerhead-PC/104-*Plus* board and with DSPs on other boards. The link ports are also available for link booting. The link ports are bidirectional and can operate at frequencies up to the same speed as the DSP's internal clock, letting each port transfer up to 8 bits of data per internal clock cycle. Each processor has two external link ports that allow you to interface directly with other boards. The remaining four link ports are connected to the other DSP for interprocessor communication. Figure 4–3 below shows how the link ports on the two DSPs are connected.

Figure 4–3 *Block Diagram of Link Port Connections*

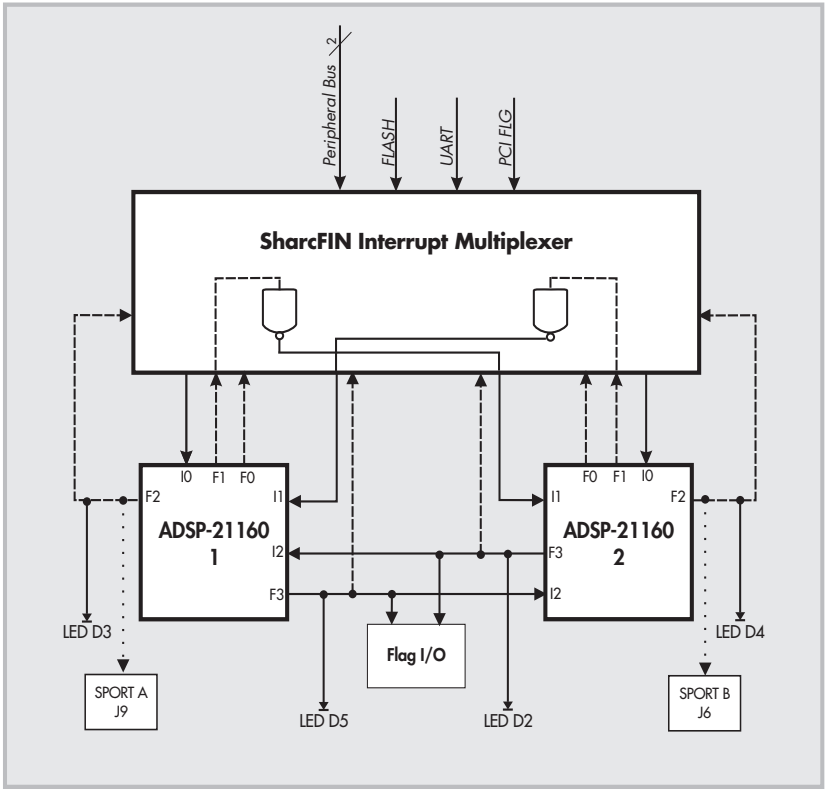


4.2.5 Flag and Interrupt Connections

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

All of the flags and one interrupt from each DSP connect to the SharcFIN ASIC. Using registers in the SharcFIN, you can change the routing of those flags and interrupts (see section 6.5.11). The flags and interrupts also connect to the other DSP, to LEDs, to rear panel I/O, and to a flag I/O connector. Figure 4–4 below illustrates the connections.

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



SharcFIN Flags and Interrupts

Each ADSP-21160 DSP has four flag signals, and all are connected to the SharcFIN ASIC. Using registers in the SharcFIN's configuration space, you can configure the routing of those flags. Additionally, two flags from each DSP connect to LEDs, external serial ports, and the flag I/O connector. Using the SharcFIN interrupt multiplexer, you can configure the DSPs to receive interrupts from the following sources:

- peripheral bus
- PCI bus
- UART
- other DSPs in the same cluster

Table 4–3 and Table 4–4 show the flag and interrupt connections on the Hammerhead-PC/104-*Plus*.

Table 4–3 *ADSP-21160 Flag Connections*

	21160-1	**21160-2**
FLAG0	SharcFIN*	SharcFIN
FLAG1	SharcFIN (via 21160-2 I2)	SharcFIN (via 21160-1 I2)
FLAG2	SharcFIN (as H3_F0) SPORT A J9 LED D3	SharcFIN (as H4_F0) SPORT B J6 LED D4
FLAG3	SharcFIN (as H3_F1) 21160-2 I2 Flag I/O J14 LED D5	SharcFIN (as H4_F1) 21160-1 I2 Flag I/O J14 LED D2

* FLAG0 – FLAG3 on each DSP connect to the SharcFIN ASIC interrupt multiplexer. From the SharcFIN, you can route them to different locations on the board.

