

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

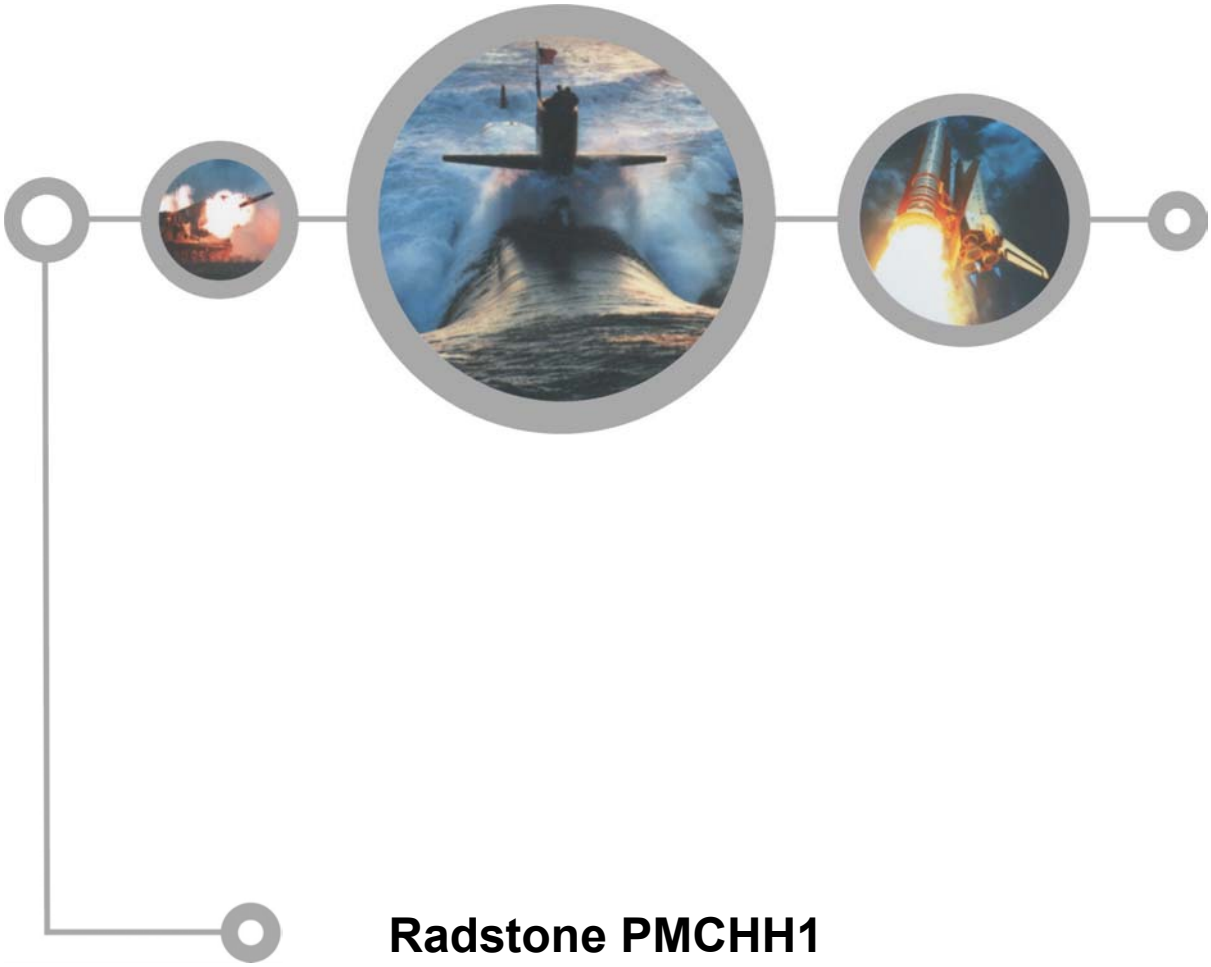
Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*



Radstone PMCHH1 Manual

Publication No. PMCHH1-0HH
Rev A1, June 2002
© Radstone Technology PLC 2002

RADSTONE
TECHNOLOGY
Why settle for less

Copyright Notice

© Radstone Technology PLC 2002. All rights reserved.

This publication is issued to provide outline information only which (unless agreed by the Company in writing) may not be used, applied or reproduced for any purpose or form part of any order or contract or be regarded as a representation relating to products or services concerned. The Company reserves the right to alter without notice the specification, design, price or conditions of supply of any product or service.

Trademarks

Radstone and the Radstone logo are trademarks of Radstone Technology PLC.

All other company and product names are acknowledged as being the trademarks or registered trademarks of their respective companies.

Document History

Note: The revision state refers *only* to the manual, ***not*** the PMC.

Revision	Date
Rev -	April 2001
Rev A	April 2001
Rev A1	June 2002

Contents

Chapter 1 - Introduction

Documentation Objectives 1-2
Documentation Audience 1-2
Documentation Scope....1-2
Documentation Structure 1-2
Warnings 1-3
Documentation Conventions....1-3
Related Documents 1-4
World Wide Web Sites....1-4

Chapter 2 - Product Overview

Introduction....2-1
Features....2-1
Functional Overview....2-3
- Host-SHARC Gateway....2-3
- SHARC-Host Gateway....2-3
- Processor....2-3
- Board Architecture 2-3
- PCI Bridge Controller....2-3
- System/User FLASH Memory 2-3
- Input/Output....2-3

Operating System Support....2-4
- Virtuoso 2-4

Build Styles 2-5
- Mechanical Overview....2-6

Compatibility 2-6

Chapter 3 - Inspection

Board Identification 3-1
Inspection....3-2

Chapter 4 - Connectors

PCI Bus Interface Connector Pinouts....4-2
- P11....4-2
- P12....4-3
- PMC Signal Descriptions....4-4

FP1A Link Port Connector Pinout....4-5
- FP1A Signal Descriptions....4-6

J1 JTAG/Debug Emulator Connector Pinout....4-7
P3 JTAG Connector Pinout....4-8

Chapter 5 - Configuration, Installation and Power-up

Configuration....5-1
Installation....5-2
Power-up....5-3
- Initialisation and Self Test....5-3

Operational Description 5-3

Chapter 6 - Board Support Firmware

Minimum Processing Environment....6-1
FLASH....6-1
BSF Code Local RAM....6-1
BSF Data Local RAM....6-1
MPE Local RAM....6-1
SHARC Instructions....6-2
BIT....6-3
Configuring the Host....6-3
Running BIT....6-4
Test Descriptions and Fail Codes....6-4
Local RAM Test....6-4
SRAM Test....6-5
FLASH Test....6-5
PLX 9054 Test....6-5
SHARC Timer Test....6-5
Link DMA Test....6-5
External Port Test....6-5
Serial Port Test....6-5
Interrupt Test....6-6
Link Connection Test....6-6
Serial Connection Test....6-6
PCI Test....6-6
Board Support Firmware Command Interface....6-7
Configuration Data Memory Usage....6-9
BIT Control Block....6-12
PCI Bus Access....6-13
FLASH Usage....6-15
Bridge Chip Configuration....6-16

Chapter 7 - Functional Description

SHARC Processor....7-1
Board Architecture....7-2
Resetting the PMCHH1....7-2
No Boot Mode....7-2
Host Boot Mode....7-2
Memory....7-3
Maps....7-3
FLASH....7-4
PLX PCI 9054....7-5
PCI Bus....7-5
PCI Access to Internal RAM....7-6
Peripheral Component Interconnect....7-6
JTAG....7-6
Software Programmable LEDs....7-7
Control Register....7-8
Interrupts....7-9
PCI Configuration....7-9

Chapter 8 - Technical Assistance

Radstone Contact Information....8-1
Helpful Hints....8-2

Appendix A - Specifications

General Specifications A-1
Local Resources A-2
EMC Regulatory Compliance and Safety A-2
Electrical Specification A-3
Reliability A-3
Environmental Specifications A-4
Mechanical Specification A-5
    Weight A-5
    Dimensions A-5
Ordering Information A-6

Appendix B - Example Code

Erase FLASH B-2
Program FLASH B-3
Check FLASH Status B-5
Initialise PLX B-6
PLX DMA Programming B-7

Appendix C - BIT Fail Data

Local Memory Test C-1
SRAM Test C-2
FLASH Checksum Test C-2
PLX 9054 Test C-2
SHARC Timer Test C-2
Link DMA Test C-2
External Port Test C-3
Serial Port Test C-3
Interrupt Test C-4
Link Connection Test C-5
Serial Connection Test C-6
PCI Test C-7

Glossary

Index

Chapter 1 - Introduction

Radstone's PMCHH1 is a SHARC ADSP-21160-based PMC that acts as a PCI to SHARC link port bridge. The PMCHH1 uses an Analog Devices ADSP-21160 operating at 80 MHz to give 480 MFLOPs peak DSP performance.

The PMCHH1 can be situated on any host card that has a fully IEEE P1386.1/VITA 20-199x PMC site. It provides six 80 Mbytes/second, 8-bit wide, bi-directional SHARC link ports at the front panel for data transfers and a 32-bit/33 MHz PCI bridge for the SHARC-to-PMC interface. The PMCHH1 also provides 8 Mbytes of FLASH, which can be configured as boot or user FLASH.

The PMCHH1 provides full software support, including Radstone's Board Support Firmware for board configuration and initialisation, Built-In-Test software, VxWorks driver support, LynxOS and Analog Devices development tools (VisualDSP and White Mountain Emulator). There is also third party support from Eonic Systems Virtuoso real time operating system.

Like all Radstone's PMC products, the PMCHH1 is available in a range of build styles, from Level 1 (Standard), suitable for a benign office-like environment, right through to Level 5 (Rugged Conduction-cooled), capable of withstanding the harshest of environments. All products are COTS/NDI, and make maximum use of low cost plastic packaged integrated circuits to ensure the most cost effective solution, whatever the market.

The air-cooled PMCHH1 is fully compliant with the IEEE P1386.1 draft 2 PMC standard, and the conduction-cooled PMCHH1 conforms to the VITA 20-199X draft 1.6 PMC standard. Its format uses only one PMC slot on the PMC host.

Documentation Objectives

This manual provides the user with sufficient information to understand the basic operation of the PMCHH1 hardware and the on-board firmware. Software support (e.g. drivers) is described in separate manuals.

Documentation Audience

This manual is written to cover, as far as possible, the range of people who will handle or use the PMCHH1, from inspectors, through system managers and installation technicians to hardware and software engineers. Most chapters assume a certain amount of knowledge on the subjects of PMC architecture, interfaces, peripherals, grounding and PCI. There is a glossary provided at the back of this manual that explains some of the terms used and expands abbreviations.

Documentation Scope

This manual describes all build standards and variants of the PMC. Applications software and software support are described in separate manuals. The host processors are also described in separate manuals.

Documentation Structure

This manual is structured in a way that will reflect the sequence of operations from receipt of the PMCHH1 up to getting it working in your system. Each topic is covered in a separate chapter and each chapter begins with a brief introduction that tells you what the chapter contains. In this way, you can skip any chapters that are not applicable or with which you are already familiar.

The chapters are:

Chapter 1 (this chapter) - gives a brief introduction, this manual's objectives, audience, scope and structure, some warnings, conventions and related documentation.

Chapter 2 - is a product overview.

Chapter 3 - contains inspection instructions.

Chapter 4 - describes the PMCHH1's connectors and signals used.

Chapter 5 - describes configuration of the PMCHH1, fitting it to a host board and subsequent power-up.

Chapter 6 - describes the Board Support Firmware.

Chapter 7 - is a functional description.

Chapter 8 - gives guidelines on obtaining technical assistance.

Appendix A - is a board specification.

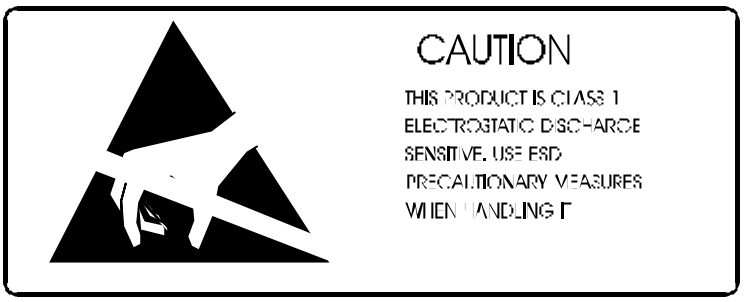
Appendix B - contains example code.

Appendix C - lists the BIT fail data.

There are also a glossary and an index provided.

Warnings

Do not exceed the maximum rated input voltages or apply reversed bias to the assembly.
If such conditions occur, toxic fumes may be produced due to the destruction of components.



Documentation Conventions

All numbers are expressed in decimal, except addresses and memory or register data, which are expressed in hexadecimal. Where confusion may occur, decimal numbers have a 'D' subscript and binary numbers have a 'b' subscript. The prefix '0x' shows a hexadecimal number, following the 'C' programming language convention. Thus:

One dozen = 12_D = 0x0C = 1100_b

'**Note:**' highlights information of particular importance.

The multipliers 'k', 'M' and 'G' have their conventional scientific and engineering meanings of $*10^3$, $*10^6$ and $*10^9$ respectively. The only exception to this is in the description of the size of memory areas, when 'K', 'M' and 'G' mean $*2^{10}$, $*2^{20}$ and $*2^{30}$ respectively.

Note: When describing transfer rates, 'k' 'M' and 'G' mean $*10^3$, $*10^6$ and $*10^9$ *not* $*2^{10}$, $*2^{20}$ and $*2^{30}$.

In SHARC terminology, multiple bit fields are numbered from 0 to n, where 0 is the LSB and n is the MSB. PCI bus terminology follows the same convention, that is, bit 0 is the LSB and bit n is the MSB.

PCI signal names follow the IEEE P1386.1. Signal names ending in with a tilde (~) denote active low signals; all other signals are active high.

When showing user interaction on a terminal, messages from the PMCHH1 are shown in courier typeface, i.e. `messages`, and user input is shown in bold courier typeface, i.e. **`user input`**. The symbol "↵" is used to represent the ↵, enter, return etc. key on your keyboard.

Related Documents

Due to the complexity of some of the devices used on the PMCHH1, it is not possible to include all the detailed data on all such devices in this document. You will need to refer to the following documents for more detailed information. In particular, you will need to refer to the IC manufacturers' manuals for an in-depth description and understanding of the various registers of these devices.

PCI Local Bus Specification, Revision 2.1, June 1 1995. *PCI Special Interest Group OR 98214 USA.*

PCI System Design Guide.

Draft Standard for a Common Mezzanine Card (CMC) Family, IEEE P1386 Draft 2.0 April 1995.

Draft Standard Physical and Environmental Layers for PCI Mezzanine Cards, IEEE P1386.1 Draft 2.0 April 1995.

Draft Standard for Conduction Cooled PMC Modules.

ADSP-2106x SHARC User's Manual (Second Edition). *Analog Devices*

ADSP-21160 SHARC Technical Specification (Rev. 3.0, February 1999). *Analog Devices*

PCI 9054 Data Book. *PLX Technology*

FLASH Memory, Volume 1 1995. *Intel corporation IL 60056-7641 USA.*

Radstone BIT V3 User Manual, publication number HH681BIT3.

Radstone PMC Installation Note, publication number HN4/3-99.

World Wide Web Sites

Manufacturers of many of the devices used on the PMCHH1 maintain FTP or world-wide-web sites. Some useful sites are:

http://www.analog.com
Analog Devices ADSP-2106x data is available through this site.

http://www.plxtech.com
PLX PCI devices data is available through this site.

Radstone on the world-wide-web is available at:

http://www.radstone.co.uk

or

http://www.radstone.com

Chapter 2 - Product Overview

This chapter contains an overview of the PMCHH1 product. Chapter 7 describes the board in more detail.