Table 4–4 *ADSP-21160 Interrupt Connections*

	21160-1	**21160-2**
IRQ0	SharcFIN*	SharcFIN
IRQ1	21160-2 F1	21160-1 F1
IRQ2	21160-2 F3 Flag I/O J14	21160-1 F3 Flag I/O J14

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

Interprocessor Flags and Interrupts

Each DSP can generate and receive interrupts to and from the other DSP on the board. Interrupts IRQ2 and IRQ3 on each processor receive interrupts from FLAG1 and FLAG3 respectively from the other processor. IRQ1 on 21160-1 is hardwired to FLAG1 on 21160-2, and IRQ1 on 21160-2 is hardwired to FLAG1 on 21160-1. Table 4–4 and Figure 4–4 show the flag and interrupt connections for each DSP.

Flag Connections to LEDs and Test Points

Two flags from each DSP (FLAG2 and FLAG3) and one interrupt (IRQ2) connect to LEDs, and a flag I/O connector. These flags are useful for testing the board. FLAG2 and FLAG3 connect to LEDs D2 – D5. FLAG3 and IRQ2 on both processors connect to the flag I/O connector J14.

Flag and Interrupt Connections to the Rear Panel

FLAG2 from each DSP connects to rear panel I/O. 21160-1 is connected to external serial ports J9, and 21160-2 is connected to serial port J6.

Jumpered Flag Connections

Two switches configure FLAG2 to connect to the SharcFIN ASIC interrupt multiplexer. Dip 5 of switch S3 configures 21160-1 FLAG3, and dip 5 of switch S2 configures 211602 FLAG3. These switches configure whether the ADSP-21160 flags will connect to the SharcFIN ASIC interrupt multiplexer. Section 2.2.2 and Table 2–4 explain the switch settings in more detail.

SharcFIN Interrupts

The SharcFIN ASIC can generate interrupts to and from the flag registers, interrupt registers, UART, peripheral bus and PCI bus. The routing of any input interrupt is user-configurable to any output.

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

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

4.2.6 ADSP-21160 Cluster Bus

The Hammerhead-PC/104-*Plus* has a 64-bit, 40 MHz ADSP-21160 cluster bus that connects the two ADSP-21160 processors and a bank of up to 512 MB SDRAM. It connects 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 SharcFIN bridge.

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

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

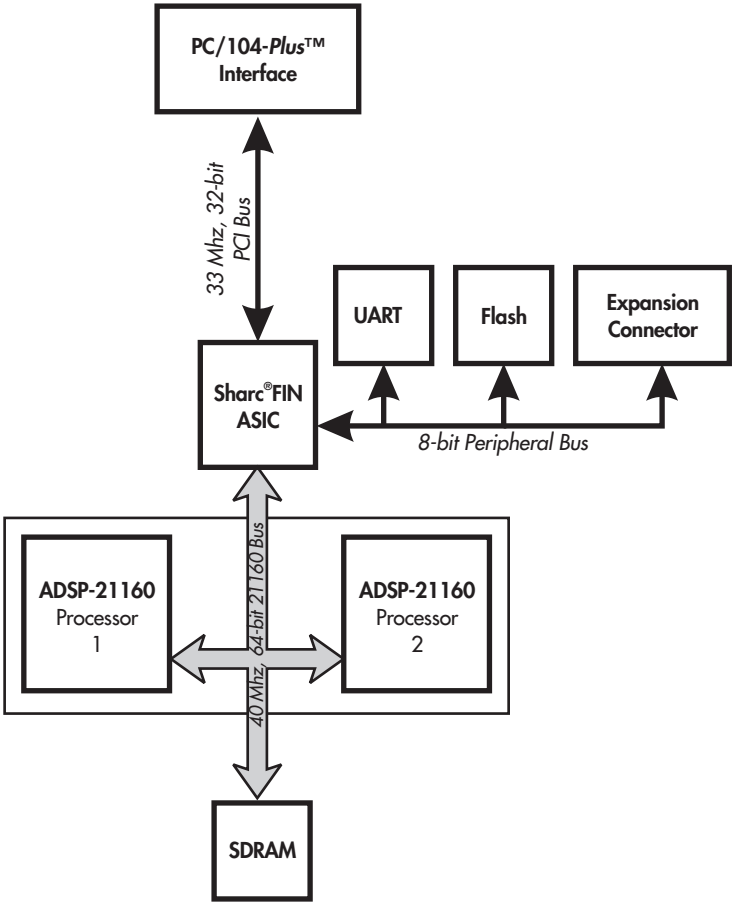
- $\overline{\text{BMS}}$[1]
- MS1

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

4.3 PCI Interface Architecture

The Hammerhead-PC/104-*Plus*'s PCI interface consists of the PCI bus, the SharcFIN ASIC, and the peripheral bus. The SharcFIN ASIC connects the 33 MHz, 32-bit PCI bus with the 40 MHz, 64-bit ADSP-21160 bus. An 8-bit peripheral bus is also connected to the SharcFIN, providing access to the Flash memory, the UART, and the expansion connector. Figure 4–5 is a diagram of the PCI interface.

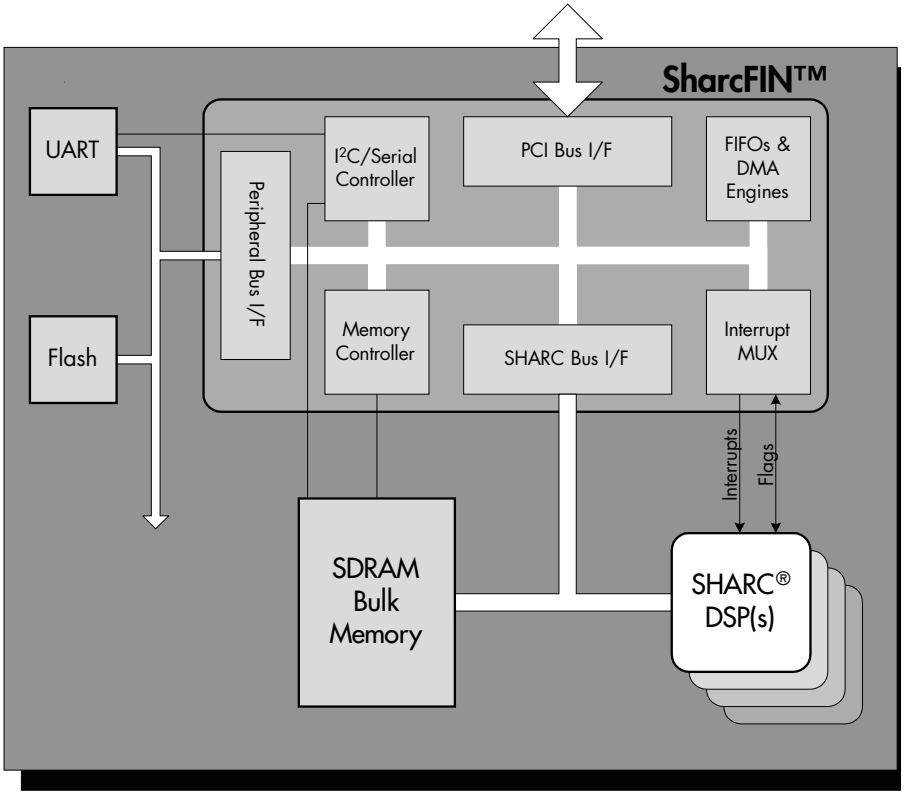
Figure 4–5 *Block Diagram of the PCI Interface*



4.3.1 SharcFIN Architecture

The Hammerhead-PC/104-*Plus* features a BittWare SharcFIN ASIC. The SharcFIN 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-PC/104-*Plus* 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 400MB/sec.