Introduction

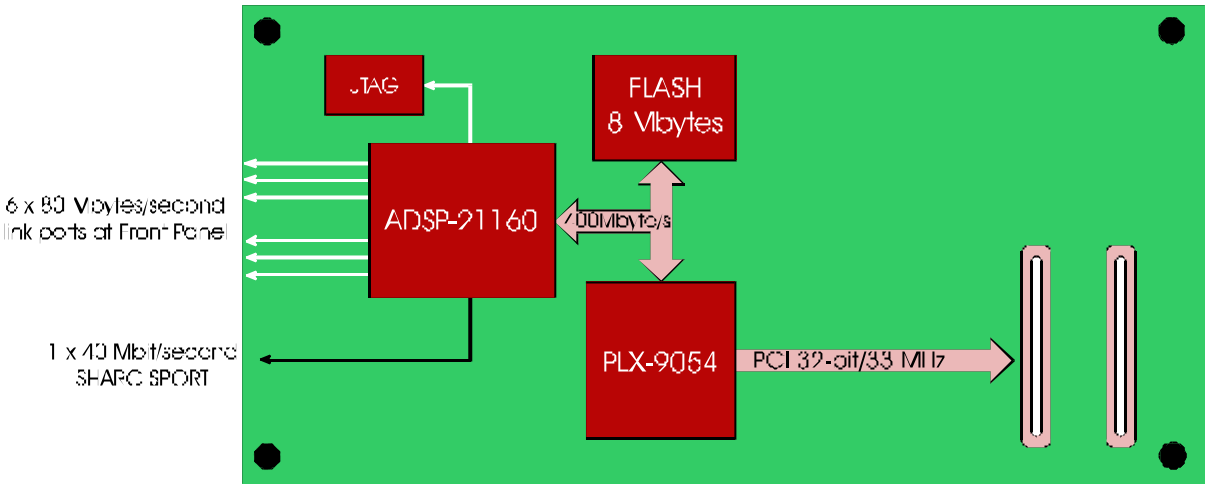
The PMCHH1 is a PMC using the Analog Devices ADSP-21160 SHARC processor. It acts as a gateway between PCI and SHARC link ports and provides an efficient means of I/O between a network of Analog Devices SHARC DSPs and sensors connected to the PCI bus.

The PMCHH1 is compliant with the IEEE P1386.1 PMC open standard (for air-cooled versions) and the VITA 20-199X PMC draft standard (for conduction-cooled versions). The PMCHH1 can be used with any motherboard that supports PMC.

Features

- ADSP-21160 SHARC DSP running at 80 MHz with 4 Mbit internal memory - 480 MFLOPS
- 8 Mbytes of FLASH for boot code and/or application code
- 6 SHARC external link port connections via front panel connector
- 1 SHARC serial port via the front panel connector
- PMC-SHARC interface using PLX PCI 9054 interface chip
- 5 ruggedization levels - air- and conduction-cooled
- JTAG connections for Emulator/debugger available via J1 connector
- Two front panel mounted LEDs for diagnostics
- Board Support Firmware (BSF) and Built-In-Test (BIT)
- IEEE P1386 Draft 2.0 and VITA 20-199x Draft 1.6 draft standard compliant
- Operating system support for Virtuoso
- 32-bit/33 MHz PCI interface via the PMC connector

Figure 2-1. Block Diagram



Functional Overview

Host-SHARC Gateway

The PMCHH1 can be used as a gateway to the network of SHARC DSPs on Radstone's VANTEGRA-1 and VANTEGRA-2 DSP boards from any host card with a PCI PMC site. The PCI data from the host card interfaces with the ADSP-21160 on the PMCHH1 via the 32-bit/33 MHz PCI to SHARC bridge (PLX 9054). The data can then be pre-processed (if desired) by the ADSP-21160 and transferred into the VANTEGRA-2 via the PMCHH1's six 80 Mbytes/second link ports at the front panel. The ADSP-21160 link ports on the PMCHH1 can also be configured as 40 Mbytes/second, 4-bit wide ADSP-2106x link ports to interface with the ADSP-21060 SHARCs on the VANTEGRA-1 series.

SHARC-Host Gateway

The PMCHH1 can also be used as a gateway from a VANTEGRA-2 to any VME host card with a PCI PMC site. The SHARC data is transferred to the PMCHH1 via the six 80 Mbyte/second link ports at the front panel, where the ADSP-21160 on the PMCHH1 can then pre-process the data (if desired). The data is then transferred to the host VME board via the SHARC-to-PCI bridge. The ADSP-21160 link ports on the PMCHH1 can also be configured as 40 Mbytes/second, 4-bit wide ADSP-2106x link ports to interface from an ADSP-2106x system such as a VANTEGRA-1.

Processor

The ADSP-211610 SHARC is a high-performance 32-bit, floating-point digital signal processor from Analog Devices. It incorporates two core processors, on-chip, dual-ported SRAM and integrated I/O peripherals supported by a dedicated I/O bus to create a complete system-on-a-chip.

Board Architecture

The SHARC has six 8-bit wide, bi-directional link ports for I/O. These are connected to the front panel connector.

The SHARC has two bi-directional serial ports. One serial port is connected to the front panel connector.

The external bus is connected to a PLX PCI 9054 to provide a PMC interface.

PCI Bridge Controller

The PLX PCI 9054 controls the PMC interface.

System/User FLASH Memory

The PMCHH1 has 8 Mbytes of FLASH memory for user and system use. The system area holds the Board Support Firmware and configuration data. The FLASH is arranged in 64-bit words and the processor has one 1 Mword bank of FLASH.

Input/Output

The PMCHH1 has 6 SHARC link ports and one SHARC serial port connected to the front panel connector.

Operating System Support

The PMCHH1 supports the Virtuoso real-time operating system.

Virtuoso

Eonic System's Virtuoso is a real-time multi-tasking operating system designed for use in DSP environments. It is based on a small, fast kernel (the "nanokernel") that allows highly efficient applications with excellent interrupt response to be written. A higher level kernel (the "microkernel") works with the nanokernel to produce a fully pre-emptive multi-tasking API for application programs.

Virtuoso comes with a Windows NT/Windows 98 development environment and is compatible with Analog Devices VisualDSP SHARC development tools.

Build Styles

The PMCHH1 is available in five electrically compatible build levels. Each level is carefully tailored to a particular set of requirements and environments. All five levels fully support the power and versatility of PMC, so no matter how large or diversified your project, absolute compatibility is assured at all stages of development. The five build levels have two basic mechanical configurations: convection-cooled in accordance with IEEE P1386, and conduction-cooled in accordance with the VITA 20-199X PMC draft standard.

Convection-cooled boards include:

Level 1
Level 2
Level 3

Conduction-cooled boards include:

Level 4
Level 5

A brief description of each build style follows. See Appendix A for more details of the build styles.

Level 1

Intended for use in benign environments, the level 1 board also provides the ideal cost effective method of complete system development. The level 1 assembly comprises an IEEE P1386 size assembly with high quality commercial grade (plastic encapsulated) components. As software compatibility throughout the build styles is absolute, a system intended for final implementation in a severe tactical environment can be developed and debugged at low cost, switching over to the target style only in the final stages of system integration.

Level 2

As Level 1, but conformally coated and tested in manufacture to provide an extended operating range.

Level 3

Level 3 boards are intended for applications that have extended temperature, shock and vibration requirements, but can be served by conventional, forced-air cooled, racking systems. These rugged boards comprise an IEEE P1386 size assembly fitted with wide temperature range, industrial grade components and are conformally coated as standard.

Level 4

Designed primarily for use in sealed ATR chassis and other conduction-cooled environments the Level 4 board features wide temperature range, industrial grade devices, an integral thermal management layer and incorporates a central stiffening bar for additional strength. Cooling is achieved through conduction of heat from the thermal management layer to the cold wall of the rack to which the boards are secured by screw driven wedgelocks. Level 4 boards are temperature-characterised and conformally coated during manufacture.

Level 5

As Level 4, but tested in manufacture to provide an extended operating and storage temperature range.

Mechanical Overview

The PMCHH1 card uses double sided component mounting. Surface mount devices are used to allow the card to keep within the specified profile.

Compatibility

The PMCHH1 conforms fully to open standards (VME and PCI/PMC) and is fully compatible with Radstone's PPCx family of PowerPC-based and VANTEGRA family of SHARC-based processor cards.

Chapter 3 - Inspection

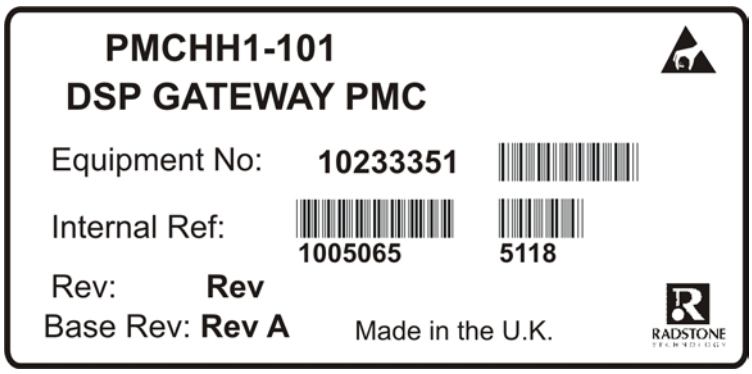
This chapter gives guidelines on inspecting the PMCHH1.

The PMCHH1 is subject to damage by static electricity, so open in a static free environment and observe antistatic precautions (wear an antistatic wrist-strap connected to an Earth point and work at an approved antistatic work station) when handling the card.

Board Identification

Radstone hardware products are identified by labels at strategic positions. You can cross-check these against the Advice Note (provided as a separate note with your delivery).

On the outside of the shipping box, and the antistatic bag, there is a label of the following form:



On the card within the antistatic bag, there is an identifying label attached to the printed wiring board. This gives a combination of details, that is the board's order number, equipment number, and revision. An example follows:

MADE IN UK	PMCHH1-101	Base Rev
10233351	Rev	Rev A

Inspection

1) Visually inspect the board for any damage and loose or dislodged components.
2) Report any defects you detect to Radstone.
3) Using the table below and Figures 3-1 and 3-2, visually inspect the board to ensure that the configuration is correct and that there are no loose or missing jumpers:

Link	Setting	Action
E1	In	Boot mode select

The following pictures show the component layout of the PMCHH1:

Figure 3-1. PMCHH1 Inward Facing Side

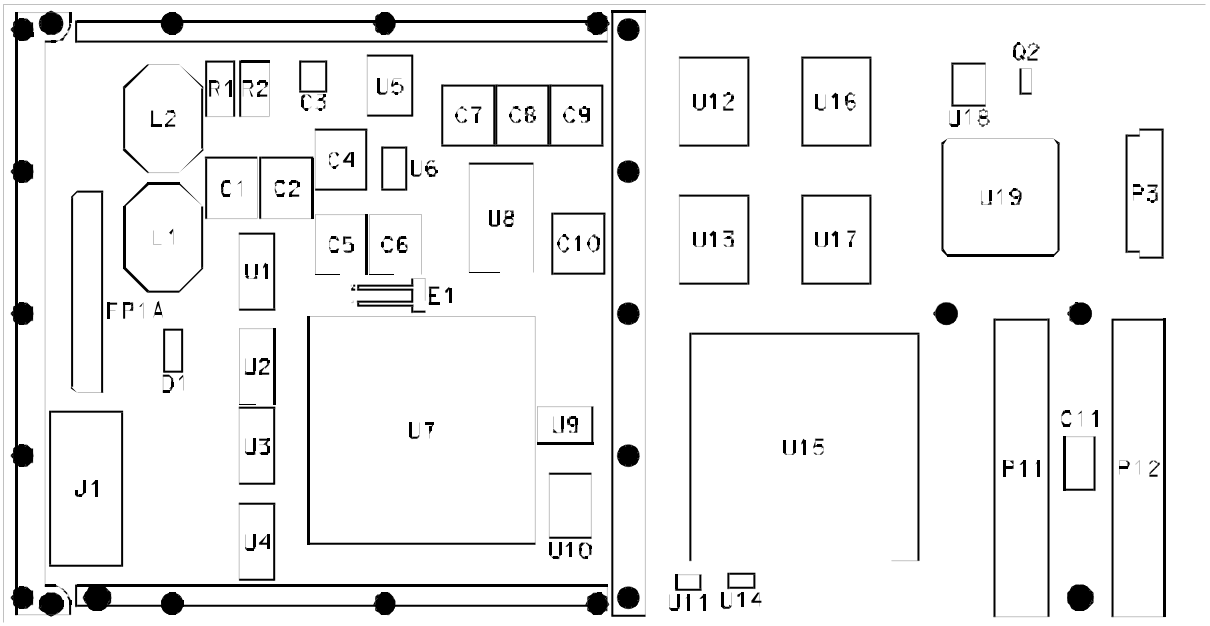
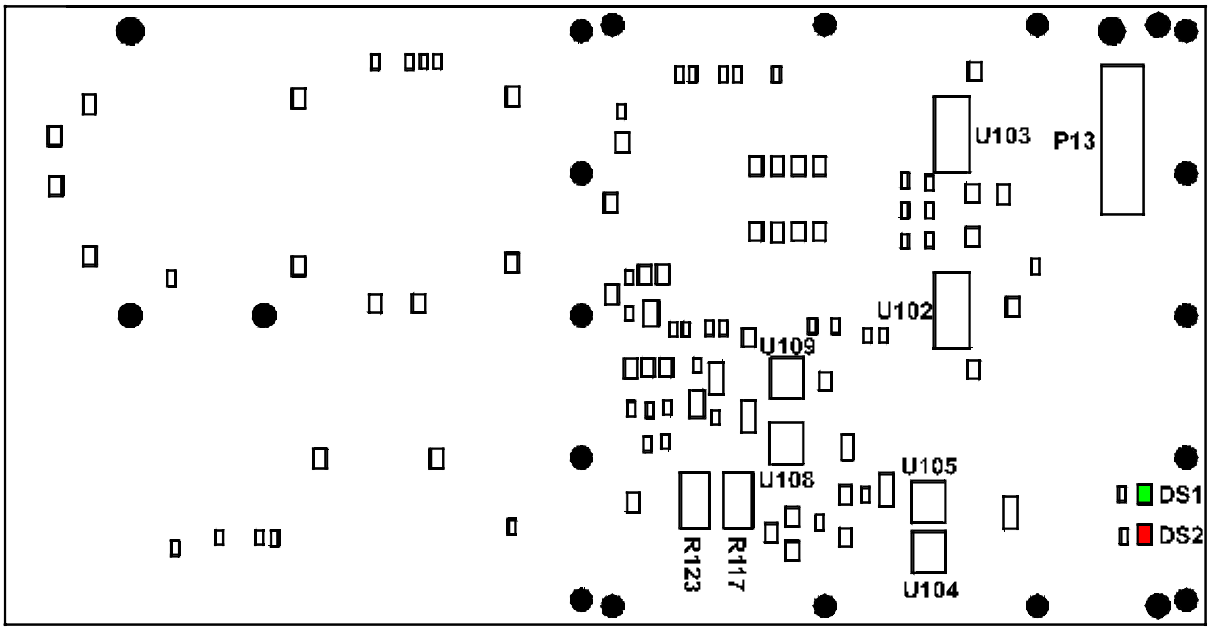


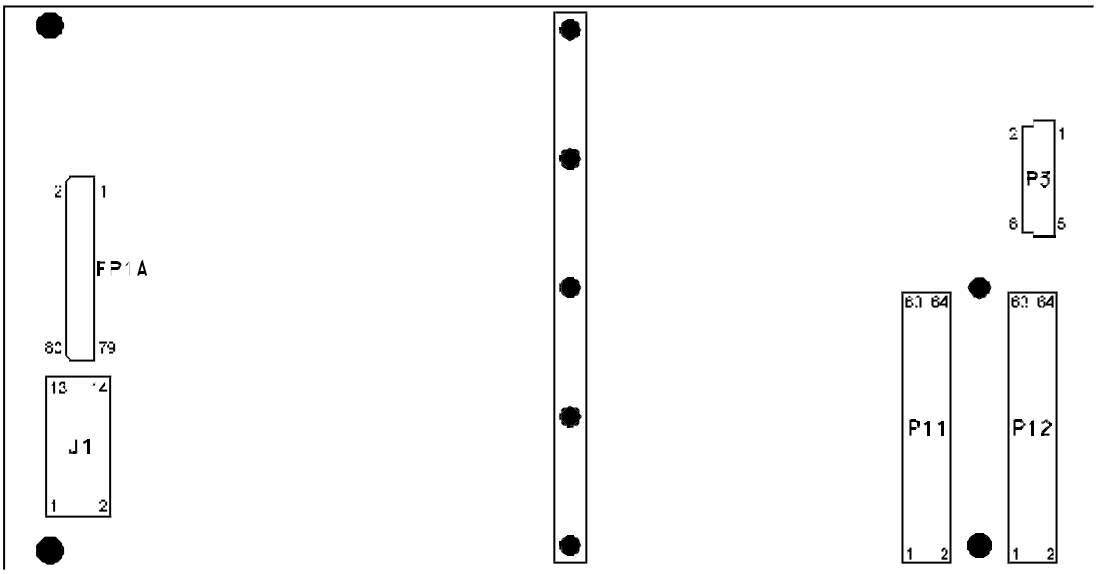
Figure 3-2. PMCHH1 Outward Facing Side



Chapter 4 - Connectors

This chapter gives the pinouts and signal descriptions for the connectors on the PMCHH1. The PMHH1 connects directly to the host through two connectors designated P11 and P12. These contain signals for the 32-bit PCI bus.