Interface to PCI

The second function of the SharcFIN is to interface to PCI. The SharcFIN implements a full 32-bit/33MHz 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-PC/104-*Plus*.

I^2C Interface

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 32-bit, 33 MHz PCI bus interface connects the host to the SharcFIN ASIC, which acts as a bridge between the PCI interface and the ADSP-21160 processors.

4.3.3 Peripheral Bus Interface

The 20 MHz, 8-bit peripheral bus connects to low-speed peripherals such as 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 bus. It connects to the SharcFIN, which interfaces it to the ADSP-21160 DSPs and the PCI interface. The peripheral bus operates at either 3.3 volts or 5 volts.

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 DSP and the host to communicate.We do not provide the programs themselves; rather, we give programming details specific to the Hammerhead-PC/104-*Plus*. You will also need to refer to documentation for the DSP21k-SF Toolkit, the ADSP-21160 SHARC processor, the Analog Devices development tools, and the SharcFIN ASIC.

5.1 Accessing the DSP's Memory

5.1.1 Accessing Internal Memory

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

5.1.2 Accessing the DSP's 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.Table 5–1 below provides the address region and processor ID variable for each DSP.

Table 5–1 Multiprocessor Memory Allocation for DSP

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 bus. On the Hammerhead-PC/104-*Plus*, these devices include the SDRAM, the Flash, and the SharcFIN ASIC.

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	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 section 2.4.4) 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-PC/104-Plus*

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-PC/104-*Plus* 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-PC/104-*Plus* board, the unbanked memory space is unused.

5.2 Booting the DSP

This section explains the three booting options for the Hammerhead-PC/104-*Plus*: link port, Flash, and PCI.

5.2.1 Booting the Board via Link Port

In link port booting, the DSP gets boot data from another DSP's link port or from a four- or 8-bit wide external device[1] after system power-up. Link 4 of 21160-1 is connected to external link port J11. After booting via link port, 21160-1 will boot the remaining DSP on the board.

To boot the Hammerhead-PC/104-Plus via link port,

1. Develop a boot program using Analog Devices VisualDSP.
2. Using the external link ports, load the boot program onto the DSP. 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.
5. "Selecting the Boot Mode" on page 15 describes how to set the board for booting via the link port.

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-PC/

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.

104-*Plus*, the Host Interface Library (HIL) and Diag21k contain functions that will perform the boot process.

To boot the Hammerhead-PC/104-Plus 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 DSP.
3. Use the HIL's *dsp21k_start* function or Diag21k's Processor Start (**ps**) command to start executing the program.
4. "Selecting the Boot Mode" on page 15 describes how to set the board's jumpers for host booting.

To boot the DSP 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 DSP via the DSP's 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 DSP's 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.

5.2.3 Booting the Board via the Flash

The Flash memory allows you to boot the Hammerhead-PC/104-*Plus* 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 processor. The example software included with the

Hammerhead-PC/104-*Plus* 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 DSP is reset.

Loading a Boot Program into the Flash Memory

Use programs provided with the Hammerhead-PC/104-*Plus* to load the Flash memory with a boot program. See "Selecting the Boot Mode" on page 15 for details on setting the board for Flash booting.

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-PC/104-Plus board together. It flexibly interfaces the ADSP-21160 cluster to the PCI bus, SDRAM, Flash, dual UART, and I^2C; provides interrupt multiplexers for the ADSP-21160s and PCI; controls the SDRAM; and provides DMA engines for moving data between interfaces.

6.1.1 The Two Sections of the SharcFIN

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

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

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

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

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

6.2 Function of the SharcFIN PCI Interface

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

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

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

6.2.1 Performing PCI Side DMAs

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

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

6.2.2 Performing a PCI Side Single Access

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

6.2.3 Performing PCI Side Interrupts

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

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

Note

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

6.3 Function of the SharcFIN SHARC Interface

The SHARC interface of the SharcFIN consists of the ADSP-21160 bus interface, the SDRAM controller, the peripheral bus (with Flash and dual UART), the I^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-PC/104-Plus 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^2C Interface

The SDRAM and configuration EEPROM sit on an I^2C bus that is connected to the SharcFIN. The SDRAM is interrogated over the I^2C 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-PC/104-Plus 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 address	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	0x1000000 – 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 – 0x13FFFFFF	Third 16 MB block of SDRAM
0x01	0x00000 – 0x3FFFFF	0x000000 – 0xFFFFFF	0x1400000 – 0x17FFFFFF	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.