Figure 4-1. Connector Positions and Numbering



PCI Bus Interface Connector Pinouts

P11

Pin	Signal	Pin	Signal
1	NC	2	-12V
3	GND	4	INTA~
5	NC	6	NC
7	BUSMODE1~	8	+5V
9	NC	10	NC
11	GND	12	NC
13	CLK	14	GND
15	GND	16	GNT~
17	REQ~	18	+5V
19	+5V	20	AD31
21	AD28	22	AD27
23	AD25	24	GND
25	GND	26	C/BE3~
27	AD22	28	AD21
29	AD19	30	+5V
31	+5V	32	AD17
33	FRAME~	34	GND
35	GND	36	IRDY~
37	DEVSEL~	38	+5V
39	GND	40	LOCK~
41	NC	42	NC
43	PAR	44	GND
45	+5V	46	AD15
47	AD12	48	AD11
49	AD09	50	+5V
51	GND	52	C/BE0~
53	AD06	54	AD05
55	AD04	56	GND
57	+5V	58	AD03
59	AD02	60	AD01
61	AD00	62	+5V
63	GND	64	REQ64~

P12

Pin	Signal	Pin	Signal
1	+12V	2	NC
3	NC	4	NC
5	NC	6	GND
7	GND	8	NC
9	NC	10	NC
11	BUSMODE2~	12	3.3V
13	RST~	14	BUSMODE3~
15	3.3V	16	BUSMODE~4
17	NC	18	GND
19	AD30	20	AD29
21	GND	22	AD26
23	AD24	24	3.3V
25	IDSEL	26	AD23
27	3.3V	28	AD20
29	AD18	30	GND
31	AD16	32	C/BE2~
33	GND	34	NC
35	TRDY~	36	3.3V
37	GND	38	STOP~
39	PERR~	40	GND
41	3.3V	42	SERR~
43	C/BE1~	44	GND
45	AD14	46	AD13
47	GND	48	AD10
49	AD08	50	3.3V
51	AD07	52	NC
53	3.3V	54	NC
55	NC	56	GND
57	NC	58	NC
59	GND	60	NC
61	ACK64~	62	3.3V
63	GND	64	NC

PMC Signal Descriptions

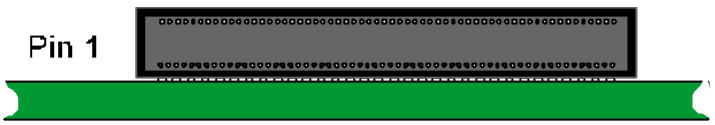
Mnemonic	Description
AD0 to AD31	**Address/Data bits.** Multiplexed address and data bus
C_BE0 to C_BE3	**Command/Byte Enables.** During the address phase, these signals specify the type of cycle to carry out on the PCI bus. During the data phase the signals are byte enables that specify the active bytes on the bus
FRAME~	**FRAME.** Driven low by the current master to signal the start and duration of an access
DEVSEL~	**Device Select.** Driven low by a PCI agent to signal that it has decoded its address as the target of the current access
PAR	**Parity.** Parity protection bit for AD0 to AD31 and BE0 to BE3
IRDY~	**Initiator Ready.** Driven low by the initiator to signal its ability to complete the current data phase
LOCK~	**LOCK.** Driven low to indicate an atomic operation that may require multiple transactions to complete
BUSMODE1~	**Bus Mode 1.** Driven low by a PMC if it supports the current bus mode
BUSMODE2~, BUSMODE3~ and BUSMODE4~	**Bus mode.** Driven by the host to indicate the bus mode. On the PMCHH1 this is always PCI
RST~	**Reset.** Driven low to reset the PCI bus
TRDY~	**Target Ready.** Driven low by the current target to signal its ability to complete the current data phase
PERR~	**Parity Error.** Driven low by a PCI agent to signal a parity error
SERR~	**System Error.** Driven low by a PCI agent to signal a system error
STOP~	**STOP.** Driven low by a PCI target to signal a disconnect or target-abort
INTA~	**Interrupt line.** Level-sensitive, active-low interrupt request
CLK	**Clock.** All PCI bus signals except RST~ are synchronous to this 33 MHz clock
REQ~	**Request.** Driven low by a PCI agent to request ownership of the PCI bus
GNT~	**Grant.** Driven low by the arbiter to grant PCI bus ownership to a PCI agent
IDSEL	**Initialisation Device Select.** Device chip select during configuration cycles
NC	**No connection**

FP1A Link Port Connector Pinout

Pin Number	Signal	Pin Number	Signal
1	Link 0 DAT(0)	2	Link 0 DAT(1)
3	Link 0 DAT(2)	4	Link 0 DAT(3)
5	GND	6	Link 0 CLK
7	Link 0 ACK	8	GND
9	Link 0 DAT(4)	10	Link 0 DAT(5)
11	Link 0 DAT(6)	12	Link 0 DAT(7)
13	Link 1 DAT(0)	14	Link 1 DAT(1)
15	Link 1 DAT(2)	16	Link 1 DAT(3)
17	GND	18	Link 1 CLK
19	Link 1 ACK	20	GND
21	Link 1 DAT(4)	22	Link 1 DAT(5)
23	Link 1 DAT(6)	24	Link 1 DAT(7)
25	Link 2 DAT(0)	26	Link 2 DAT(1)
27	Link 2 DAT(2)	28	Link 2 DAT(3)
29	GND	30	Link 2 CLK
31	Link 2 ACK	32	GND
33	Link 2 DAT(4)	34	Link 2 DAT(5)
35	Link 2 DAT(6)	36	Link 2 DAT(7)
37	Link 3 DAT(0)	38	Link 3 DAT(1)
39	Link 3 DAT(2)	40	Link 3 DAT(3)
41	GND	42	Link 3 CLK
43	Link 3 ACK	44	GND
45	Link 3 DAT(4)	46	Link 3 DAT(5)
47	Link 3 DAT(6)	48	Link 3 DAT(7)
49	Link 4 DAT(0)	50	Link 4 DAT(1)
51	Link 4 DAT(2)	52	Link 4 DAT(3)
53	GND	54	Link 4 CLK
55	Link 4 ACK	56	GND
57	Link 4 DAT(4)	58	Link 4 DAT(5)
59	Link 4 DAT(6)	60	Link 4 DAT(7)
61	Link 5 DAT(0)	62	Link 5 DAT(1)
63	Link 5 DAT(2)	64	Link 5 DAT(3)
65	GND	66	Link 5 CLK
67	Link 5 ACK	68	GND
69	Link 5 DAT(4)	70	Link 5 DAT(5)
71	Link 5 DAT(6)	72	Link 5 DAT(7)
73	TX CLK	74	RX CLK
75	TX DATA	76	RX DATA
77	GND	78	GND
79	TX FRAME SYNC	80	RX FRAME SYNC

FP1A Signal Descriptions

Mnemonic	Decription
Link n DAT(0-7)	Link Port n Link Data Bits
Link n CLK	Link Port n Link Clock
Link n ACK	Link n Link Acknowledge
TX CLK	Serial Port 0 Transmit Clock
TX DATA	Serial Port 0 Transmit Data
TX FRAME SYNC	Serial Port 0 Transmit Frame Sync
RX CLK	Serial Port 0 Receive Clock
RX DATA	Serial Port 0 Receive Data
RX FRAME SYNC	Serial Port 0 Receive Frame Sync
GND	Ground



J1 JTAG/Debug Emulator Connector Pinout

J1 combines a JTAG/Emulator connector allowing the connection of software debugging tools that use the processor's JTAG port to control the operation of the processors. J1 has the following pinout:

Pin	Signal	Pin	Signal
1	GND	2	EMU~
3	KEY	4	CLKIN (optional)
5	No Connection	6	TMS
7	+3.3V	8	TCK
9	GND	10	TRST~
11	No Connection	12	TDI
13	GND	14	TDO

Emulator Section Signal Descriptions

Signal	Pin	Description
GND	1, 9, 13	Signal ground
EMU~	2	JTAG Emulator
KEY	3	Keying pin
CLKIN	4	Optional clock input
TMS	6	JTAG Test Mode Select
+3.3V	7	+3.3V supply voltage
TCK	8	JTAG Test Clock
TRST~	10	JTAG Test Reset
TDI	12	JTAG Test Data In
TDO	14	JTAG Test Data Out

P3 JTAG Connector Pinout

P3 is a JTAG connector allowing the programming of the XILINX CPLD. P3 has the following pinout:

Pin	Signal	Pin	Signal
1	VCC	2	CPLD TCLK
3	CPLD TDI	4	CPLD TDO
5	CPLD TMS	6	GND

JTAG Signal Descriptions

Signal	Pin	Description
VCC	1	5 Volts Supply
CPLD TCLK	2	CPLD JTAG Test Clock
CPLD TDI	3	CPLD JTAG Test Data In
CPLD TDO	4	CPLD JTAG Test Data Out
CPLD TMS	5	CPLD JTAG Test Mode Select
GND	6	Signal Ground

Chapter 5 - Configuration, Installation and Power-up

This chapter describes configuration of the links on the PMCHH1, connection of the PMCHH1 to an air-cooled or conduction-cooled host and subsequent power-up.

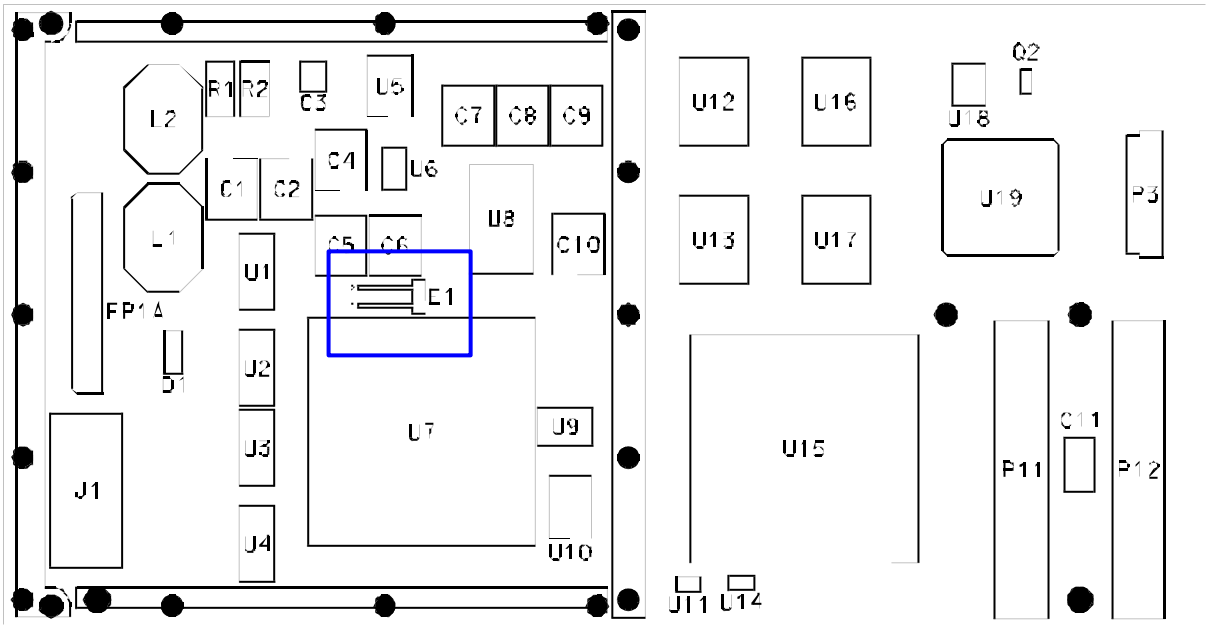
Configuration

The PMCHH1 is delivered with push-on jumper links, but for rugged or Military applications, you are recommended to connect the posts using wire wraps.

There is only one link on the PMCHH1, the Boot link (E1). This controls the boot mode of the processor. The options are no boot mode (the SHARC starts executing code from FLASH) or host boot mode (the SHARC does not execute code but waits for code to be loaded by the host). These are selected as follows:

Fitting	Meaning
In (default)	No boot mode
Out	Host boot mode

Figure 6-1. Link Position

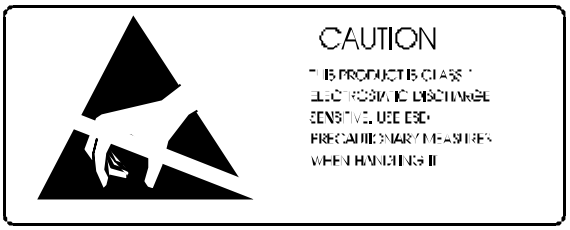


The default configuration is the likely end system configuration. In this configuration, the PMCHH1 runs the Board Support Firmware from FLASH, carries out a self-test and then runs the user's application code.

When using an Emulator, link E1 should be removed (not fitted). In this configuration, the emulator loads code to the internal memory of the processor and starts execution of the code.

Installation

This section describes connection of the PMCHH1 to an air-cooled or conduction-cooled host.



The PMCHH1 conforms to the size/power requirements of a single PMC. The card is designed to fit to any host card that is keyed to accept 5V or 3.3V PMC modules.

The PMCHH1 is supplied with a full kit of parts for mounting it, full fitting instructions and a user guide. The installation of driver software or other firmware configuration may be required to achieve full functionality of a PMC module. PMC modules ordered with a Radstone host card can be supplied factory-fitted. Consult your Sales person for information.

To achieve compliance with the Rugged CMC specification, additional metalwork is required. This only affects level 3 product, where a half-height centre stiffening bar is fitted at Radstone. This bar interfaces between the PMCHH1 and the metalwork on a Radstone host card. For level 1 and 2 PMCHH1s, the use of a half-height centre stiffening bar is not required. For levels 4 and 5, the host card metalwork should be designed to accept rugged PMCs without the need for additional metalwork.

The PMC Installation application note, HN4/2-99, describes the fitting of any level to PMC to any level of host, including non-Radstone hosts.