Note

The SharcFIN ASIC used on this dual-processor Hammerhead-PC/104-Plus board is identical to those used on BittWare's quad-processor boards. As a result, there are registers that retain designations for 21160-3 (H3) and 21160-4 (H4) within the SharcFIN. Those registers are accessed by 21160-1 (H1) and 21160-2 (H2) on the Hammerhead-PC/104-Plus, as follows:

- 21160-1 FLAG2 accesses H3_Flag0
- 21160-1 FLAG3 accesses H3_Flag1
- 21160-2 FLAG2 accesses H4_Flag0
- 21160-2 FLAG3 accesses H4_Flag1

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-PC/104-Plus 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			
5:4	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
7:6	H4F1 En, H3F1 En†	WORM	0000	Selects which processor 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.

† These bits are accessed by 21160-1 and 21160-2 as described in the Note on page97. Therefore, 21160-1 FLAG3 accesses H3F1 and 21160-2 FLAG3 accesses H4F1.

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 I²C Control register (offset 0x46) controls the I²C interface. The I²C interface is a two-wire bus; one wire is a clock signal, and the other is a data signal. Both the clock and the data lines are pulled up. Table 6–14 shows the contents of the register.

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

Table 6–14 *Contents of the I²C 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 I²C line

6.5.7 PMC+ I²C Control Register

The PMC+ I²C Control register (offset 0x47) controls the I²C interface to the PMC+ interface. The settings for this register are the same as the settings for the Onboard I²C register, except that all settings apply to the PMC+ interface I²C instead of the on-board I²C.

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.

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).

* 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, 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			
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.

† These bits are accessed by 21160-1 and 21160-2 as described in the Note on page97. Therefore, 21160-1 FLAG2 accesses H3F0, 21160-1 FLAG3 accesses H3F1, 21160-2 FLAG2 accesses H4F0, and 21160-2 FLAG3 accesses H4F1.

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			
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.

† These bits are accessed by 21160-1 and 21160-2 as described in the Note on page97. Therefore, 21160-1 FLAG2 accesses H3F0, 21160-1 FLAG3 accesses H3F1, 21160-2 FLAG2 accesses H4F0, and 21160-2 FLAG3 accesses H4F1.

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.

† These bits are accessed by 21160-1 and 21160-2 as described in the Note on page97. Therefore, 21160-1 FLAG2 accesses H3F0, 21160-1 FLAG3 accesses H3F1, 21160-2 FLAG2 accesses H4F0, and 21160-2 FLAG3 accesses H4F1.

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		

* These bits are accessed by 21160-1 and 21160-2 as described in the Note on page97. Therefore, 21160-1 FLAG2 accesses H3F0, 21160-1 FLAG3 accesses H3F1, 21160-2 FLAG2 accesses H4F0, and 21160-2 FLAG3 accesses H4F1.

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

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

Appendix A
Debugging Your DSP Programs

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

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

This section discusses attaching an in-circuit emulator (ICE) from Analog Devices to the Hammerhead-PC/104-*Plus* board. To attach an ICE to the Hammerhead-PC/104-*Plus*, follow the steps below:

1. Connect the probe on the ICE card to the Hammerhead-PC/104-*Plus*'s JTAG connector.
2. Depending on the type of ICE you are using, either install it in or connect it to your PC.
3. Set up the Hammerhead-PC/104-*Plus* for operation.
4. Apply power to the Hammerhead-PC/104-*Plus*.
5. Start the emulator software on the PC.

A.1.1 Overview of the ICE Emulator

The Hammerhead-PC/104-*Plus* is compatible with Analog Devices's ICE emulators, which are separate ISA bus, PCI bus, Ethernet, or USB cards that connect to the Hammerhead-PC/104-*Plus*'s JTAG connector and connect to your PC. The emulator provides a controlled environment for observing, debugging, and testing real-time activities in a target hardware environment by connecting directly to the target processor through its JTAG interface.

When the ICE is attached to the Hammerhead-PC/104-*Plus*, the Hammerhead-PC/104-*Plus* becomes the target system for the emulator, allowing you to operate it completely from the emulator's user interface. A powerful tool for debugging

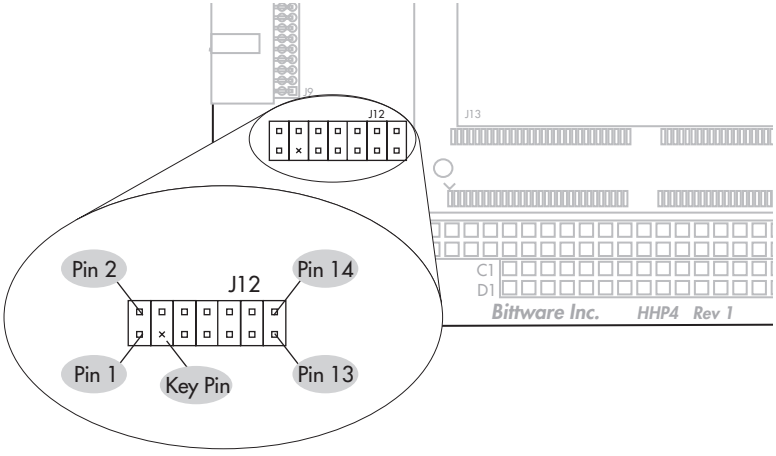
programs running on the ADSP-21160 processors, the emulator monitors system behavior while running at full speed, and you can use it to download programs, start and stop program execution, set breakpoints, and observe and change the contents of the registers and memory.

A.1.2 Attaching the ICE to the Hammerhead-PC/104-*Plus*

To attach the ICE to the Hammerhead-PC/104-Plus, follow the instructions below.

1. Locate the JTAG connector (J12) on the Hammerhead-PC/104-*Plus* (see Figure 3–3).
2. A cable extends from the ICE card to a probe that connects to the JTAG connector on the Hammerhead-PC/104-*Plus*. Connect the ICE probe to the JTAG connector. Figure A–1 shows the location of the pins on the connector.

Figure A–1 *JTAG Connector*



Pin 3 on the JTAG connector is missing (see Figure A–1) to prevent you from installing the emulator incorrectly. One of the sockets in the ICE probe has a plug inserted in place of the pin. Table 3–9 in Chapter 3 shows the connector pinout.

A.1.3 Installing the ICE and Hammerhead-PC/104-*Plus* In a PC

Once you have connected the ICE to the Hammerhead-PC/104-*Plus*, install the emulator in your host computer. The install procedure for the emulator depends on the platform you are using: ISA, PCI, Ethernet, or USB. Refer to the *Emulator Hardware and Software Installation Guide* from Analog Devices for instructions on installing the emulator.

A.1.4 Operating the ICE

To start operating the ICE with the Hammerhead-PC/104-Plus,

1. Apply power to the Hammerhead-PC/104-*Plus*.

Note

As long as the emulator software is not running, you can safely attach and remove the ICE probe while the Hammerhead-PC/104-Plus is running.

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

A.2 Debugging with a Software Emulator

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

A.2.1 About the VisualDSP Target

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

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

A.2.2 Installing the VisualDSP Target

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

Appendix B
Setting Up for Standalone Operation

The Hammerhead-PC/104-*Plus* can boot from a boot program stored in its Flash memory, which allows it to operate in standalone mode, free from a host computer. This section lists the steps necessary to prepare your Hammerhead-PC/104-*Plus* board to operate in standalone mode.

Note

If you are not planning to operate the Hammerhead-PC/104-Plus in standalone mode, follow the instructions in Chapter 2.