Power-up

Having attached the PMCHH1 to the host, as described previously, you can proceed with installing and powering it up as described in the relevant host's Manual.

Initialisation and Self Test

On power-up, as explained in the relevant host's Manual, the host runs through a self-test diagnostic package called BIT that initialises all the on-board resources into their default conditions and interrogates them. PMCHH1's inclusion in these tests are described in Chapter 6.

The first part of BIT is called the Minimum Processing Environment (described in Chapter 6). If MPE finds an error, the red LED (DS2) is lit and the PMC halts. If MPE completes successfully, the green LED (DS1) is lit and the PMC is initialised for running the rest of BIT.

After MPE, the PMCHH1 runs through a more comprehensive self-test (described in Chapter 6). While the tests are running, both LEDs should be lit. When the tests are complete, only the green LED should be lit, signifying that the tests have passed. If the red LED remains lit, this signifies that one or more of the tests has failed. The initial self-test routine should complete within 3 seconds.

Operational Description

Once the initial self-tests have been completed, the Board Support Firmware (BSF) enters the BIT environment with the PMCHH1 set up as a BIT Slave. In this mode, the system BIT master can control the board via the BIT Control Block (BCB). The Radstone BIT V3 User Manual describes this interface to the PMCHH1. Chapter 6 details the location in memory of the BCB and describes the BIT tests.

The BSF also provides a command interface to the PMCHH1, which allows the user to:

- Configure the PMCHH1
- Programme applications into PMCHH1 FLASH memory
- Run BIT
- Execute application code

The command interface, which is described in Chapter 6, can be entered from the BIT slave mode by setting bit 11 of the BIT enable word.

Chapter 6 describes the BSF.

Chapter 6 - Board Support Firmware

This chapter describes the functionality provided by the Board Support Firmware, which controls the PMCHH1's operation after power-up.

The PMCHH1 is supplied with BSF programmed into FLASH memory. When the board is configured in no boot mode, this software runs automatically.

Minimum Processing Environment

MPE is always the first module of the BSF to be executed when the PMCHH1 is powered on. It gives the user confidence that the hardware has a basic functionality that enables the execution of power-up BIT. Any failure during MPE causes the PMCHH1 to halt and the red LED (DS2) to be turned on.

MPE consists of a series of low-level tests. If a failure occurs, the fail code is written to the MPE fail flag in the BCB (see the Radstone BIT V3 User Manual). If all the MPE tests complete successfully, the green LED (DS1) is turned on and Initial BIT is carried out (see later).

The following sections describe the MPE tests.

FLASH

The area of FLASH that contains the BSF is tested to ensure that the contents are valid. A checksum value is calculated for each sector of FLASH containing executable code. The checksum for a sector is calculated by adding together each byte in the sector to give a 32-bit total. The calculated value is then compared with the checksum stored in the checksum table in the configuration sector of the FLASH. The fail code is 0xfd02.

BSF Code Local RAM

A memory checksum test is carried out on the area of internal RAM that is used by the BSF code. The fail code is 0xfc03.

BSF Data Local RAM

A memory checksum test is carried out on the area of internal RAM that is used by the BSF to store data. The fail code is 0xfb04.

MPE Local RAM

A memory checksum test is carried out on the area of internal RAM that is used by the MPE to run further tests. The fail code is 0xfa05.

The memory checksum test involves calculating the sum of all the bytes in the memory area to form three checksums. The data is inverted and written back to memory after the first and second checksums are calculated but not after the last checksum is calculated. The test passes if the second and third checksums have the correct values when compared to the first checksum.

SHARC Instructions

All the SHARC instructions are tested. As well as the MPE fail code 0xfe01, one of the sub-test values shown in the table below is written to the BIT Fail Word in the BCB.

Fail Code	SHARC Test
0x01	Jump command
0x02	Delayed jump command
0x03	Delayed jump instruction executed
0x04	Indirect jump
0x05	Read/write to program memory
0x06	Read/write to data memory
0x07	Bit commands
0x08	Equality condition
0x09	Less than/greater than conditions
0x0A	Negative/positive conditions
0x0B	Fixed point overflow condition
0x0C	Fixed point carry condition
0x0D	Multiplier negative and overflow conditions
0x0E	Fractional integer arithmetic
0x0F	Negating and ABS commands
0x10	Logical commands
0x11	Compare command
0x12	Min/max and clip commands
0x13	Floating point arithmetic
0x14	Floating point scalb command
0x15	Floating point fix and trunc commands
0x16	Floating point logb command
0x17	Floating point recips command
0x18	Floating point rsqrts command
0x19	Lshift command
0x1A	Ashift command
0x1B	Rot command
0x1C	Bclr/bset/btgl commands
0x1D	Btst commands
0x1E	Fdep command
0x1F	Fext commands
0x20	Exp/leftz/lefto commands

At the conclusion of MPE, and assuming all the tests have passed, the BSF code and data segments are copied from FLASH to internal RAM. Execution then jumps to internal RAM and the ‘C’ environment is initialised. MPE completes by initialising the hardware, in preparation for running Initial BIT.

BIT

This section details the Built-In Test support provided by the PMCHH1 BSF.

Configuring the Host

The board hosting the PMCHH1 needs to be correctly configured to recognise the PMC if the BIT code is to run successfully. The BIT code is preloaded into the on-board FLASH device. If this becomes corrupted in any way, it can be reprogrammed from within VxWorks or LynxOS. Please contact Radstone before undertaking this procedure.

To configure the host to recognise the PMCHH1 board:

1) Press any key on the VT100 keyboard to stop the board booting into VxWorks/LynxOS.

2) Select Option 1 to enter the SDE environment.

3) PMCHH1 BIT is invoked by the host processor by including a suitable entry in the System Description. From the options menu, select System Description and add a new entry as follows. Type:

```
System Description>create 1¿
System Description>mod n ciin=PMCHH1,method=BCB_TEST,mask=0¿
System Description>mod n bus=B0S6,address=000be000,am=PCI¿
System Description>save¿
```

(where n is the number of the SDE entry created by the initial '`create`')

This results in the following output:

```
SDE  CIIN    ROUTINE   ADDRESS   BUS   AM   METHOD    LENGTH    GRAN  SKP  TESTMASK
---  ----    -------   -------   ---   --   ------    ------    ----  ---  --------
…
00n  PMCHH1         0  000be000  B0S6  PCI  BCB_TEST                NO   00000000
```

Notes:

1) This is for a PPC4 host with a single PMCHH1 fitted directly in slot 2.

2) For other hosts or where a carrier card is used, the bus and slot of the PMC need to be modified in the '`bus=`' entry. The bus and slot values for some other Radstone cards are:

Card	Slot 1	Slot 2
PPC1	B1S0	B1S1
PPC1A	B0S5	B0S6
PPC2	B0S5	Not available
PPC2A	B0S5	Not available
PPC4	B0S5	B0S6

Running BIT

Use BIT's Factory level to verify the PMCHH1 functionality. BIT is run as follows:

1) Press any key on the VT100 keyboard to stop the host booting into VxWorks.
2) Select Option 6 to enter the interactive BIT environment.
3) Type '**`mo 1f`**¿' for single shot mode (1 cycle through the tests) or '**`mo 2f`**¿' for continuous mode (continuous cycling through the tests).
4) Type '**`run`**¿' to start the tests.

Test Descriptions and Fail Codes

PMCHH1 BIT is split into a series of tests, each one concerned with a single hardware functional block. Appendix C gives the fail data for the tests.

Local RAM Test

Initial Level

16-, 32-, 48- and 64-bit non-destructive checksum tests are carried out on the internal memory that is not used by the BSF.

The first checksum is calculated by adding together the contents of each word of the memory into a rolling 32-bit total. As each location is read, the value is inverted and written back to the memory. A second checksum is calculated in the same way, except that after the values are read, they are inverted before adding them to the total. The inverted value is written back to memory. The third checksum is calculated in the same way as the first, except no data is written back to memory. A comparison is made to confirm that all three checksum values are equal. To calculate the 48-bit checksum, the most significant 32 bits and least significant 16 bits are added to the 32-bit total separately. To calculate the 64-bit checksum, the two 32-bit words are added separately.

Comprehensive and Factory Level

The same tests as at Initial level are carried out, plus 16-, 32-, 48-and 64-bit destructive memory tests on the local memory that is not used by the BSF.

The first part of the destructive memory test involves writing the address of a memory location to that location and checking that the expected value is read back. Each of the 32 possible rotations of the address is tested. For the 48-bit test, the 48-bit data written to memory consists of the 32-bit test data concatenated with the least significant 16 bits of the test data. For the 16-bit test, the least significant 16 bits of the test data is written to the memory location. For the 64-bit test, the 64-bit data written to memory consists of the 32-bit test data written to each half of the 64-bit word.

The second part of the destructive memory tests involves zeroing the memory, writing a non zero value to each memory location that is a power of 2 and checking that no other memory locations become non zero. For example, if the memory range is 0x100 to 0x1FF, the addresses written to are 0x100, 0x101, 0x102, 0x104, 0x108, 0x110, 0x120, 0x140 and 0x180.

SRAM Test

There is no SRAM on the PMCHH1. This test always returns passed.

Note: This test is included, although not applicable, because the BIT self-test run on the PMCHH1 is a subset of that run on other Radstone ADSP-21160-based products (VANTEGRA-2), which may have SRAM.

FLASH Test

For all test levels, this test calculates the checksum of a FLASH sector and compares the result with the value stored in the configuration sector. The checksum is calculated by adding together each byte of each word in the sector to give a 32-bit result. The configuration sector's checksum is calculated by clearing the configuration sector's checksum in the checksum table and calculating the checksum as for any other sector. All sectors in the FLASH are tested.

PLX 9054 Test

For all test levels, this test consists of checking the PLX ID register, followed by a rolling '1' and a rolling '0' test on all the read/write registers of the PLX chip.

SHARC Timer Test

For all test levels, this test checks that the SHARC timer decrements and that high and low priority timer interrupts are generated.

Link DMA Test

For all test levels, a link port is put into loopback mode and data transferred by DMA from one block of local memory to another using that link port. The transfers are carried out using both 32- and 48-bit data. This is then repeated using chained DMA. The associated DMA channel interrupts are also tested. The test is carried out on all link ports.

External Port Test

There is no external port test on the PMCHH1 as there is no external SRAM. This test always returns passed. See the Note for the SRAM test.

Serial Port Test

For all test levels, each serial port is put into loopback mode and data transferred by DMA from one block of local memory to another using a serial port. This is then repeated using chained DMA. The associated DMA channel interrupts are also tested. The test is carried out on both serial ports.

Interrupt Test

For all test levels, this test consists of generating SHARC interrupts. The interrupts that are tested are:

- IOP access interrupt
- Unaligned 64-bit access interrupt
- PC stack full interrupt
- Fixed and floating point interrupts
- Software interrupts
- Circular buffer interrupts
- External interrupts 1 and 2

Note: External interrupt 1 is only tested at factory level.

Link Connection Test

This test is only carried out at Factory level. For Initial and Comprehensive levels, the test always returns passed. Transfers across links and the link port service request interrupt are tested. The test uses a front panel connector to connect the links together in pairs.

Serial Connection Test

This test is only carried out at Factory level. For Initial and Comprehensive levels, the test always returns passed. The test is carried out on the serial port that is connected to the front panel connector. This port is looped back so that the transmit and receive channels are connected together. Data is sent on the transmit channel to ensure that it is received on the receive channel.

PCI Test

This test is carried out at Comprehensive and Factory levels only. For Initial level, the test always returns passed. Data is written to RAM over the PCI bus and then read back. The test is carried out using DMA transfers and processor writes and reads. The size of the data blocks written and read is 0x10000 bytes. Five fixed data patterns (0xffffffff, 0x0, 0xffffffff, 0xa5a5a5a5 and 0x5a5a5a5a) followed by five incrementing patterns are used for each test. The incrementing patterns use randomly generated start values.

Board Support Firmware Command Interface

The PMCHH1 BSF provides a command interface that allows the BSF VxWorks driver to run BIT, set up the board's configuration and to programme FLASH.

Each command consists of a command code and up to eight parameters. The parameters are passed to the PMCHH1 by writing them to the PLX PCI 9054 mailbox registers (parameter 1 in mailbox 0, parameter 2 in mailbox 1 etc.). The command code is written to the PLX PCI 9054 PCI to Local Doorbell register and this generates an interrupt that causes the command to be executed.

When the command has completed, the PMCHH1 puts the result parameters in the PLX PCI 9054 mailbox registers and the result code in the Local to PCI Doorbell register. The allowed result codes are Command OK (0x7F) and Command Failed (0x7E). The driver can determine when the command has completed by polling the Local to PCI Doorbell register, or set up the system so that the INTA interrupt line causes an interrupt when the command has completed.

The available commands are listed in the following table:

Command Code	Command Parameters	Command Results	Command Description
1	Index	Value	Get element of configuration data structure
2	Index	None	Set element of configuration data structure
	Value		
3	None	None	Save configuration data to FLASH
4	None	None	Restore configuration data from FLASH
5	None	None	Restore factory default configuration data from FLASH
6	None	None	Update FLASH checksums
7	None	None	Restart BSF code
8	Mode	None	Set new BIT mode
9	None	Mode	Get current BIT mode
10	Mask	None	Set BIT Ignore mask
11	Skip	None	Set BIT Skip status
12	None	None	Run BIT
13	None	None	Start BIT Environment
14	None	None	Start application
15	Source	None	Set up application code structure. Source is FLASH address of code section, Destination is RAM address to copy code to, Length is the number of words to copy, Data size is 32- or 48-bit data, Section is in range 0-9
	Destination		
	Length		
	Data size[1]		
	Section		
16	None	None	Clear application code structure

(continued overleaf)

Command Code	Command Parameters	Command Results	Command Description
17	None	None	Start link booting mode
18	Address	None	Erase FLASH sector starting at address
19	FLASH address	None	Programme FLASH block. FLASH address is where to start programming, Source address is start of data to programme, Length is number of words to programme, Data size is 32- or 48-bit data
	Length		
	Source address		
	Data size[1]		
20	FLASH address	None	Programme FLASH word. FLASH address is where to programme the word data, Data size is 32- or 48-bit word
	Data size[1]		
	Data[2]		
	Data[3]		
21	Data size[1]	None	Set up SHARC so that PLX PCI 9054 can write 32- or 48-bit data to internal memory
22	None	None	Turn the green LED on
23	None	None	Turn the green LED off
24	None	None	Toggle the green LED
25	None	None	Turn the red LED on
26	None	None	Turn the red LED off
27	None	None	Toggle the red LED
28	Address	Data[2]	Get word from memory. Address is location of word to read, Data size is 32-or 48-bit word
	Data size[1]	Data[3]	
29	Address	None	Set word in memory. Address is location of word to write, Data size is 32- or 48-bit word, Data is the word to write
	Data size[1]		
	Data[2]		
	Data[3]		
30	None	Rev[4]	Get software revision

Notes:

1) 0 = 32-bit data
   1 = 48-bit data

2) Data for 32-bit data, most significant 32 bits of 48-bit data.

3) Least significant 16 bits of 48-bit data.

4) Rev is an ASCII string defining the software revision, e.g. 0x00313030 means revision 1.00.

Configuration Data Memory Usage

The configuration data is copied from FLASH to local RAM, after MPE is complete, where it is stored as 32-bit words. The FLASH is 64 bits wide, so when stored in FLASH, the configuration data uses the least significant 32 bits for even addresses and the most significant 32 bits for odd addresses. The table below details the layout of the configuration data (sizes are given as the number of 32-bit words used by a particular element):

Contents	Size
Checksum value for each sector of the FLASH [1]	32
Magic Word indicating valid configuration data [1]	1
Number of FLASH Sectors [1]	1
Type of FLASH on board [1]	1
Software Revision [1]	1
Configure PLX	1
System Description	92
PLX Registers [2]	66
SHARC Registers	3
FLASH Types [1]	6
Reserved [1]	1
FLASH Write Counts [1]	32
Application Code Description	42

Notes:

1) These elements should only be read using the command interface.

2) Only the ‘value’ elements of this array should be changed using the command interface. The ‘addr’ elements should only be read.

3) The Restore Configuration command copies all the configuration data from the current FLASH version to the current RAM version. The Restore Factory Configuration command copies all the configuration data, except for the FLASH write counts and application code description sections, from the factory default FLASH version.

The structures used to store the System Description, the PLX registers, the FLASH types and the Application Code Description are listed below. All other elements of the configuration data use integers (i.e. 32-bit words). In the following, parameters of type 'char' and pointers have the same size as integers.

Structure for System Description

```
typedef struct {
      char ciin[9];           /* Type of card */
      void (*routine)();      /* Address of the test routine */
      int  offset;            /* Routine offset (Visual DSP fix) */
      char type[12];          /* Type of test to run */
      char bus[8];            /* Type of bus */
      char am[8];             /* Address data width */
      void *address;          /* Address of control block */
      int  memory_length;     /* Memory length */
      int  granularity;       /* Granularity of memory testing */
      int  test_skip;         /* Skip test flag */
      int  test_ignore_mask;  /* Test ignore mask */
      int  iterations;        /* Number of times run */
      int  failures;          /* Number of times failed */
} system_description_item_t;
```

Structure for PLX Registers

```
typedef struct {
      int  *addr;             /* Register address */
      int  value;             /* Register value */
} reg_config_t;
```

Structures for Application Code Description

```
typedef struct {
      int  block[NO_PROCS];
      code_block_t app_code[NO_PROCS];
} application_code_t;
```

For the PMCHH1, `NO_PROCS` has the value 1 and so `block[0]` will either have the value 0 or 6 (no code) indicating that the processor will run application programme 0 or no code.

```
typedef struct {
      int  programmed;
      code_section_t section[MAX_SECTORS];
} code_block_t;
```

For the PMCHH1, `MAX_SECTORS` has the value 10.

```
typedef struct {
      int  source;            /* Location in FLASH */
      int  dest;              /* Location in RAM */
      int  length;            /* Length to copy */
      int  memory;            /* Memory type 48/32 bit */
} code_section_t;
```

Structure to Store FLASH Types PMCHH1 Recognises

```
typedef struct {
      int  man_code;          /* Manufactures code */
      int  dev_code;          /* Device code */
      int  flash_type;        /* BSF code */
} flash_types_t;
```

BIT Control Block

This section describes the PMCHH1 BCB. This has the following structure:

	31 24	23 16	15 8	7 0
0	BIT Enable Word		BIT Mode	Reserved
1	BIT Ignore Mask			
2	Processor ID	Card ID	MPE Error~/MPE Error	
3	BIT Fail Word			
4	BIT Error			
5	Dual-Ported RAM Size			
6	Not Used			
7	Not Used			
8	BIT Test Index			
9	Not Used			
A	Not Used			
B	Not Used			
C	Not Used			
D	Not Used			
E	Not Used			
F	Address Pointer			

The bits in the BIT Enable Word that the PMCHH1, in BIT slave mode, responds to are:

Bit	Meaning
0	Cold-start BIT; executes as though from reset
1	WARM-start BIT; executes as required by BIT master
2	Switch off the BIT Fail LED
3	Switch on the BIT Fail LED
4	Run user-supplied code from the offset specified by Address Pointer
10	Reload the BCB
11	Enter Command interface
14	Run card as a BIT Slave with parameters supplied by the BIT Master
15	Start application code

The Card ID for the PMCHH1 is 36.

The VME address of the BCB is set up by the PMC host card.

For more information about the BCB, see the Radstone BIT V3 User Manual.

To enable the BCB, and the fail data blocks, to be accessed correctly by the PCI bus, the local start address of the BCB is 0x5F000 and the local increment between elements is 2.

PCI Bus Access

The SHARCs use the address range 0xB00000 to 0xB000FF to access the PLX PCI 9054 registers. The PCI bus is accessed in the range 0xC00000 to 0xFFFFFF. To access PCI memory space, the following PLX PCI 9054 registers need to be set up:

Register	SHARC Address	Value
Range for Direct Master to PCI	0xB00027	0xFF800000
Local Base Address for Direct Master to PCI Memory	0xB00028	0x30000000
PCI Base Address (Re-map) for Direct Master to PCI	0xB0002A	0xXXX00001

where XXX represents the 1 Mbyte boundary base address of the PCI region to be accessed.

To access PCI I/O space the following PLX PCI 9054 registers need to be set up:

Register	SHARC Address	Value
Range for Direct Master to PCI	0xB00027	0xFF800000
Local Base Address for Direct Master to PCI IO/CFG	0xB00029	0x30000000
PCI Base Address (Re-map) for Direct Master to PCI	0xB0002A	0xXXX00002

where XXX represents the 1 Mbyte boundary base address of the PCI region to be accessed.

To enable PCI access to the SHARC memory space, local address space registers must be set up. The following is an example of setting up local address space 0 to access internal RAM of the SHARC:

Register	SHARC Address	Value
Range for PCI to Local Address Space 0	0xB00020	0xFFE00000
Local Base Address (Re-map) for PCI to Local Address Space 0	0xB00021	0x00000001

The PMC host card sets up the “PCI Base Address for Local Address Space 0” register to the PCI base address to access the local memory region.

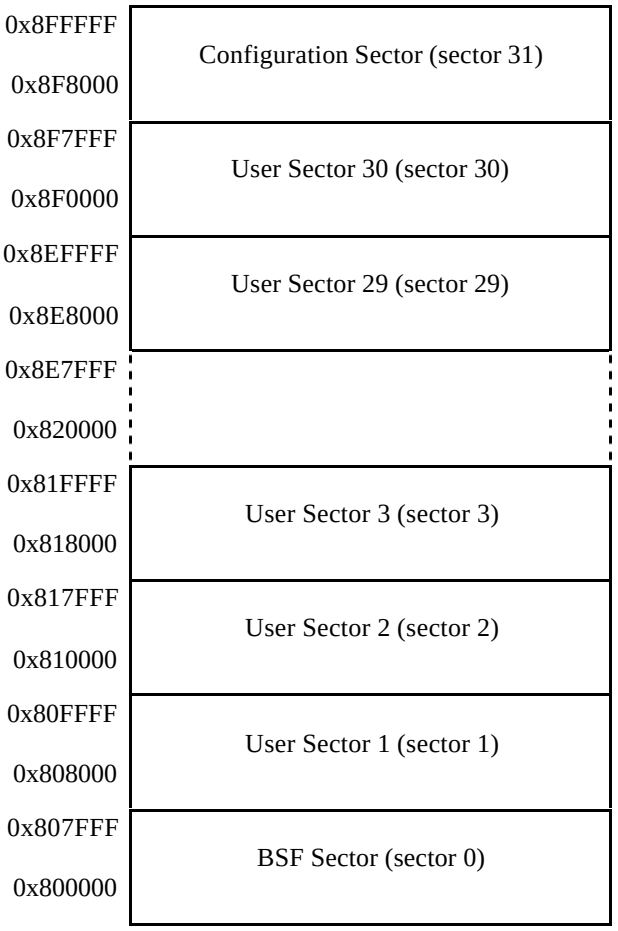
The wait states for PCI accesses to memory should be set to zero (in the “Bus Region Descriptors for PCI to Local Addresses” register, address 0xB00026).

To set up a DMA transfer using DMA channel 0 from the start of external FLASH to PCI address 0x12345678, the following registers must be set up:

Register	SHARC Address	Value
DMA Channel 0 PCI Address Register	0xB00041	0x12345678
DMA Channel 0 Local Address Register	0xB00042	0x01000000
DMA Channel 0 Transfer Size (bytes) Register	0xB00043	0x00001000
DMA Channel 0 Descriptor Pointer Register	0xB00044	0x00000008

The wait states for PCI accesses to local memory should be set to zero (in the DMA Channel 0 Mode register, address 0xB00040).

FLASH Usage


0x8FFFFF
0x8F8000
Configuration Sector (sector 31)
0x8F7FFF
0x8F0000
User Sector 30 (sector 30)
0x8EFFFF
0x8E8000
User Sector 29 (sector 29)
0x8E7FFF
0x820000
0x81FFFF
0x818000
User Sector 3 (sector 3)
0x817FFF
0x810000
User Sector 2 (sector 2)
0x80FFFF
0x808000
User Sector 1 (sector 1)
0x807FFF
0x800000
BSF Sector (sector 0)

Bridge Chip Configuration

This section lists the PLX PCI 9054 bridge chip registers that can be edited and their default values. These values are programmed to the PLX PCI 9054 when application code is started provided that the configuration data 'Configure PLX' is set to YES (its default value is YES).

Index	Register	Description	Default
0	0x80	Range for PCI to Local Address Space 0	0xFFC00000
1	0x84	Local Base Address (Re-map) for PCI to Local Address Space 0	0x00000001
2	0x88/0x12C	Mode/DMA Arbitration	0x00200000
3	0x8C	Big/Little Endian Descriptor and Local Miscellaneous Control	0x00300500
4	0x90	Range for PCI toLocal Expansion ROM	0xFFFF0000
5	0x94	Local Base Address (Re-map) for PCI to Local Expansion ROM and BREQ Control	0x00000000
6	0x98	Bus Region Descriptors for PCI to Local Addresses	0x404300C3
7	0x9C	Range for Direct Master to PCI	0xFFE00000
8	0xA0	Local Base Address for Direct Master to PCI Memory	0x01800000
9	0xA4	Local Base Address for Direct Master to PCI IO/CFG	0x01800000
10	0xA8	PCI Base Address (Re-map) for Direct Master to PCI	0x00000001
11	0xAC	PCI Configuration Address Register for Direct Master to PCI IO/CFG	0x00000000
12	0xB4	Outbound Post Queue Interrupt Mask	0x00000008
13	0xE8	Interrupt Control/Status	0x00010100
14	0xEC	Serial EEPROM Control, PCI Command Codes, User I/O, Init Control	0x000D767E
15	0x130	DMA Threshold Register	0x00000000
16	0x134	DMA 0 PCI Dual Address Cycle Address	0x00000000
17	0x138	DMA 1 PCI Dual Address Cycle Address	0x00000000
18	0x140	Messaging Queue Configuration	0x00000002
19	0x144	Queue Base Address	0x00000000
20	0x148	Inbound Free Head Pointer	0x00000000
21	0x14C	Inbound Free Tail Pointer	0x00000000
22	0x150	Inbound Post Head Pointer	0x00000000
23	0x154	Inbound Post Tail Pointer	0x00000000
24	0x158	Outbound Free Head Pointer	0x00000000
25	0x15C	Outbound Free Tail Pointer	0x00000000
26	0x160	Outbound Post Head Pointer	0x00000000
27	0x164	Outbound Post Tail Pointer	0x00000000
27	0x168	Queue Status/Control	0x00000050
29	0x170	Range for PCI-to-Local Address Space 1	0xFFF00000
30	0x174	Local Base Address(Re-map) for PCI-to-Local Address Space 1	0x00000000
31	0x178	Local Bus Region Descriptor (Space 1) for PCI-to-Local Accesses	0x00000043
32	0x17C	Direct Master PCI Dual Address Cycle	0x00000000

Chapter 7 - Functional Description

This chapter gives a detailed description of the PMCHH1's functional blocks. Chapter 2 gives a product overview.

SHARC Processor

The ADSP-21160 SHARC DSP device incorporates a dual core processor, on-chip, dual-ported SRAM and integrated I/O peripherals, supported by a dedicated I/O bus to create a complete system-on-a-chip. The SHARC extends the traditional Harvard Architecture by providing four independent internal buses (two for program and two for data) that allow two simultaneous and independent accesses to instructions and data.

The SHARC processing cores each contain three independent, floating point computational units (multiplier, ALU and Barrel Shifter), all of which carry out single cycle instructions. The I/O core contains four 80 Mbytes/second external interface ports, six 80 Mbytes/second link ports, two 80 Mbits/second serial ports and a 14 channel DMA controller.

The internal dual-ported SRAM (4 Mbits) is organised in two separate blocks and can be configured for different combinations of program and data storage. Each memory block is dual-ported for single cycle, independent accesses from the core processors and I/O processor. This allows the dual-ported SRAM to service two 32-bit data transfers from each processing core and one 64-bit transfer from the I/O core in a single cycle.

The SHARC has a 320 Mbytes/second, 64-bit wide bus that interfaces to the FLASH memory and PLX PCI 9054 PCI bridge chip.

The six, 8-bit wide, bi-directional link ports provide for I/O support. Each link port can operate independently and simultaneously at up to 80 Mbytes/second. The aggregate bandwidth of the six link ports is 480 Mbytes/second peak. Link port data can be transferred as 32-, 48- or 64-bit words and can be read directly by the core processors or DMA transferred to on-chip memory.

The four external ports can be used to transfer data from FLASH memory to internal SRAM. The two serial ports support independent transmit and receive functions.

The 14 DMA channels support zero overhead data transfers between the SHARC's internal memory and external memory, link ports and serial ports. The SHARC also provides full IEEE JTAG standard 1149.1 access and on-chip emulation.

Board Architecture

The SHARC has a 64-bit external bus that connects it to the FLASH memory and the PLX PCI 9054 bridge chip. The PLX PCI 9054 provides the PMC interface.

Six link ports and one serial port of the SHARC are connected to the front panel connector.

The SHARC can be configured in the Board Support Firmware so that it can be link booted via link port 4.

Resetting the PMCHH1

A reset may be caused by power-on or a PCI reset. All devices on the PMC are reset.

The PMCHH1 can be configured to boot up in one of two modes: “No Boot Mode” or “Host Boot Mode”.

No Boot Mode

This is the mode selected when the jumper E1 is fitted. The SHARC starts executing code from address 0x800000, which is at the start of FLASH memory.

This is the recommended when using an emulator to debug code that has been programmed into FLASH. If being used in this way, the emulator must not carry out a reset when it is started.

Host Boot Mode

This is the mode selected when the jumper E1 is not fitted. The SHARC does not start, but waits for the host (the PLX PCI 9054) to load bootstrap code. When the bootstrap code has been loaded, the SHARC starts executing from address 0x40005.

This is the recommended mode when using the emulator to load code onto the SHARC and then control execution of the code.

Memory

Maps

Seen by the Processor

Processor Address Range		PCI Address Range	Cycle Type
00000000	0001FFFF	No PCI cycle	I/O Processor Registers
00020000	0003FFFF	No PCI cycle	Internal SRAM – Long Word Addressing
00040000	0007FFFF	No PCI cycle	Internal SRAM – Normal Word Addressing
00080000	000FFFFF	No PCI cycle	Internal SRAM – Short Word Addressing
00800000	008FFFFF	No PCI cycle	FLASH Memory
00B00000	00BFFFEF	No PCI cycle	PLX PCI 9054 Registers
00C00000	00FFFFFF	PCI cycle †	PCI Address Space

† The software sets up the PCI address range targeted.

Local Memory Space as seen by the PLX PCI 9054

PLX Local Address Range		Local Memory Destination
00000000	003FFFFF	SHARC Internal Memory Space
02000000	023FFFFF	FLASH

FLASH

There is 1 Mword of FLASH on the PMCHH1. The word size is 64 bits. The SHARC processor sees the FLASH as bank 0 with the address range 0x00800000 to 0x008FFFFF. The FLASH has a write capacity of 100,000 cycles.

The access time for the memory is 95 nanoseconds, so it can be accessed in 4 wait states. The values required for FLASH by the SHARC WAIT register are:

WAIT Parameter	Value
External Bank 0 Access Mode (EB0AM)	0
External Bank 0 number of Wait States (EB0WS)	4

The access state mode means that the processor uses asynchronous accesses.