1. While in development mode, develop a boot loader and a standalone operating program for the DSPs (see B.1.1).
2. Program the Flash with the boot loader and standalone operating program (see B.1.2).
3. Power down and place a jumper on Standalone Mode JP1 to operate the board in standalone mode (see B.2.1.).
4. Set the Hammerhead-PC/104-*Plus*'s boot mode to "Flash boot" or "link boot" (see "Selecting the Boot Mode" on page 15).
5. Mount standoffs on the board (see B.3).
6. Apply an external power source to the board (see B.4).
7. Boot the board in either Flash or link boot mode (see B.5)

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-PC/104-*Plus* 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-PC/104-*Plus* to load the Flash memory with a boot program. The `flash` directory contains utilities that provide easy access to the Flash memory on the Hammerhead-PC/104-*Plus* board. The `flash` directory also contains a test program that uses the utilities.

B.1.3 Loading a Link Boot Program

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

B.2 Configuring the Board for Standalone Mode

B.2.1 Setting the Standalone Mode Jumper

To operate the board in standalone mode, power down the board and set standalone mode jumper JP1. Place the jumper across the two pins of JP1 as described in "Configuring the Board for Standalone Mode" on page 14. See also Figure 2–1 on page 13 for the location of the pins.

B.2.2 Selecting the Boot Mode

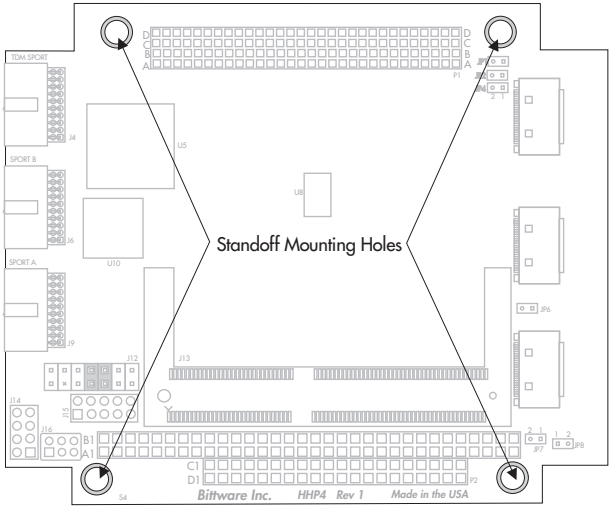
To operate in standalone mode, the Hammerhead-PC/104-*Plus* must be set to boot from a boot program stored in the on-board Flash memory. Configuration jumpers JP4 and JP5 allow you to select the Hammerhead-PC/104-*Plus*'s boot mode (see section "Setting the Hammerhead-PC/104-Plus Configuration Jumpers" on page 13 for instructions).

B.3 Mounting Standoffs on the Board

If you are not mounting the Hammerhead-PC/104-*Plus* on a PC/104-*Plus* carrier board (see section 2.4), mount standoffs on the board to provide proper support and adequate clearance between the bench and the Hammerhead-PC/104-*Plus*'s components.

1. Place a ¼" standoff in each of the four standoff mounting holes on the Hammerhead-PC/104-*Plus*. Figure B–1 shows where the holes are located.
2. Secure each standoff with a ¼" screw.

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



B.4 Connecting an External Power Supply

When operating in standalone mode, the Hammerhead-PC/104-*Plus* requires a +5V power source. The external power connector (J1) supplies +3.3V, +5V and GND to the Hammerhead-PC/104-*Plus*. Section 3.2.2 gives the pinout of the connector and shows where it is located.

To connect an external power source to the Hammerhead-PC/104-*Plus*,

1. Plug a power adapter cable into the Hammerhead-PC/104-*Plus*'s external power connector (J1). Be sure to align pin 1 (+5V) on the cable with the +5V pin on J1.
2. Connect the remaining end of the cable to an external power source, such as a switching standalone power supply or the PC's power supply.
3. Apply power to the system.
4. Reset the Hammerhead-PC/104-*Plus* (see section 2.6).

B.5 Booting the Board

Connect a cable to link port 4, external connector J11 (Table 3–5, on page 47). J11 provides a link port connection for booting 21160-1. When booting via link port, 21160-1 will boot and then boot 21160-2.

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 be better 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.