The MSIZE parameter in the SYSCON register of the SHARC must be set to 7 for FLASH.

The following table shows the bits of the 64-bit FLASH words used when the SHARC carries out various sized accesses:

SHARC FLASH Access Size	Bits of 64-bit FLASH Word Used
48-bit	63 to 16
32-bit to even addresses	31 to 0
32-bit to odd addresses	63 to 32

The FLASH is divided into 32 sectors each of 32 Kwords. Sector 0 stores the BSF, and sector 31 stores the configuration data. The other 30 sectors are available for user code.

To protect the FLASH from accidental corruption, a programming voltage, Vpp, must be applied to erase or program the FLASH. This voltage is controlled by the Vpp bit in the Control register.

PLX PCI 9054

The PLX PCI 9054 registers are located in bank 3 of external memory.

The values required for PLX PCI 9054 by the SHARC WAIT register are:

WAIT Parameter	Value
External Bank 3 Access Mode (EB3AM)	0
External Bank 3 number of Wait States (EB3WS)	0

The Access Mode means that the processor uses asynchronous accesses. No wait states are required as the CPLD signals the SHARC when the PLX is ready.

The base address for the PLX PCI 9054 is 0xB00000 as the MSIZE parameter in the SYSCON register of the SHARC is set to 7.

The PLX PCI 9054 bus is connected to bits 63 to 32 of the 64-bit SHARC bus. This means that the PLX PCI 9054 registers must be accessed by longword accesses. If P is the PLX PCI 9054 local offset to a register (a byte offset) and S is the SHARC offset to a register (a word offset), then the relationship between the two is:

$$S = P \div 4$$

For example, to read the PLX Command and Status register (offset 0x04) use:

```
r0 = dm (0xB00001)(lw);
```

The register contents will be in register r1.

PCI Bus

The PCI bus is located in unbanked external memory with the address range0xC00000 to 0xFFFFFF.

The values required for PCI bus by the SHARC WAIT register are:

WAIT Parameter	Value
Unbanked memory Access Mode (UBAM)	0
Unbanked memory number of Wait States (UBWS)	1

The Access Mode means that the processor uses asynchronous accesses. No wait states are required as the CPLD signals the SHARC when the PCI bus is ready.

The PCI bus is connected to bits 63 to 32 of the 64-bit SHARC bus. This means that the PCI bus is accessed by longword accesses or by DMA transfers using 32 External to 32/64 Internal Packing mode.

The SHARC can only make 32-bit accesses to the PCI bus, not 8- or 16-bit accesses.

PCI Access to Internal RAM

The PCI bus can access the SHARC's internal RAM. The PCI bus should only attempt accesses to the IOP area and the normal addressing range of SHARC internal memory (0x40000 to 0x5FFFF).

If P is the PCI bus offset from the start of SHARC internal memory (a byte offset) and S is the SHARC offset (a word offset), then the relationship between the two is:

$$S = (P \div 4)$$

The PCI bus can only make 32-bit accesses to SHARC memory, not 8- or 16-bit accesses.

Peripheral Component Interconnect

The PCI is a high performance synchronous bus running at 33 MHz. The PMCHH1 implements a 32-bit PCI bus with burst data rates up to 132 Mbytes/second. The PCI bus provides the standard PMC interface to the PMC host card.

JTAG

There are two connectors on the PMCHH1 that carry JTAG-type signals: J1 and P3 (see Chapter 4 for more details on these connectors).

P3 JTAG

The JTAG signals from P3 connect to the CPLD. They are used solely for factory programming of this device.

J1 JTAG

This is the JTAG connection for the SHARC processor. The pinout for this connector is detailed in Chapter 4.

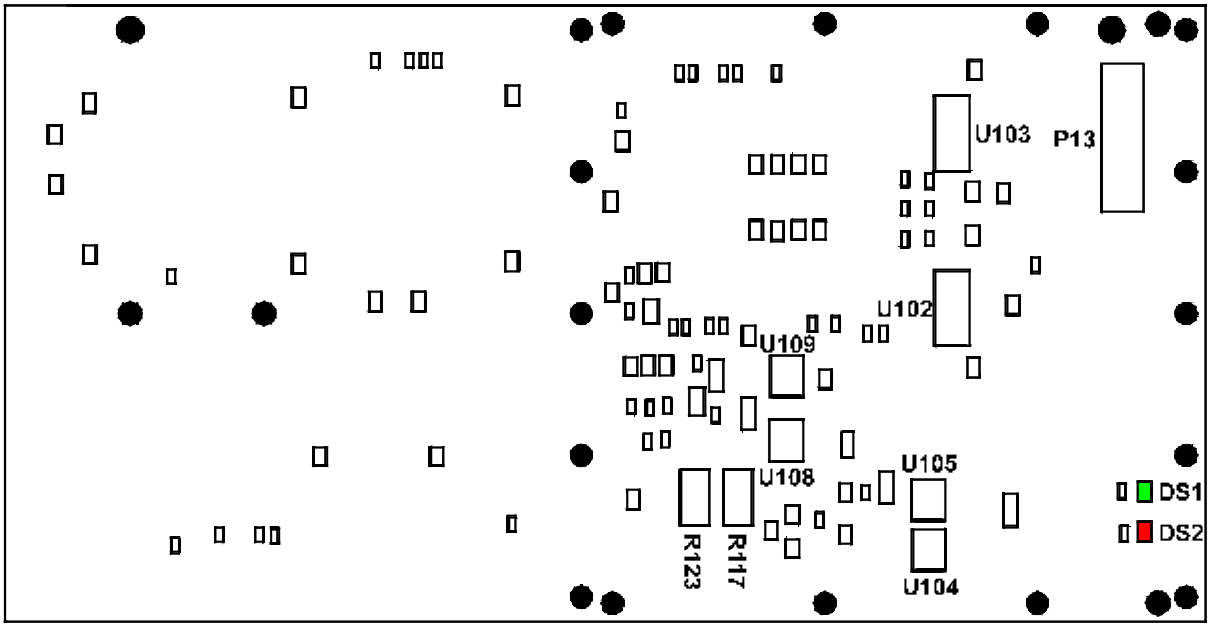
JTAG Clock Frequency

The main chain clock frequency should not exceed 10 MHz.

Software Programmable LEDs

There are two software programmable LEDs on the PMCHH1.

- The green LED (DS1) is controlled by flag pin 0
- The red LED (DS2) is controlled by flag pin 1



The LEDs are visible on the back of the PCB.

Control Register

The PMCHH1 Control register is located at address 0x80000000 in unbanked memory. The values required for the Control register by the SHARC WAIT register are:

WAIT Parameter	Value
Unbanked memory Access Mode (UBAM)	0
Unbanked memory number of Wait States (UBWS)	1

The access mode means that the processor uses asynchronous accesses.

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

D0: FLASH programming enable (read/write).

1 = Enabled
0 = Disabled

D1 to D7: Reserved

Interrupts

The SHARC has three external interrupt lines that are connected as follows:

SHARC Interrupt Pin	Priority	Source
IRQ0	0	PLX PCI 9054 Local Interrupt Pin LINT~
IRQ1	1	PLX PCI 9054 System Error Interrupt Output Pin LSERR~
IRQ2	2	PLX PCI 9054 General Purpose Output Pin USER

IRQ0 Interrupts

The PLX PCI 9054 can be configured to send DMA, Mailbox and PCI to Local Doorbell interrupts to the SHARC on IRQ0.

IRQ1 Interrupts

The PLX PCI 9054 can be configured to send Parity, Retry, Master Abort and Target Abort interrupts to the SHARC on IRQ1.

IRQ2 Interrupts

If the PLX PCI 9054 General Purpose Output pin is configured as an output, writing to the PLX to set the state of the output to low will send an interrupt to the SHARC on IRQ2.

PCI Interrupts

The SHARC can generate PCI INTA interrupts by writing to the PLX PCI 9054 Local to PCI doorbell register.

PCI Configuration

The PCI configuration of the PMCHH1 is set up by the host card.

Chapter 8 - Technical Assistance

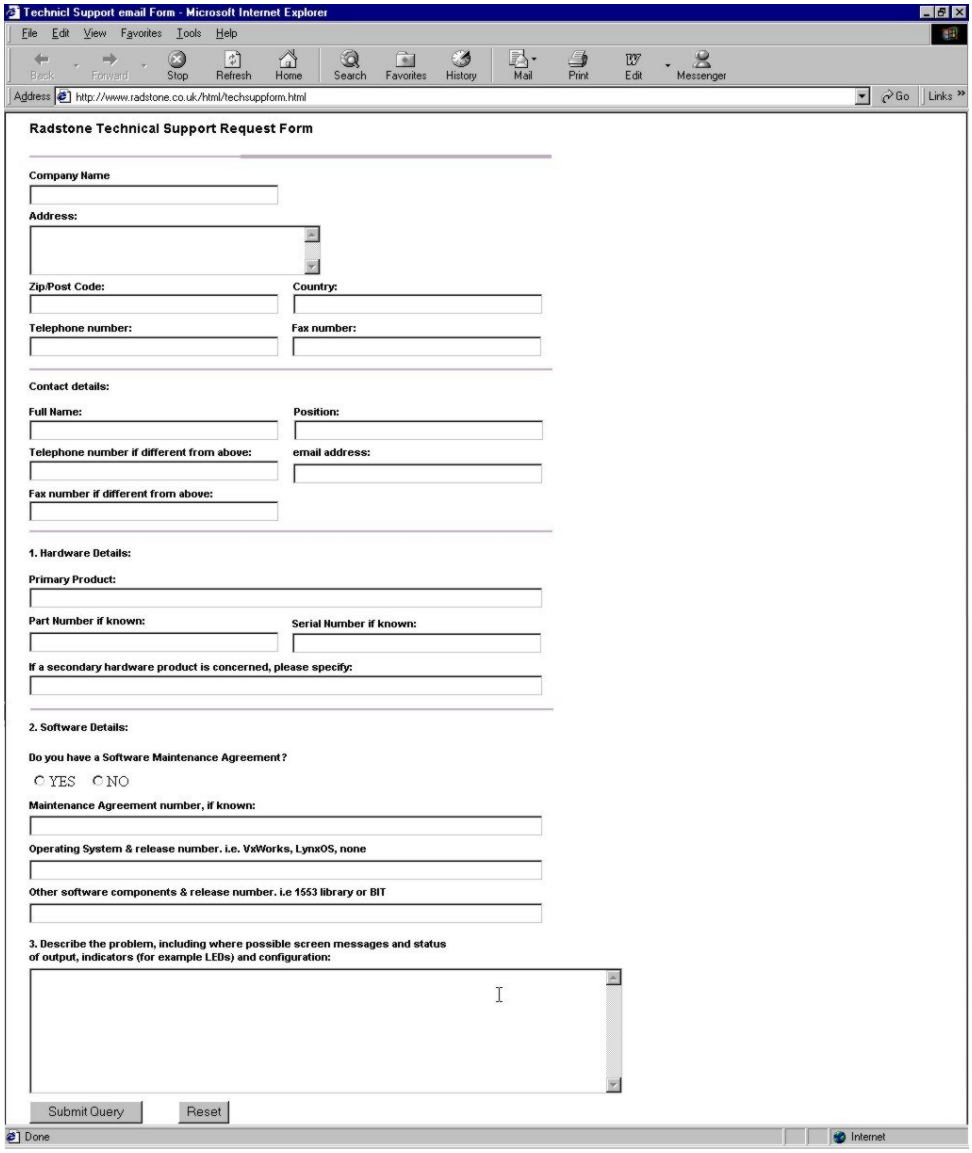
This chapter provides Radstone contact information (including details of information to have available when you call to report a problem) and some helpful hints for troubleshooting your PMCHH1.

Radstone Contact Information

Radstone does not anticipate that you will experience problems with your PMCHH1. However, in the unlikely event that you do have problems, for assistance please contact Radstone's Technical Help Line on::

☎ +44 1327 359804 outside the UK
01327 359804 in the UK

Alternatively, you may register support requests using the Technical Support Request Form available through the Radstone web site at http://www.radstone.co.uk. This form looks like:



If you telephone for technical support, please be prepared to provide the same information as required above.

Your query will be logged on Radstone's Technical Support database and allocated a unique Call Reference Number (CRN) for use in future correspondence.

Helpful Hints

If you are experiencing a problem with your PMCHH1, there follow some general suggestions of actions you may take, which may resolve the problem without the need to contact Radstone's technical support.

If you have fitted the PMCHH1 to a Radstone host or carrier card, you should also follow the suggestions in the troubleshooting chapter in the appropriate host or carrier card's manual. By doing this, you can eliminate the host/carrier as being the source of the problem and narrow the scope to the PMCHH1 itself or the PMCHH1 to host/carrier connection.

To make doubly sure that it is not the host that is causing the problem, if you have not already done so, ensure that the host operates successfully in isolation (i.e. without the PMCHH1 fitted).

Having satisfied yourself that the source of the problem is the PMCHH1, try the following suggestions.

- Ensure that there is no obvious damage to the PMCHH1 (e.g. broken or missing components), as mentioned in Chapter 3
- Ensure that the PMCHH1 is properly connected to the PMC host or carrier card, with the P11 and P12 connectors properly located and firmly fitted. Ensure that the PMCHH1 is firmly attached to the host/carrier and (conduction-cooled versions only) the centre stiffening bar

These suggestions only apply if you are fitting the PMCHH1 to a PMC host yourself. If Radstone has fitted your PMCHH1, it will have been extensively factory tested to provide correct operation. Please contact Radstone if this is not the case.

These suggestions mainly apply if you are fitting the PMCHH1 to a Radstone PMC host. If you are fitting the PMCHH1 to another manufacturer's PMC host, although the suggestions still apply in general, there may be different or additional requirements for the PMCHH1/PMC host combination. See the appropriate manufacturer's PMC host documentation for more details.

Appendix A - Specifications

This appendix gives a specification of the PMCHH1. It also covers items such as the power requirements, the MTBF, the general measurements etc. and lists the available variants.

General Specifications

PCI Bus Interface

Compliance	IEEE P1386.1 and VITA 20-199x (PMC) standard
Environment	5V supply and signalling, 33 MHz clock
Type	32-bit Master/Slave
Stacking Height	10 mm

Connectors

P11/P12	EIA E700 AAAB. Provides the control signals and address/data of the PMC bus

Local Memory

FLASH	8 Mbytes, 1Mword x 32-bit devices

Software

Operating system support	VxWorks/Tornado
	LynxOS
	Virtuoso

Dimensions

PMC Slots	1
Air cooled	149 x 74 mm
Conduction cooled	143.75 x 74 mm

Local Resources

Processor	1 Analog Devices ADSP-21160 SHARC DSP at 80 MHz. Total peak performance 480 MFLOPS
On-board Memory	Memory resources available to the ADSP-21160 and PCI interface: 8 Mbytes of word-wide on-board programmable FLASH.
PCI Interface	PLX 9054 32-bit/33 MHz PCI to ADSP-21160 interface
Link Ports	The ADSP-21160 provides six 80 Mbytes/second, 8-bit wide bi-directional link ports at the front panel
Serial I/O	One SHARC serial port via the Front Panel connector.
Shared Bus	400 Mbytes/second ADSP-21160 shared bus for ADSP-21160, FLASH and PCI
JTAG Interface	IEEE 1149.1 compatible JTAG interface, available for development and de-bug. Supports Analog Devices' EZ-ICE system
Status LEDs	2 software programmable LEDs (DS1 is green, DS2 is red) front panel mounted on all levels
Form Factor	PMC: IEEE P1386 Draft 2.0/VITA 20-199x Draft 1.6
Operating Systems	Virtuoso

EMC Regulatory Compliance and Safety

Air-cooled versions of the PMCHH1 are designed for use in systems meeting VDE class B, EN and FCC regulations for EMC emissions and susceptibility.

Conduction-cooled versions are designed for integration into EMC hardened cabinets/boxes.

All PCBs are manufactured by UL approved manufacturers and have a flammability rating of 94V-0.

Electrical Specification

Power Consumption

The power consumption is typically 2W at +5V. Power supplies and ground are shared between the connectors.

Voltage Requirements

+5V +5% -2.5% DC total excursion, including all transients
+12V +5% -2.5% DC total excursion, including all transients

Vripple (+5V) = 50mV RMS maximum contained in the total excursion
Vripple (+12V) = 120mV RMS maximum contained in the total excursion

Reliability

Using MIL-HBK-217F Notice 1 as a database and the parts count method, the following estimates have been made of the MTBF for the PMCHH1-11 (see the Ordering Information section).

Environment	Failure Rate (failures per million hours)	MTBF (Hours)
Ground Benign 30°C	1.487494	672,272
Ground Fixed 40°C	4.19518	238,369
Ground Mobile 45°C	7.92194	126,232
Naval Sheltered 40°C	6.77546	147,591
Naval Unsheltered 45°C	11.14312	89,741
Airborne Inhabited Cargo 55°C	9.42769	106,071
Airborne Inhabited Fighter 55°C	12.76078	78,365
Airborne Uninhabited Cargo 70°C	18.32964	54,556
Airborne Uninhabited Fighter 70°C	24.6409	40,583
Space Flight 30°C	1.394253	717,230
Missile Launch 55°C	24.4679	40,870
Missile Flight 45°C	9.59048	104,270
Airborne Rotary Wing 55°C	17.87208	55,953

Environmental Specifications

The PMCHH1 will operate and may be stored or transported without damage within the same limitations as the host board.

Convection-cooled Boards

Build Style	Operating Temp (°C)	Storage Temp (°C)	Vibration	Shock	Humidity	Comments
Standard (Level 1)	0 to +55 with airflow of 300 feet/minute	-50 to +100	$0.002g^2$/Hz from 10 to 2000 Hz random, and 2g sinusoidal from 5 to 500 Hz	20g peak sawtooth, 11ms duration	Up to 95% RH	Commercial grade cooled by forced air, for use in benign environments and software development applications.
Extended Temp (Level 2)	-20 to +65 with airflow of 300 feet/minute	-50 to +100	$0.002g^2$/Hz from 10 to 2000 Hz random, and 2g sinusoidal from 5 to 500 Hz	20g peak sawtooth, 11ms duration	Up to 95% RH with varying temperature. 10 cycles, 240 hours	As Standard but conformally coated and temperature characterised.
Rugged Air-cooled (Level 3)	-40 to +75 with airflow of 600 feet/minute	-50 to +100	$0.04g^2$/Hz from 10 to 2000 Hz with a flat response to 1000 Hz. 6db/Octave roll-off from 1000 to 2000 Hz	20g peak sawtooth, 11ms duration	Up to 95% RH with varying temperature. 10 cycles, 240 hours	Wide temperature rugged, cooled by forced air. Conformally coated for additional protection.

Conduction-cooled Boards

Build Style	Operating Temp (°C)	Storage Temp (°C)	Vibration	Shock	Humidity	Comments
Rugged Conduction-cooled (Level 4)	-40 to +75 at the thermal interface	-50 to +100	Random $0.1g^2$/Hz from 5 to 2000 Hz per MIL-STD-810E	40g peak sawtooth, 11ms duration	Up to 95% RH with varying temperature. 10 cycles, 240 hours	Mechanically compliant with IEEE 1101.2-1992. Designed for severe environment applications with high levels of shock and vibration, small space envelope and restricted cooling supplies. Conformally-coated as standard. Optional ESS.
Rugged Conduction-cooled (Level 5)	-40 to +85 at the thermal interface	-50 to +100	Random $0.1g^2$/Hz from 5 to 2000 Hz per MIL-STD-810E	40g peak sawtooth, 11ms duration	Up to 95% RH with varying temperature. 10 cycles, 240 hours	Mechanically compliant with IEEE 1101.2-1992. Designed for severe environment applications with high levels of shock and vibration, small space envelope and restricted cooling supplies. Conformally-coated as standard. Optional ESS.

Mechanical Specification

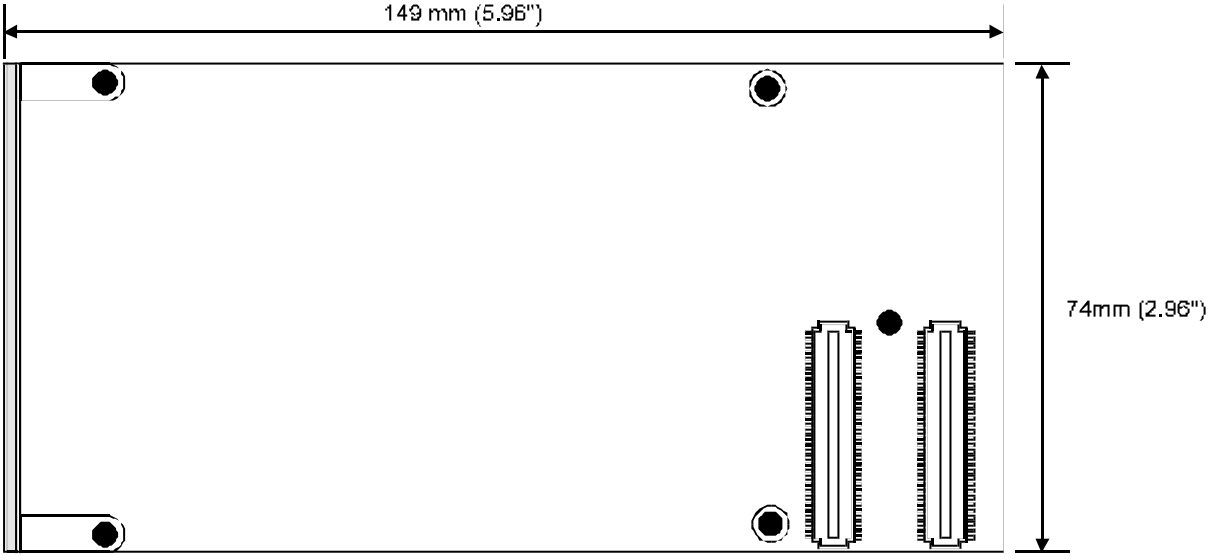
Weight

The approximate weight of the PMCHH1 is 67g.

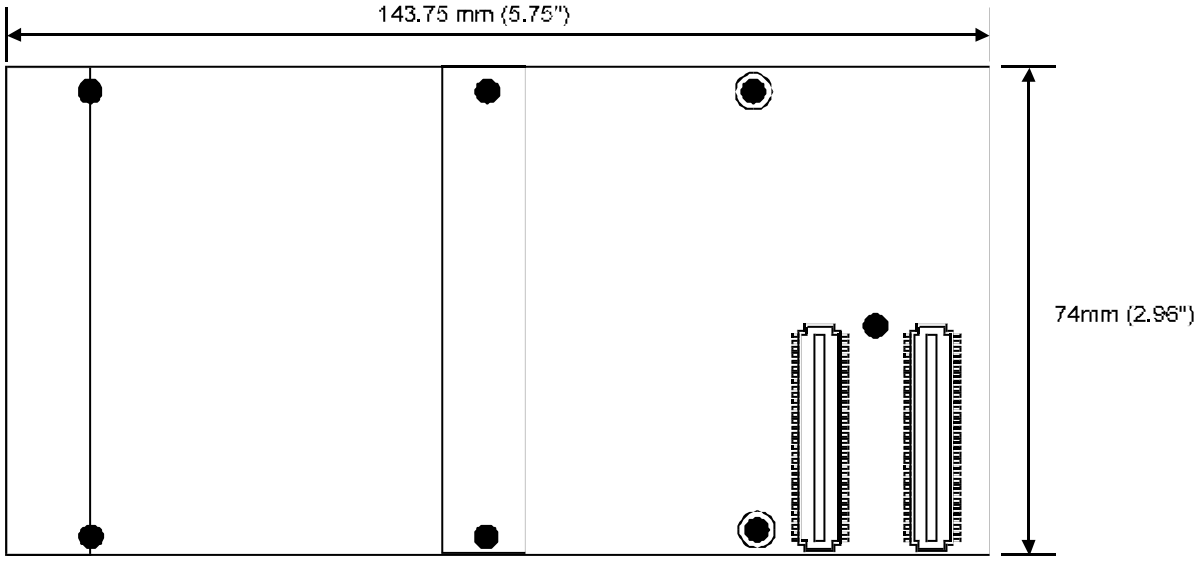
Dimensions

Each build standard of the PMCHH1 conforms to the appropriate (draft conduction cooled) PMC specification. This allows all styles of assembly to be fitted to any PMC compatible host.

The following diagrams show the overall dimensions in millimeters, with inches (in parentheses) for general guidance only:



Air cooled PMCHH1



Conduction cooled PMCHH1

Ordering Information

The order code is specified as PMCHH1-**x**1, where **x** is 1 to 5 to select the corresponding build level. See the Environmental Specifications section for more details on the build styles.

Operating Systems

Availability includes Virtuoso. Please contact your nearest Radstone sales office for more details.

Appendix B - Example Code

This appendix lists sample code to:

1) Erase FLASH
2) Program FLASH
3) Check the FLASH status
4) Initialise the PLX
5) Carry out a DMA write from SRAM to the PCI bus

In the following code segments, which can be called from C, the register usage shown below is defined by the C compiler:

m13 = 0

m14 = 1

m7 = -1

i6 and i7 are for stack control

r0, r1, r2, r4, r8, r12, i12, i4 are scratch registers.

Erase FLASH

The following assembly code function erases a sector of FLASH memory. It can be called from C as:

```
erase_flash_sector(addr);
```

where `addr` is the start address of the sector to be erased. The parameter is passed to the assembly code in register r4.

```
_erase_flash_sector:
        i12 = r4;
        /* Enable FLASH programming */
        r1 = dm(0x80000000);
        r0 = 0x1;
        r1 = r0 OR r1;
        dm(0x80000000) = r1;
        /* Erase FLASH sector */
        px2 = 0x00200020;
        px1 = 0x00200020;
        pm(i12, m13) = px;
        px2 = 0x00D000D0;
        px1 = 0x00D000D0;
        pm(i12, m13) = px;
        /* Wait for erase to complete */
        cjump (PC, _check_flash_status) (DB);
        dm(i7,m7) = r2;
        dm(i7,m7) = pc;
        /* Return FLASH sector to read mode */
        px2 = 0x00FF00FF;
        px1 = 0x00FF00FF;
        pm(i12, m13) = px;
        /* Disable FLASH programming */
        r2 = dm(0x80000000);
        r1 = 0xFFFFFFFE;
        r2 = r1 AND r2;
        dm(0x80000000) = r2;
        /* Return from function */
        i12 = dm(m7, i6);
        jump (m14, i12) (DB);
        nop;
        rframe;
```

Program FLASH

The following assembly code function programs a block of 48-bit data to FLASH memory. It can be called from C as:

```
program_block (dest, source, length);
```

where `dest` is the FLASH address to start programming, `source` is the start address of the data to be programmed, and `length` is the number of words to be programmed. The parameters are passed to the assembly code in registers r4, r8 and r12 respectively.

```
_program_block:

        /* Save registers to stack */
        dm(i7, m7) = l13;
        dm(i7, m7) = i13;

        /* Disable circular buffers */
        l12 = 0;
        l13 = 0;

        /* Set up FLASH start address to program and start of source data */
        i12 = r4;
        i13 = r8;

        /* Enable FLASH programming */
        r1 = dm(0x80000000);
        r0 = 0x1;
        r1 = r0 OR r1;
        dm(0x80000000) = r1;

        /* Program block to FLASH */
        lcntr = r12, do program_block_loop until lce;

        /* Write program code to address to program */
        px2 = 0x00400040;
        px1 = 0x00400040;
        pm(m13, i12) = px;
        nop;
        nop;

        /* Read data to program */
        px = pm(i13, m14);

        /* Write data to program to FLASH address */
        pm(m13, i12) = px;

        /* Wait for programming of address to finish */
        r4 = i12;
        cjump (PC, _check_flash_status) (DB);
        dm(i7, m7) = r2;
        dm(i7, m7) = pc;

        /* Return FLASH sector to read mode */
        px2 = 0x00FF00FF;
        px1 = 0x00FF;

        program_block_loop: pm(i14, m14) = px;

        /* Disable FLASH programming */
        r2 = dm(0x80000000);
        r1 = 0xFFFFFFFE;
```

```
r2 = r1 AND r2;
dm(0x80000000) = r2;

/* Restore registers from stack */
l13 = dm(2, i7);
i13 = dm(1, i7);

/* Return from function */
i12 = dm(m7, i6);
jump (m14, i12) (DB);

nop;
rframe;
```

Check FLASH Status

The following assembly code function checks the status of FLASH memory. It can be called from C as:

```
check_flash_status (addr);
```

where `addr` is the FLASH address to check. The parameter is passed to the assembly code in register r4.

```
_check_flash_status:
        r0 = 0x00800080;
        r1 = 0x00800080;
        i12 = r4;
        not_ready:
        /* Read FLASH status */
        px = pm(m13, i12);
        /* Check ms 32 bits */
        r2 = px2;
        comp(r0,r2);
        if ne jump (PC, not_ready);
        /* Check ls 16 bits */
        r2 = px1;
        comp(r1, r2);
        if ne jump (PC, not_ready);
        i12 = dm(m7, i6);
        jump (m14, i12) (DB);
        nop;
        rframe;
```

Initialise PLX

The following assembly code segment demonstrates an example initialisation of the PLX chip. This example enables PCI to local memory accesses of SHARC internal memory. The PCI system master sets up the PCI address used for these accesses. It also maps local address 0xC00000 to PCI address 0x10000000 for local to PCI accesses. For these transfers, the PLX must be set up so it can be a PCI bus master.

Set the PCI to local memory range to 2 Mbytes:

```
r0 = 0;
r1 = 0xFFE00000;
dm(0xB00020) = r0(lw);
```

Set the local base address for PCI accesses to internal SHARC memory (SHARC address 0x0):

```
r1 = 0x1;
dm(0xB00021) = r0(lw);
```

Set the local base addresses for memory and I/O accesses to PCI:

```
r1 = 0x03000000;
dm(0xB00028) = r0(lw);
dm(0xB00029) = r0(lw);
```

Set the local to PCI range to 2 Mbytes:

```
r1 = 0xFFE00000;
dm(0xB00027) = r0(lw);
```

Enable local to PCI memory accesses starting at PCI base address 0x10000000:

```
r1 = 0x10000001;
dm(0xB0002A) = r0(lw);
```

Enable PLX to be a PCI bus mater:

```
r0 = dm(0xB00001)(lw);
r2 = 0x4;
r1 = r2 OR r1;
dm(0xB00001) = r0(lw);
```

PLX DMA Programming

The following assembly code segment demonstrates an example of setting up a DMA from internal memory to the PCI bus.

In this example, data is transferred starting from local address 0x50000 to PCI address 0x10000000.

Set up the PCI destination address:

```
r0 = 0;
r1 = 0x10000000;
dm (0xB00041) = r0(lw);
```

Set up the local memory source address:

```
r1 = 0x00140000;
dm (0xB00042) = r0(lw);
```

Set up the data length in bytes:

```
r1 = 0x100;
dm (0xB00043) = r0(lw);
```

Set up the transfer direction - to PCI in this case:

```
r1 = 0x8;
dm (0xB00044) = r0(lw);
```

Set up the local bus width, wait states and enable READY#:

```
r1 = 0x43;
dm (0xB00040) = r0(lw);
```

Start the transfer:

```
r1 = 0x3;
dm (0xB0004A) = r0(lw);
```

Wait for the transfer to complete:

```
dma_not_complete:
r0 = dm(0xB0004A)(lw);
btst r1 by 0x4;
if sz jump (pc, dma_not_complete);
```

Appendix C - BIT Fail Data

This appendix details the PMCHH1 fail data associated with each test module. Each test module within the overall test has a mask code associated with it (the test mask in the System Description entry). Setting a particular bit in the test mask causes that test to be ignored, e.g. setting the Self-Test test mask to 0x00000F7B would only run the “Serial Port” and “FLASH Checksum” tests. The table below lists the tests with their test mask:

Self-Tests:

Test	Mask Code
Local Memory	0x00000001
SRAM	0x00000002
FLASH Checksum	0x00000004
PLX	0x00000008
Timer	0x00000010
Link Port	0x00000020
External Port	0x00000040
Serial Port	0x00000080
Interrupt	0x00000100
Link Connection	0x00000200
Serial Connection	0x00000400
PCI	0x00000800

Local Memory Test (Mask Code = 0x00000001)

Fail Code	Event Code	Reason	Fail Data
00000010	00000001 00000002 00000003 00000004	Failed local RAM 64-bit checksum test Failed local RAM 48-bit checksum test Failed local RAM 32-bit checksum test Failed local RAM 16-bit checksum test	00 = First checksum value 01 = Second checksum value 02 = Third checksum value 03 = Start address of test area 04 = End address of test area
00000020	00000001 00000002 00000003 00000004	Failed local RAM 64-bit destructive memory test Failed local RAM 48-bit destructive memory test Failed local RAM 32-bit destructive memory test Failed local RAM 16-bit destructive memory test	00 = Number of the test loop that failed 01 = Pointer to the memory location that failed 02 = Expected data 03 = Actual data read back 04 = Start address of test area 05 = End address of test area
00000020	00000005 00000006 00000007 00000008	Failed local RAM 64-bit zero memory test Failed local RAM 48-bit zero memory test Failed local RAM 32-bit zero memory test Failed local RAM 16-bit zero memory test	00 = Pointer to the memory location that failed 01 = Expected data 02 = Actual data read back 03 = Start address of test area 04 = End address of test area

SRAM Test (Mask Code = 0x00000002)

There is no SRAM on the PMCHH1 so the test always reports passed.

FLASH Checksum Test (Mask Code = 0x00000004)

Fail Code	Event Code	Reason	Fail Data
00000010	00000001	One of the USER FLASH sectors failed	00 = Sector number that failed 01 = Checksum that was calculated 02 = Checksum that was expected
00000020	00000001	The configuration sector failed	00 = Sector number that failed (should be 31) 01 = Checksum that was calculated 02 = Checksum that was expected

PLX 9054 Test (Mask Code = 0x00000008)

Fail Code	Event Code	Reason	Fail Data
00000010	00000001	PLX ID error	00 = Expected ID value 01 = Read ID value
00000010	00000002 00000003	PLX register walking ‘1’ test failed PLX register walking ‘0’ test failed	00 = Register address 01 = Data that was written to the register 02 = Data that was read from the register

SHARC Timer Test (Mask Code = 0x00000010)

Fail Code	Event Code	Reason	Fail Data
00000010	00000001 00000002 00000003	SHARC timer failed SHARC timer high priority interrupt failed SHARC timer low priority interrupt failed	No fail data

Link DMA Test (Mask Code = 0x00000020)

Fail Code	Event Code	Reason	Fail Data
00000010 00000020	00000001 00000001	Link port DMA failed Link port chained DMA failed	00 = Iteration that failed 01 = Link port/buffer that failed 02 = Data size (0 = 32 bit, 1 = 48 bit)
00000010 00000020	00000002 00000002	Link port DMA data error Link port chained DMA data error	00 = Iteration that failed 01 = Link port/buffer that failed 02 = Data size (0 = 32 bit, 1 = 48 bit) 03 = Offset of data that failed 04 = Expected data value 05 = Actual data value 06 = Expected data value (LS 16 bits for 48 bit data) 07 = Actual data value (LS 16 bits for 48 bit data)

External Port Test (Mask Code = 0x00000040)

There is no SRAM on the PMCHH1 so the test always reports passed.

Serial Port Test (Mask Code = 0x00000080)

Fail Code	Event Code	Reason	Fail Data
00000010 00000020	00000001 00000001	Serial port DMA failed Serial port chained DMA failed	00 = Iteration that failed 01 = Serial port/buffer that failed 02 = Test code
00000010 00000020	00000002 00000002	Serial port DMA data error Serial port chained DMA data error	00 = Iteration that failed 01 = Serial port/buffer that failed 02 = Test code 03 = Offset of data that failed 04 = Expected data value 05 = Actual data value

The test code defines which serial port mode was being tested when the failure occurred. The different codes refer to:

- Frame sync active high or low (H/L)
- Frame sync early or late (E/L)
- Frame sync required (Y/N)
- Sampling on rising or falling clock edges (R/F)

The meaning of the codes is as follows:

0 = HENF
1 = HEYF
2 = HENR
3 = HEYR
4 = LENF
5 = LEYF
6 = LENR
7 = LEYR
8 – HLNF
9 = HLYF
10 = HLNR
11 = HLYR
12 = LLNF
13 = LLYF
14 = LLNR
15 = LLYR

Interrupt Test (Mask Code = 0x00000100)

Fail Code	Event Code	Reason	Fail Data
00000010	00000001	IOP access interrupt failed	00 = Iteration
00000010	00000002	Unaligned 64-bit access interrupt failed	00 = Iteration
00000010	00000003	PC stack full interrupt failed	00 = Iteration
00000010	00000004	Fixed-point interrupt failed	00 = Iteration
00000010	00000005	Floating point overflow interrupt failed	00 = Iteration
00000010	00000006	Floating point underflow interrupt failed	00 = Iteration
00000010	00000007	Floating point invalid interrupt failed	00 = Iteration
00000010	00000008	Circular buffer 7 overflow interrupt failed	00 = Iteration
00000010	00000009	Circular buffer 15 overflow interrupt failed	00 = Iteration
00000010	0000000A	Software interrupt 0 failed	00 = Iteration
00000010	0000000B	Software interrupt 1 failed	00 = Iteration
00000010	0000000C	Software interrupt 2 failed	00 = Iteration
00000010	0000000D	Software interrupt 3 failed	00 = Iteration
00000010	0000000E	External interrupt 2 failed	00 = Iteration
00000010	0000000F	External interrupt 1 failed	00 = Iteration

Link Connection Test (Mask Code 0x00000200)

Fail Code	Event Code	Reason	Fail Data
00000010	00000001	Link master initial data transmission error	00 = Iteration 01 = Link that failed
00000010	00000002	Link master initial data no data error	00 = Iteration 01 = Link that failed
00000010	00000003	Link master initial data data error	00 = Iteration 01 = Link that failed 02 = Expected data 03 = Actual data
00000010	00000004	Link master data error	00 = Iteration 01 = Link that failed 02 = Expected data 03 = Actual data 04 = Data index
00000010	00000005	Link master failed	00 = Iteration 01 = Link that failed 02 = Status code (see list 1 below)
00000010	00000006	Link master unexpected interrupt	00 = Iteration 01 = Link that failed 02 = Interrupt (1 = link interrupt, 2 = LSRQ interrupt) 03 = Status code (see list 1 below)
00000020	00000001	Link slave initial data transmission error	00 = Iteration 01 = Link that failed
00000020	00000002	Link slave initial data no data error	00 = Iteration 01 = Link that failed
00000020	00000004	Link slave data error	00 = Iteration 01 = Link that failed 02 = Expected data 03 = Actual data 04 = Data index
00000020	00000005	Link slave failed	00 = Iteration 01 = Link that failed 02 = Status code (see list 2 overleaf)
00000020	00000006	Link slave unexpected interrupt	00 = Iteration 01 = Link that failed 02 = Interrupt (1 = link interrupt, 2 = LSRQ interrupt) 03 = Status code (see list 2 overleaf)
00000030	00000001	LSRQ interrupt error	00 = LSRQ interrupt flag register
00000030	00000002	Link interrupt error	00 = Link port interrupt flag register

The status codes for the master links are:

1) Started iteration, sent initial data.
2) Initial data sent.
3) Waiting to receive initial data back.
4) Initial data received.
5) DMA send started.
6) DMA send completed.

The status codes for the slave links are:

1) Started iteration, waiting for LSRQ interrupt.
2) Initial data received and sent.
3) Receive DMA initialised.
4) Receive DMA completed.
5) Send DMA started.

Serial Connection Test (Mask Code = 0x00000400)

Fail Code	Event Code	Reason	Fail Data
00000010	00000001	Serial port initial data failed	00 = Iteration 01 = Port that failed 02 = Expected data 03 = Actual data
00000010	00000002	Serial port data failed	00 = Iteration 01 = Port that failed 02 = Expected data 03 = Actual data
00000010	00000003	Serial port transfer failed	00 = Iteration 01 = Port that failed 02 = Status (see below)

The status codes are as follows:

1) Iteration started
2) Initial data sent
3) Initial data received
4) DMA transfer started
5) DMA transfer sent
6) DMA transfer received

PCI Test (Mask Code = 0x00000800)

Fail Code	Event Code	Reason	Fail Data
00000010	00000001	Data error	00 = Iteration that failed 01 = Pattern index 02 = Expected data 03 = Actual data 04 = Address offset
00000020 00000020	00000001 00000002	Local DMA write failed Local DMA read failed	00 = Iteration that failed 01 = Pattern index
00000020	00000003	Local DMA data error	00 = Iteration that failed 01 = Pattern index 02 = Expected data 03 = Actual data 04 = Address offset
00000030 00000030	00000001 00000002	PLX DMA write failed PLX DMA read failed	00 = Iteration that failed 01 = Pattern index 02 = PLX DMA channel 03 = Abort status
00000030	00000003	PLX DMA data error	00 = Iteration that failed 01 = Pattern Index 02 = PLX DMA channel 03 = Expected data 04 = Actual data 05 = Address offset
00000040 00000040	00000001 00000002	PLX chained DMA write failed PLX chained DMA read failed	00 = Iteration that failed 01 = Pattern Index 02 = PLX DMA channel 03 = Abort status
00000040	00000003	PLX chained DMA data error	00 = Iteration that failed 01 = Pattern Index 02 = PLX DMA channel 03 = Expected data 04 = Actual data 05 = Address offset
00000050	00000001	Data error	00 = Iteration that failed 01 = Pattern index 02 = Expected data 03 = Actual data 04 = Address offset
00000050	00000002	Malloc of data buffer failed	No fail data

Glossary

See also the general Radstone Glossary.

BIT
Built-In-Test.

BSF
Board Support Firmware.

DSP
Digital Signal Processing/Processor.

MPE
Minimum Processing Environment.

PMCHH1
Radstone's ADSP-21160 PCI to Link Port bridge PMC board.

SDE
System Description Environment.

SHARC
Super Harvard Architecture Computer.

VANTEGRA-2
Radstone's 12 ADSP-21160 DSP VME board.

Index

A

Antistatic Precautions 3-1

B

BIT 5-3, 6-3
- *See Also* ..... MPE
- Configuring Host 6-3
- Control Block 6-12
- Fail Data C-1
- Running 6-4
- Self Tests
  - FLASH 6-5, C-2
  - Interrupts 6-6, C-4
  - Link Connection 6-6, C-5
  - Link DMA 6-5, C-2
  - Local Memory 6-4, C-1
  - PCI 6-6, C-7
  - PLX 9054 ..... 6-5, C-2
  - Serial Connection 6-6, C-6
  - Serial Port 6-5, C-3
  - SHARC Timer 6-5, C-2
- Test Descriptions and Fail Codes 6-4

Block Diagram 2-2
Board Architecture 2-3, 7-2
Board ID 3-1
BSF 6-1
- Command Interface 6-7

Build Styles 2-5
- *See Also* ..... Environmental Specification

C

Code Examples B-1
- Check FLASH Status B-5
- Erase FLASH B-2
- Initialise PLX B-6
- PLX DMA Programming B-7
- Program FLASH B-3

Command Interface ..... *See* BSF
Compatibility 2-6
Component Layout 3-2
Configuration
- Link 5-1
  - Default 3-2
  - Emulator 5-1
  - Position 5-1

Configuration Data Memory Usage 6-9

C (continued)

Connectors 4-1, A-1
- FP1A 4-5
- J1 4-7
- P11 4-2
- P12 4-3
- P3 4-8
- PCI Bus Interface ..... *See* P11, P12
- Position Diagram 4-1

Contact Information 8-1
Control Register ..... *See* Registers

D

Dimensions A-5
Documentation Audience 1-2
Documentation Conventions 1-3
Documentation Objectives 1-2
Documentation Scope 1-2
Documentation Structure 1-2

E

Electrical Specification ..... *See* Specifications
EMC Regulatory Compliance A-2
Emulator Connector ..... *See* J1 Connector
Environmental Specification ..... *See* Specifications
Example Code ..... *See* Code Examples

F

Fail Data ..... *See* BIT
Features List 2-1
FLASH ..... *See* Memory
FP1A Connector ..... *See* Connectors
Functional Description 7-1
Functional Overview 2-3

H

Host Boot Mode 7-2
Host-SHARC Gateway 2-3
Humidity ..... *See* Environmental Specification

I

I/O Capabilities 2-3
Initialisation 5-3
Inspection 3-2
Installation 5-2
Interrupts 7-9
- PCI 7-9

Introduction 1-1

J

JTAG 7-6
- *See Also* ..... J1, P3 Connectors

L

LEDs 7-7
Link Port Connector ..... *See* FP1A Connector
Links ..... *See* Configuration

M

Mechanical Overview 2-6
Mechanical Specification ..... *See* Specifications
Memory
  FLASH 2-3, 7-4, A-1
    See Also ..... Code Examples
    Usage 6-15
  Maps
    PLX Local Memory Space 7-3
    Processor 7-3
MPE 6-1
MTBF ..... *See* Reliability

N

No Boot Mode 7-2

O

Operating Environment
  See ..... Environmental Specification
Operating Systems ..... *See* Software Support
Operational Description 5-3
Ordering Information A-6

P

Part Number ..... *See* Board ID
PCI 7-6
  Access to SHARC RAM 7-6
  Bridge Controller 2-3
    See Also ..... PLX PCI 9054
  Bus 7-5
    Access 6-13
    Interface A-1
      See Also ..... Connectors
  Configuration 7-9
  Interrupts ..... *See* Interrupts
PLX PCI 9054 ..... 2-3, 7-5
  Configuration 6-16
  Registers ..... *See* Registers
PMC Signal Descriptions 4-4
Power
  Consumption A-3
  Up 5-3
Problems ..... *See* Technical Support
Product Overview 2-1

R

Registers
  Control 7-8
  PLX 6-16
Related Documents 1-4
Reliability A-3
Resets 7-2
Revision State ..... *See* Board ID
Ruggedization ..... *See* Build Styles

S

Safety A-2
Self Test ..... *See* BIT
Serial Number ..... *See* Board ID
SHARC Processor 2-3, 7-1
SHARC-Host Gateway 2-3
Shock ..... *See* Environmental Specification
Size ..... *See* Dimensions
Software Support 2-4, A-1
Specifications A-1
  Electrical A-3
  Environmental A-4
  General A-1
  Mechanical A-5
Storage Environment
  See ..... Environmental Specification

T

Technical Assistance 8-1
  Telephone Number 8-1

V

Vibration ..... *See* Environmental Specification
Virtuoso ..... *See* Software Support
Voltage Requirements A-3

W

Warnings 1-3
Weight A-5
World Wide Web Sites 1-4

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP