

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.

Find the *Emerson / Motorola / Force Computers CPCI-9120* at our website: ***Click HERE***



IXP/CPCI-9120

Reference Guide

6806800A77B

October 2006

© Copyright 2006 Motorola, Inc.

All rights reserved.

Printed in the United States of America.

Trademarks

Motorola® and the stylized M logo are trademarks of Motorola, Inc., registered in the U.S. Patent and Trademark Office.

All other names, products, or services mentioned in this document may be trademarks or registered trademarks of their respective holders.

Notice

While reasonable efforts have been made to assure the accuracy of this document, Motorola, Inc. assumes no liability resulting from any omissions in this document, or from the use of the information obtained therein. Motorola reserves the right to revise this document and to make changes from time to time in the content hereof without obligation of Motorola to notify any person of such revision or changes.

Electronic versions of this material may be read online, downloaded for personal use, or referenced in another document as a URL to the Motorola ECC website. The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of Motorola, Inc.

It is possible that this publication may contain reference to or information about Motorola products (machines and programs), programming, or services that are not available in your country. Such references or information must not be construed to mean that Motorola intends to announce such Motorola products, programming, or services in your country.

Limited and Restricted Rights Legend

If the documentation contained herein is supplied, directly or indirectly, to the U.S. Government, the following notice shall apply unless otherwise agreed to in writing by Motorola, Inc.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data clause at DFARS 252.227-7013 (Nov. 1995) and of the Rights in Noncommercial Computer Software and Documentation clause at DFARS 252.227-7014 (Jun. 1995).

Motorola, Inc.
Embedded Communications Computing
2900 South Diablo Way
Tempe, Arizona 85282

Using this Guide

Other Sources of Information

Safety Notes

Sicherheitshinweise

1 Introduction

32

Features 33

Standard Compliance 35

Ordering Information 36

Product Nomenclature 36

Order Numbers 36

2 Installation

Action Plan 41

Preparing for Installation 42

Take Antistatic Precautions 42

Unpacking Kit Contents 42

Accessories 43

Requirements 44

Environmental Requirements 44

Power Requirements 46

Hardware Upgrades and Accessories 47

IO Daughtercard 47

IO-920 47

PMC Module 47

Installation and Removal Procedure 48
LA1-920/8M 48

Switch Settings 49
Installing PMC 50
Removing PMC 53

IXP/CPCI-9120 Board Installation 54
Installation in a Non-Powered System 54
Installation Procedure 54
Removal Procedure 57
Control Elements 57
Installation in Basic Hot Swap System 58
Installation in Full Hot-Swap System 59
Removal from Full Hot-Swap System 60

3 Controls, Indicators and Connectors

Front Panel 63
LEDs 65
Keys 66
Front Panel Reset 66
Connectors 66
D-Type Subminiature Connector 66
On-Board Connectors 67
CompactPCI Connectors 67
J1 and J2 67
J3 67
J4 68
J5 68
PrPMC Connectors 68
MSF Mezzanine Connector 69
LA-1 Mezzanine Connector 77

4 Devices' features and Data Paths

Block Diagram 83
The IXP2400 83
PCI Controller 84

DDR Controller 84
QDR SRAM Interface 84
Media and Switch Fabric Interface 84
PHY Modes Supported 84
Serial Port 85
PCI Bus 85
Gigabit Ethernet Controller 85
Processor PMC 85
PCI Bus Signaling on the CompactPCI bus 86
Hot Swap Support 86
Packet Switching Backplane 86
Programmable devices 86
BootROM Flash 86
User Flash 86
IBMU 86
ICMB 87

5 Daughter Card Details

Media Switch Fabric mezzanine 91
IO-920 91
Quad Gigabit Ethernet MAC 91
Quad Gigabit Ethernet Transceiver 91
IO-920 Variants 92
Two 1000Base-T PSB (Single Slot Complex PSB Ethernet variant) 92

LA-1/QDR mezzanine 93
LA1-920/8M 93
I2C Interface 93

6 Maps and Registers

PCI memory map 97

SlowPort map 98

I2C memory map 101
NPU I2C memory map 101

SENSE I2C memory map 102
BIB I2C Memory Map 102

Register Details 103
General Purpose EPLD 103
Variant ID 103
Version Number 103
Board Revision Number 104
User Flash Select 104
CompactPCI Register 104
PrPMC register 105
System specific register 105
User LED Control register 106
Watchdog Timer Register 107
Watchdog Timer Interrupt Clear Register 107
Watchdog Enable and Watchdog Reset Enable Register 107
Watchdog Time Set Register 108
Interrupt Mask Register-1 108
Interrupt Mask Register-2 109
Interrupt Mask Register-3 110
Interrupt Status Register-1 110
Interrupt Status Register-2 111
Interrupt Status Register-3 112
I2C MUX Reset Register 112
MAC EPLD-IO-920 113
Variant ID 113
Version Number 113
Board Revision Number 113
PHY Reset Control Register 114
MUX Control Register 114
MAC PAUSE Address Register 115
MAC PAUSE Control Register 115
Interrupt Mask Register 115
Interrupt Status Register 116

7 RedBoot

Action Plan 119

Introduction 120

Building Images 121

Loading Images from RedBoot 122
RedBoot 122
Diagnostics 122

RedBoot Commands 123
Common Commands 123
Arguments 123

Download Process 132
Arguments 132

Flash Image System (FIS) 133
Arguments 134
Arguments 135

Executing Programs from RedBoot 140
Arguments 140

8 Diagnostics

Introduction 145
Requirements 145
Setup 145
Procedure 145
Configuring the Platform 146
Executing Diagnostics 146
Board Diagnostics Commands 146

A Troubleshooting

Index

Table 1 Organization of this Manual 15

Introduction

Table 1 Ordering Information Excerpt: RoHS Part Number 36
Table 2 Ordering Information Excerpt: Non RoHS Part Number. 37

Installation

Table 3 Accessories: RoHS compliant 43
Table 4 Accessories:Non RoHS compliant 43
Table 5 Environmental Requirements. 44
Table 6 Power Requirements 46
Table 7 Switch settings 49
Table 8 IPMI Mode Selection 50

Controls, Indicators and Connectors

Table 9 Description of IO Interfaces, Switches and LEDs. 64
Table 10 Description of Front Panel LEDs 65

Devices' features and Data Paths

Daughter Card Details

Maps and Registers

Table 11 PCI Memory Map 97
Table 12 SlowPort Map. 98
Table 13 NPU I2C Memory Map 101
Table 14 SENSE I2C Memory Map. 102
Table 15 BIB I2C Memory Map. 102
Table 16 Variant ID 103
Table 17 General Purpose EPLD-Version Number. 103
Table 18 General Purpose EPLD-Board Revision Number 104
Table 19 User Flash Select. 104
Table 20 CompactPCI Register. 104
Table 21 PrPMC Register 105
Table 22 System Specific Register 106
Table 23 User LED Control Register. 106
Table 24 Watchdog Timer Register. 107
Table 25 Watchdog Timer Interrupt Clear Register. 107
Table 26 Watchdog Enable and Watchdog Reset Enable Register 107
Table 27 Watchdog Time Set Register 108
Table 28 Interrupt Mask Register-1. 108
Table 29 Interrupt Mask Register-2. 109
Table 30 Interrupt Mask Register-3. 110
Table 31 Interrupt Status Register-1 111
Table 32 Interrupt Status Register-2 111

Table 33 Interrupt Status Register-3 112
Table 34 I2C MUX Reset Register 112
Table 35 MAC EPLD - IO-920-Variant ID 113
Table 36 PHY Reset Control Register 114
Table 37 MAC PAUSE Address Register 115
Table 38 MAC PAUSE Control Register 115
Table 39 Interrupt Mask Register 116
Table 40 Interrupt Status Register 116

RedBoot

Diagnostics

Troubleshooting

Introduction

Figure 1 Single Slot Complex Functional Block Diagram change to Single NPU 33

Installation

Figure 2 Location of Switches 49
Figure 3 Inserting the Bezel of the PMC into the Cutout on the Front Panel 51
Figure 4 Engaging the Connectors 51
Figure 5 Securing the PMC using Screws provided 52
Figure 6 IXP/CPCI-9120 with the installed PMC 52
Figure 7 Top and bottom Injector/Ejector Handles in the outward position 55
Figure 8 Installing Board -- Slide Board into Chassis 56
Figure 9 Location of Screw on Injector/Ejector Handle 56

Controls, Indicators and Connectors

Figure 10 Front Panel of IXP/CPCI-9120 64
Figure 11 D Subminiature Connector Pinout 66
Figure 12 J3 Connector Pinout (rows A-E) 67
Figure 13 J5 Connector Pinout (rows A-E) 68
Figure 14 Jn4 Connector Pinout 69
Figure 15 MSF Mezzanine Connector Pinout (rows A-C). 70
Figure 16 MSF Mezzanine Connector Pinout (rows D-E) 71
Figure 17 MSF Mezzanine Connector Pinout (rows F-H) 72
Figure 18 MSF Mezzanine Connector Pinout (rows J-K) 73
Figure 19 MSF Mezzanine Connector Pinout (MDI, SlowPort, I2C and JTAG) rows A-C 74
Figure 20 MSF Mezzanine Connector Pinout (MDI, SlowPort, I2C and JTAG) rows D-E 75
Figure 21 MSF Mezzanine Connector Pinout (MDI, SlowPort, I2C and JTAG) rows F-H 76
Figure 22 MSF Mezzanine Connector Pinout (MDI, SlowPort, I2C and JTAG) rows J-K 77
Figure 23 LA-1 Pinout rows A-C 78
Figure 24 LA-1 Pinout rows D-E 78
Figure 25 LA-1 Pinout rows F-H 79
Figure 26 LA-1 Pinout rows J-K 79

Devices' features and Data Paths

Figure 27 Single Slot Complex Functional Block Diagram 83

Daughter Card Details

Maps and Registers

Figure 28 MAC EPLD – IO-920-Version Number 113
Figure 29 Board revision Number 114
Figure 30 Mux Control Register 114

RedBoot

Diagnostics

Troubleshooting

Using this Guide

This Reference Guide is intended for users qualified in electronics or electrical engineering. Users must have a working understanding of Peripheral Component Interconnect (PCI), Compact Peripheral Component Interconnect (CPCI), and telecommunications.

Organization of this Manual

The Reference Guide is organized as follows.

Table 1: *Organization of this Manual*

Chapter	Description
Using this Guide	Lists all conventions and abbreviations used in this manual and outlines the revision history
Other Sources of Information	Lists related documentation and specifications
Safety Notes	Provides safety relevant information when handling the product
Sicherheitshinweise	German translation of the Safety Notes section
Introduction	Provides a basic overview of the features of the product and this manual
Installation	Outlines the installation requirements, hardware accessories, switch settings, installation and removal procedures
Controls, Indicators and Connectors	Describes the LEDs, keys, and connectors of the product
Devices' features and Data Paths	Provides detailled information on the devices, such as controllers, CPU etc., used on the blade and how they are interconnected
Daughter Card Details	IXP/CPCI-9120 has two daughter cards, one of which provides I/O and the other provides additional SRAM.
Maps and Registers	Provides information that is relevant for programmers, such as register reference and memory mapsa
RedBoot	Provides the details of the RedBoot firmware that runs on the IXP 2400 processor on the IXP/CPCI-9120 board and details the commands that are supported by the firmware.
Diagnostics	Provides detailed testing of various hardware interfaces and modules on the IXP/CPCI-9120.

Conventions

Notation	Description
57	All numbers are decimal numbers except when used with the notations described below
00000000_{16}	Typical notation for hexadecimal numbers (digits are 0 through F), e.g. used for addresses and offsets
0000_2	Same for binary numbers (digits are 0 and 1)
x	Generic use of a letter
n	Generic use of numbers
0.75	Decimal number
Bold	Used to emphasize a word
`Courier`	Used for on-screen output
`Courier+Bold`	Used to characterize user input and to separate it from system output
Italics	For references, table, and figure descriptions
File > Exit	Notation for selecting a submenu
<text>	Notation for variables and keys
[text]	Notation for software buttons to click on the screen and parameter descriptions
...	Repeated item, e.g. node 1, node 2, ..., node 12
. . .	Omission of information from example/command that is not necessary at the time being
..	Ranges, e.g.: 0..4 means one of the integers 0, 1, 2, 3, and 4 (used in registers)
|	Logical OR

Notation	Description
Note:	No danger encountered. Pay attention to important information marked using this layout
Caution 	Possibly dangerous situation: slight injuries to people or damage to objects possible
Danger 	Dangerous situation: injuries to people or severe damage to objects possible

Abbreviations

BIB	Board Information Block
BMC	Baseboard Management Controller
CPCI	Compact Peripheral Component Interconnect
CPSB	CompactPCI Packet Switching Backplane
DDR	Double Data Rate
DRAM	Dynamic Random Access Memory
EEPROM/E^{2}PROM	Electrically Erasable Programmable Read Only Memory
EPLD	Erasable Programmable-Logic Device
ESD	Electrostatic Discharge
FRU	Field Replaceable Unit
GPIO	General Purpose Input/output
I2C	Inter IC Communication
I/O	Input/Output
IBMU	Intelligent Board Management Unit
ICMB	Intelligent Chassis Management Bus
IEEE	Institute of Electrical and Electronic Engineers
IPMB	Intelligent Platform Management Bus
IPMI	Intelligent Platform Management Interface

IRQ	Interrupt Request
IXP	Internet Exchange Processor
JTAG	Joint Test Actions Group
KCS	Keyboard Controller Style
LA-1	Look Aside Phase 1
LFM	Linear Feet per Minute
LVTTL	Low Voltage Transistor-Transistor Logic
MAC	Media Access Controller
MII	Media Independent Interface
MMSC	Microprocessor and Microcomputer Standards Committee
MSF	Media Switch Fabric
N/A	Not Applicable
NIC	Network Interface Card
NPU	Network Processor Unit
OC-n	Optical Carrier level-n
PCB	Printed Circuit Board
PCI	Peripheral Component Interconnect
PCISIG	PCI Special Interest Group
PHY	Physical Layer Device
PICMG	PCI Industrial Computer Manufacturers Group
PM	Peripheral Management (controller)
PMC	PCI Mezzanine Card
PPMC/PrMC	Processor PCI Mezzanine Card
POS	Packet Over Sonet
POS-PHY	Packet Over Sonet PHYsical Layer device
PSB	Packet Switching Backplane
POSIX	Portable Operating Systems Interface
QDR	Quad Data Rate
RFU	Reserved for Future Use
RGMII	Reduced Gigabit Media Independent Interface

RTB	Rear Transition Board
SDK	Software Development Kit
SDRAM	Synchronous Dynamic Random Access Memory
SFP	Small Form Pluggable
SONET	Synchronous Optical Network
SPD	Serial Presence Detect
SPI3	System Packet Interface Level 3
SRAM	Static Random Access Memory
SREC	S Record
SSRAM	Synchronous Static Random Access Memory
SSTL_2	Stub Series Terminated logic for 2.5 Volts
TBD	To Be Decided
UART	Universal Asynchronous Receiver-Transmitter
VITA	VMEbus International Trade Association

Note: SPI-3 is the name associated with POS-PHY Level 3.

Revision History

Order No.	Revision	Date	Description
224535 420 000	AA	September 2004	Initial Release
224535 420 000	AB	December 2004	Updated Table of PCI Memory Map in Maps and Registers chapter Changes in Daughter Card Details chapter Three commands added to Diagnostics chapter
224535 420 000	AC	August 2005	Added commands in RedBoot and Diagnostics chapters Additions to registers, changes in pinouts and switch settings.

Order No.	Revision	Date	Description
6806800A77	A	June 2006	Updated with Motorola logo; formatted RedBoot commands; change and update in Part Numbers in Ordering Information and Accessories; name change to IXP/CPCI-9120; RoHS compliance added to Standards table; block diagrams updated. This manual has the new Part Number.
6806800A77	B	October 2006	Updated RoHS status of the product

Feedback

We welcome and appreciate your comments on our documentation. We want to know what you think about our manuals and how we can make them better.
Mail comments to:

Motorola GmbH
ECC Embedded Communications Computing
Lilienthalstr. 15
85579 Neubiberg-Munich/Germany
Email: reader-comments@mcg.mot.com

In all your correspondence, please list your name, position, and company. Be sure to include the title, part number, and revision of the manual and tell how you used it.

Other Sources of Information

For further information refer to the following documents:

Company	www.	Document
Altera	altera.com	MAX 7000A programmable Logic Device Family
Altera	altera.com	MAX 3000A programmable Logic Device Family
Intel	intel.com	Intel® IXF1104 4-Port Gigabit EthernetMedia Access Controller Revision Number: 006 Revision Date: August 21, 2003
Intel	intel.com	3 Volt Intel StrataFlash® Memory
Intel	intel.com	82546GB Dual Port Gigabit Ethernet Controller Datasheet, Revision 1.2 November 2003
Intel	intel.com	Intel® IXP2400 Network Processor datasheet, April 2003
Intel	intel.com	Intel® IXP2400 Network Processor Hardware Reference Manual April 2003
Linear Technology	linear-tech.com	LTC3728, Dual, 550kHz, 2-Phase Synchronous Step-Down Switching Regulator
Linear Technology	linear-tech.com	LTC3832/LTC3832-1, High Power Step-Down synchronous DC/DC controllers for low voltage operation
Linear Technology	linear-tech.com	LTC1649, 3.3V Input High Power Step-Down Switching Regulator Controller
Marvell	marvell.com	88E1141/88E1145 Datasheet MV-S100707-00, Rev.C, November 12, 2003
PCISIG	pcisig.com	PCI Local Bus Specification 2.2
PICMG	picmg.com	PICMG 2.0 R3.0
PICMG	picmg.com	PICMG 2.1 R2.0
PICMG	picmg.com	PICMG 2.16 R1.0
PICMG	picmg.com	PICMG 2.9 R1.0
Summit Micro-electronics	summitmicro.com/home/	SMH4042
Texas Instruments	ti.com	TPS54672

Company	www.	Document
VITA Standards Organisation (VSO)	vita.com	Processor PMC Standard, VITA 32 Draft 1.0a
Vitesse	vitesse.com	VSC215 Baseboard Management Controller Data Manual, Version 0.56, JAN/11/2001

Safety Notes

This section provides safety precautions to follow when installing, operating, and maintaining the IXP/CPCI-9120 and all sub-boards.

We intend to provide all necessary information to install and handle the IXP/CPCI-9120 in this Reference Guide. However, as the product is complex and its usage manifold, we do not guarantee that the given information is complete. If you need additional information, ask your representative.

The IXP/CPCI-9120 has been designed to meet the standard industrial safety requirements. It must not be used except in its specific area of office telecommunication industry and industrial control.

Only personnel trained by Motorola ECC or persons qualified in electronics or electrical engineering are authorized to install, maintain, and operate the IXP/CPCI-9120. The information given in this manual is meant to complete the knowledge of a specialist and must not be taken as replacement for qualified personnel.

EMC

The board has been tested in a standard Motorola ECC system and found to comply with the limits for a Class A digital device in this system, pursuant to part 15 of the FCC Rules respectively EN 55022 Class A. These limits are designed to provide reasonable protection against harmful interference when the system is operated in a commercial, business or industrial environment.

The board generates and uses radio frequency energy and, if not installed properly and used in accordance with this Reference Guide, may cause harmful interference to radio communications. Operating the system in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

If you use the board without a PMC module, cover empty slots with blind panels to ensure proper EMC shielding. If boards are integrated into open systems, always cover empty slots.

Installation

Electrostatic discharge and incorrect board installation and removal can damage circuits or shorten their life. Therefore:

- **Before installing or removing the board, read see “Action Plan” page -41.**
- **Before touching boards or electronic components, make sure that you are working in an ESD-safe environment.**
- **When plugging the board in or removing it, do not press on the front panel but use the handles.**
- **Before installing or removing an additional device or module, read the respective documentation.**
- **Make sure that the board is connected to the CompactPCI backplane via all assembled connectors and that power is available on all power pins.**

Hot Swap

The IXP/CPCI-9120 provides hot-swap support, i.e. it may be installed in or removed from a powered system supporting hot swap. Never install the board in or remove it from a system under hot-swap conditions unless the basic hot-swap, full hot-swap or high-availability platform is used and the system documentation explicitly includes appropriate guidelines. For detailed information on hot-swap support and the relevant safety notes, see “Installation in a Powered System Supporting Hot Swap” page -57”.

Only install the board under hot-swap conditions if the board is to be operated in slots for which hot swap is explicitly permitted by the system documentation (in most hot-swap systems peripheral slots only).

Operation

While operating the board ensure that the environmental and power requirements are met.

Do not operate the product outside the specified environmental limits. High humidity and condensation may cause short circuits. Make sure the product is completely dry and there is no moisture on any surface before applying power. Do not operate the product below 0°C.

When operating the board in areas of electromagnetic radiation ensure that the board is bolted on the CompactPCI system and the system is shielded by enclosure.

Make sure that contacts and cables of the board cannot be touched while the board is operating.

Replacement/Expansion

Only replace or expand components or system parts with those recommended by Motorola ECC. Otherwise, you are fully responsible for the impact on EMC and the possibly changed functionality of the product.

Check the total power consumption of all components installed (see the technical specification of the respective components). Ensure that any individual output current of any source stays within its acceptable limits (see the technical specification of the respective source).

RJ-45 Connector

The Board provides several RJ-45 connectors which commonly serve as different interfaces (RS-485, twisted pair Ethernet and telephone). Connecting different interfaces (e.g. Ethernet and RS-485) may damage the board. Therefore, make sure that you only connect matching interfaces. Furthermore, take note of the following:

- **Clearly mark TPE (Twisted Pair Ethernet) connectors near your working area as network connectors.**
- **TPE bushing of the system has to be connected only to safety extra low voltages (SELV) circuits.**
- **The length of the electric cable connected to a TPE bushing must not exceed 100 meter.**

If you have further questions, ask your system administrator.

Environment

Always dispose of old boards according to your country's legislation, if possible in an environmentally acceptable way.

Sicherheitshinweise

Dieser Abschnitt enthält Sicherheitshinweise, die beim Einbau, Betrieb und bei der Wartung des IXP/CPCI-9120 und alle dazugehörigen Boards zu beachten sind.

Wir sind darauf bedacht, alle notwendigen Informationen, die für die Installation und den Betrieb erforderlich sind, in diesem Handbuch bereit zu stellen. Da es sich jedoch um ein komplexes Produkt handelt bzw. viele verschiedene Einsatzmöglichkeiten bestehen, können wir die Vollständigkeit der im Handbuch enthaltenen Informationen nicht garantieren. Falls Sie weitere Informationen benötigen sollten, wenden Sie sich bitte an die für Sie zuständige Geschäftstelle von Motorola ECC.

Das IXP/CPCI-9120 erfüllt die für die Industrie geforderten Sicherheitsvorschriften und darf ausschließlich für Anwendungen in der Telekommunikationsindustrie und im Zusammenhang mit Industriesteuerungen verwendet werden.

Einbau, Wartung und Betrieb dürfen nur von durch Motorola ECC ausgebildetem oder im Bereich Elektronik oder Elektrotechnik qualifiziertem Personal durchgeführt werden. Die in diesem Handbuch enthaltenen Informationen dienen ausschließlich dazu, das Wissen von Fachpersonal zu ergänzen, können es aber in keinem Fall ersetzen.

EMV

Das Board wurde entsprechend der PCI Spezifikation in einem Motorola ECC Standardsystem getestet. Es erfüllt die für digitale Geräte der Klasse A gültigen Grenzwerte in einem solchen System gemäß den FCC-Richtlinien Abschnitt 15 bzw. EN 55022 Klasse A. Diese Grenzwerte sollen einen angemessenen Schutz vor Störstrahlung beim Betrieb des Boards in Gewerbe- sowie Industriegebieten gewährleisten.

Das Board arbeitet im Hochfrequenzbereich und erzeugt Störstrahlung. Bei unsachgemäßem Einbau und anderem als in diesem Handbuch beschriebenen Betrieb können Störungen im Hochfrequenzbereich auftreten.

Warnung!
Dies ist eine Einrichtung der Klasse A. Diese Einrichtung kann im Wohnbereich Funkstörungen verursachen. In diesem Fall kann vom Betreiber verlangt werden, angemessene Maßnahmen durchzuführen.

Wenn Sie das Board ohne PMC Modul verwenden, schirmen Sie freie Steckplätze mit einer Blende ab, um einen ausreichenden EMV Schutz zu gewährleisten. Wenn Sie Boards in Systeme einbauen, schirmen Sie freie Steckplätze mit einer Blende ab.

Installation

Elektrostatische Entladung und unsachgemäßer Ein- und Ausbau des Boards kann Schaltkreise beschädigen oder ihre Lebensdauer verkürzen. Beachten Sie deshalb die folgenden Punkte:

- **Lesen sie vor Ein- oder Ausbau des Boards den Abschnitt “Action Plan” auf Seite 2-41"**
- **Bevor Sie Boards oder elektronische Komponenten berühren, vergewissern Sie sich, dass Sie in einem ESD-geschützten Bereich arbeiten.**
- **Drücken Sie bei Ein- oder Ausbau des Boards nicht auf die Frontplatte, sondern benutzen Sie die Griffe.**
- **Lesen Sie vor dem Ein- oder Ausbau von zusätzlichen Geräten oder Modulen das dazugehörige Benutzerhandbuch.**
- **Vergewissern Sie sich, dass das Board über alle Stecker an die CompactPCI Backplane angeschlossen ist und Spannung an allen Versorgungskontakten anliegt.**

Hot Swap

Ein- oder Ausbau des Boards **in einem System unter Hot Swap Bedingungen darf nur dann stattfinden, wenn die Basic Hot Swap Plattform, die Full Hot Swap Plattform, oder die High-Availability Hot Swap Plattform verwendet wird und die Systembeschreibung dies ausdrücklich erlaubt. Im Abschnitt “Installation in a Powered System Supporting Hot Swap” auf Seite 2- finden Sie ausführliche Informationen zur Unterstützung von Hot Swap und die entsprechenden Sicherheitshinweise.**

Bauen Sie das Board nur dann unter Hot Swap Bedingungen ein, wenn das Board in Steckplätzen betrieben werden soll, die laut Systembeschreibung ausdrücklich für Hot Swap vorgesehen sind (in den meisten Hot Swap Systemen handelt es sich dabei um periphäre Steckplätze).

Betrieb

Achten Sie darauf, dass die Umgebungs- und die Leistungsanforderungen während des Betriebs eingehalten werden.

Betreiben Sie das Produkt nur innerhalb der angegebenen Grenzwerte für die relative Luftfeuchtigkeit und Temperatur, da durch hohe Luftfeuchtigkeit Kurzschlüsse entstehen können. Stellen Sie vor dem Einschalten des Stroms sicher, dass sich auf dem Produkt kein Kondensat befindet und betreiben Sie das Produkt nicht unter 0°C.

Wenn Sie das Board in Gebieten mit elektromagnetischer Strahlung betreiben, stellen Sie sicher, dass das Board mit dem CompactPCI System verschraubt ist und das System durch ein Gehäuse abgeschirmt wird.

Stellen Sie sicher, dass Anschlüsse und Kabel des Boards während des Betriebs nicht berührt werden können.

Austausch/Erweiterung

Verwenden Sie bei Austausch oder Erweiterung nur von Motorola ECC empfohlene Komponenten und Systemteile. Andernfalls sind Sie für mögliche Auswirkungen auf EMV und geänderte Funktionalität des Produktes voll verantwortlich.

Überprüfen Sie die gesamte aufgenomme Leistung aller eingebauten Komponenten (siehe die technischen Daten der entsprechenden Komponente). Stellen Sie sicher, dass die Stromaufnahme jedes Verbrauchers innerhalb der zulässigen Grenzwerte liegt (siehe die technischen Daten des entsprechenden Verbrauchers).

RJ-45 Stecker

Das CPU Board ist mit RJ-45 Steckern ausgestattet. Dieser Stecker wird sowohl für Telefonanschlüsse als auch für Netzwerkkabel (Twisted Pair Ethernet - TPE) verwendet. Die Verwechslung dieser Anschlüsse kann sowohl das Telefon als auch das Board zerstören. Beachten Sie deshalb folgende Punkte:

- **Vergewissern Sie sich, dass Anschlüsse deutlich als Netzwerkanschlüsse gekennzeichnet sind.**
- **Schließen Sie TPE-Stecker/Netzwerkstecker Ihres Systems nur an Sicherheits-Kleinspannungs-Kreise (SELV) an.**
- **Vergewissern Sie sich, dass die an einem TPE-Anschluss angeschlossene Leitung eine Gesamtlänge von 100 Metern nicht überschreitet.**

Falls Sie Fragen haben, wenden Sie sich an Ihren Systemadministrator.

Umweltschutz

Entsorgen Sie alte Boards gemäß der in Ihrem Land gültigen Gesetzgebung, wenn möglich immer umweltfreundlich.

1

Introduction

Features

IXP/CPCI-9120 is a 6U 4HP CompactPCI Single Board Computer complex based on the Intel IXP2400 network processor for up to duplex OC-48 or 4 Gbit Ethernet bandwidth applications. IXP/CPCI-9120 has a Processor PMC expansion for expandable Control plane processing power, a flexible media interface mezzanine to support different standard and OEM media flavors (e.g., Copper Ethernet, Fiber Ethernet, SONET/ATM, DS1/E1, etc.), and a SRAM expansion mezzanine site for memory expansion or co-processing support.
The IXP/CPCI-9120 complies with the following industry standard specifications:

- PICMG 2.0 R3.0 (CompactPCI Core Specifications)
- PICMG 2.1 R2.0 (Hot Swap Specification)
- PICMG 2.16 R1.0 (Packet Switching Backplane Specification)
- PICMG 2.9 R1.0 (System Management Specification)
- VITA 32 (Processor PMC)

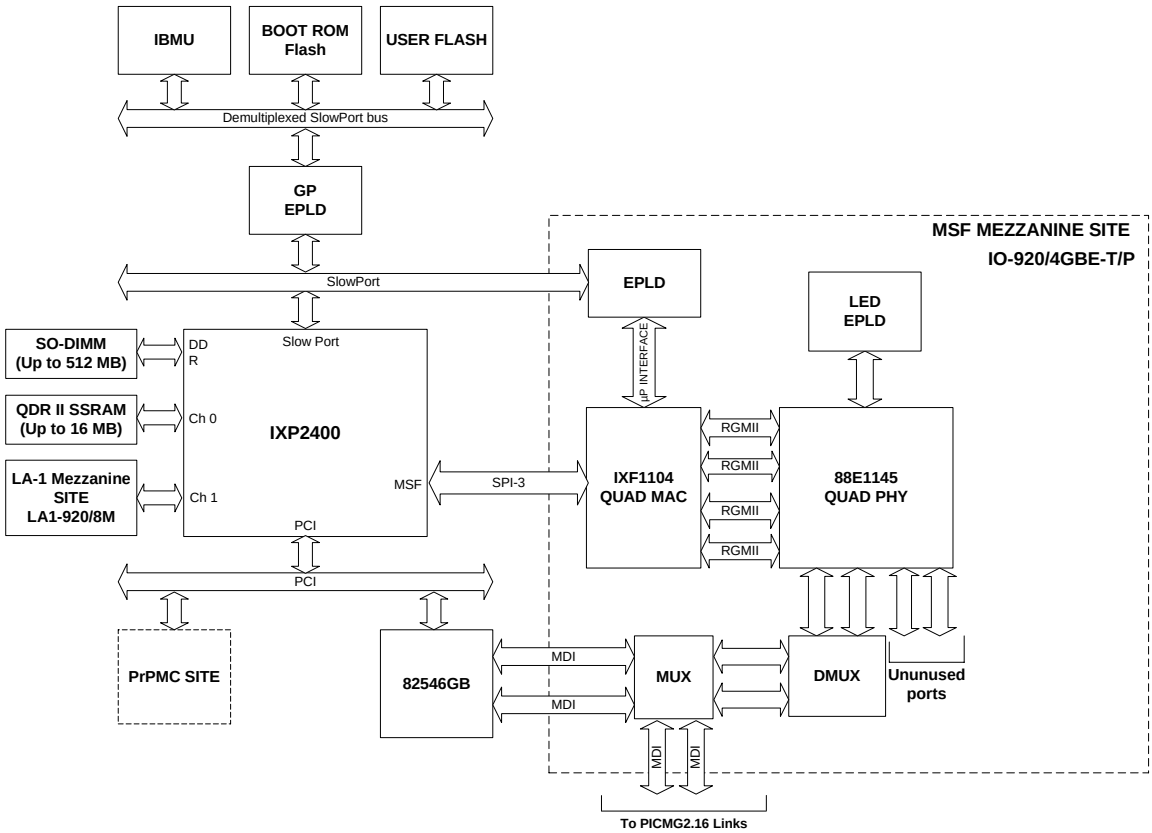


Figure 1: *Single Slot Complex Functional Block Diagram change to Single NPU*

The 2.16 variant provides two Gigabit Ethernet ports on the PICMG 2.16 links.

IXP/CPCI-9120 single slot board complex comprises of the following features:

- Single-NPU Intel IXP2400 board at 600 MHz
- 256 MB SO-DIMM DDRSDRAM with ECC
- 8 MB QDRSRAM with ECC onboard
- 8 MB QDRSRAM with ECC on accessory LA-1 mezzanine card
- 4 MB Boot Flash onboard
- 16 MB User Flash onboard
- 2 x PCI GbE MAC or 2 x SPI-3 GbE MAC on MSF mezzanine routed to PICMG 2.16 links.

Standard Compliance

The IXP/CPCI-9120 and all sub-boards meet the following standards:

Standard	Description
EN 60950-1 UL 60950-1 CSA 60950-1 IEC 60950-1	Legal safety requirements
EN 55022 EN 55024 EN 300 386 FCC part 15 CISPR 22	EMC requirements on system level (in a predefined Motorola system)

Note: The product has been designed to meet the directive on the restriction of the use of certain hazardous substances in electrical and electronic equipment (RoHS) Directive 2002/95/EC.

Ordering Information

When ordering the IXP/CPCI-9120 board variants, upgrades and accessories, use the order numbers given below.

Product Nomenclature

In the following you find the key for the product name extensions.

IXP/CPCI-9120/xxx/ccc/Lyyy/zz	
xxx	SO-DIMM DDRSDRAM size in Mbyte
ccc	QDRSRAM size in Mbyte
Lyyy	QDRSRAM size in Mbyte on LA-1 mezzanine card
zz	User Flash size in Mbyte

Order Numbers

Depending on the IXP/CPCI-9120 type, the available upgrades and accesories may differ. Consult your local sales representative to check the possibility of combinations.

Table 1: *Ordering Information Excerpt: RoHS Part Number*

Order No.	IXP/CPCI-9120	Description
122754	IXP/CPCI-9120/256/8-8/16/2GBE-2.16/6E	Single NPU solution with 2x 1000 Base- T PICMG 2.16 Mezzanine
Hardware Accessories IXP/CPCI-9120		
122753	ACC/Serial/Cable/MD9/RD9/2 5E	MD9 to DB9 Serial cable

Note: The Serial cable is used only for debug purpose.

Table 2: *Ordering Information Excerpt: Non RoHS Part Number*

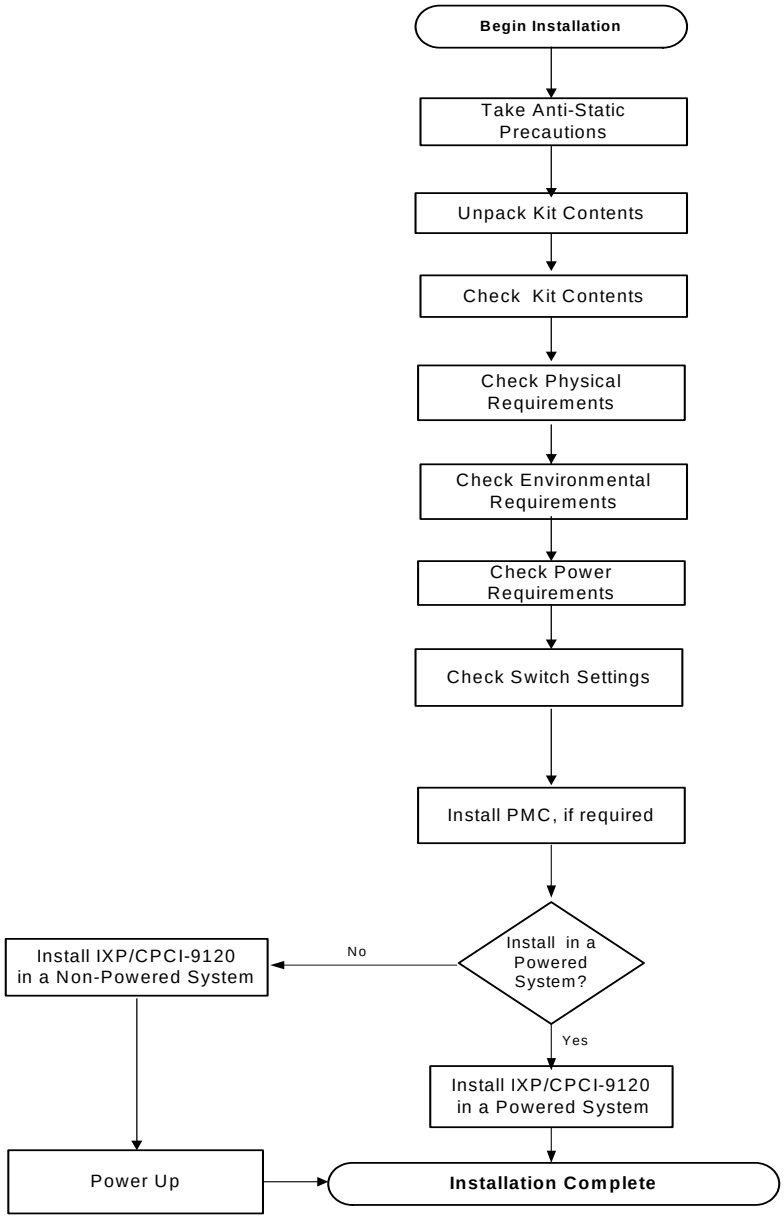
Order No.	IXP/CPCI-9120	Description
Hardware Accessories IXP/CPCI-9120		
111013	ACC/Serial/Cable/MD9/RD9/2	MD9 to DB9 Serial cable **Note: The Serial cable is used only for debug purpose.**
120460	ACC/Serial Cable DTE	
214451	CABLE, 9 Rnd Shielded	

2

Installation

Action Plan

To install the board, the following steps are necessary and described in detail in the sections of this chapter.



Preparing for Installation

Prepare for installation by following these steps:

- Take Antistatic Precautions
- Unpacking Kit Contents
- Environmental Requirements
- Power Requirements

Take Antistatic Precautions

When handling circuit boards and associated internal computer components, use an antistatic wristwrap or wear isolation gloves.

Caution **Circuit boards and associated system components are sensitive to and can be damaged by ESD. To avoid damaging boards or components, take appropriate precautions when handling them.**

In addition to using an antistatic wristwrap or wearing isolation gloves, consider the following precautions:

- Do not allow a circuit board or other components to make contact with nonconductors, including your clothing.
- Keep loose circuit boards inside or on top of conductive plastic bags.
- Before touching a loose circuit board or component, discharge static electricity.

Unpacking Kit Contents

To unpack kit contents:

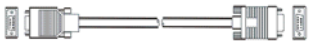
1. Locate and put on an antistatic wristwrap.
2. Clip the wire attached to the wristwrap to the frame of your system chassis.
3. Verify that the order numbers on labels of your kit carton match the numbers for the product you ordered. If the order numbers do not match, contact your local sales representative.
4. Remove all items from the carton.
5. Remove the hardware components from the conductive plastic bags in which they were packaged and place the components on the bags.

Note: Do not discard the original packaging material in case a factory return is necessary.

Accessories

The following table lists the accessories available for IXP/CPCI-9120.

Table 3: *Accessories: RoHS compliant*

	ACC/Serial/Cable/MD9/RD9/25E	122753
		
	ACC/Serial Port Y-Adapter/5E	122734

Note: The Serial cable is used only for debug purpose.

Table 4: *Accessories:Non RoHS compliant*

	ACC/Serial/Cable/MD9/RD9/2	111013
		
	ACC/Serial Port Y-Adapter	110973

Note: The Serial cable is used only for debug purpose.

Requirements

To meet the environmental requirements, the IXP/CPCI-9120 has to be tested in the system where it is to be installed. Before you power up the board, calculate the power needed according to your combination of board upgrades and accessories.

Environmental Requirements

The environmental values must be tested and proven in the used system configuration. The conditions listed below refer to the surroundings of the board within the user environment.

Note: Operating temperatures refer to the temperature of the air circulating around the board and not to the component temperature.

Danger



Do not operate the product outside the specified environmental limits. High humidity and condensation may cause short circuits. Make sure the product is completely dry and there is no moisture on any surface before applying power. Do not operate the product below 0°C.

Table 5: *Environmental Requirements*

Feature	Operating	Non-Operating
Temperature	0°C to +55°C	–40 °C to +85°C
Forced air flow	300 LFM	
Temp. change	± 0.5°C/min	± 1°C/min
Rel. humidity	5% to 95% non condensing at +40°C	5% to 95% non condensing at +40°C
Altitude	–300 m to +3,000 m	–300 m to +13,000 m
Vibration		
Sinusoidal	Level: 5-500Hz 1g Sweep rate:1 oct/min Duration:9 sweeps = 60 min/axis	
Random		Frequency range:10-2000Hz Level:0.0045 g2 /Hz Acceleration:3.03gRMS Duration: 60 min per axis

Table 5: *Environmental Requirements*

Feature	Operating	Non-Operating
Shock	Shape: Half sine Acceleration:5g Duration:2ms Number of shocks: 3 per direction (6 per axis) total of 18 shocks	Shape:half sine Accelaration:15g Duration: 11ms Number of shocks: 3 per direction (6 per axis) total of 18 shocks
Free Fall	100 mm/3 axis	1,200 mm /all edges and corners (packed state)

Power Requirements

The board's power requirements depend on the installed hardware accessories. The following table gives examples of maximum power requirements for 5V and 3.3V for a processor running at 600 MHz without any PMC card installed. If you want to install any accessories, the load of the respective accessory has to be added to the load of the used board variant.

For information on the accessories' power requirements, refer to the documentation coming with the respective accessory or ask your local representative.

Table 6: *Power Requirements*

IXP/CPCI-9120	*+3.3V*	*+5V*	*+12V*	*-12V*
Typical power requirements				
Min. voltage	+3.201V	+4.85	+11.4V	-12.6V
Max. voltage	+3.465V	+5.25V	+12.6V	-11.4V
Max. current	3.5 A	3.75 A	0.03 A	0.01 A

Hardware Upgrades and Accessories

The IXP/CPCI-9120 itself allows an easy and cost-efficient way to adapt the system board to your application needs.

IO Daughtercard

The MSF mezzanine connects to the MSF interface of the IXP2400. Several flavors are planned, thereby providing added expansion and flexibility.

IO-920

IO-920 MSF Mezzanine card is designed for use in Single Slot complex board. The IO-920 accommodates an Intel® IXF1104 Four Port Gigabit Ethernet Media Access Controller (Vallejo), a Marvell 88E1145 Quad Gigabit Ethernet Transceiver, Multiplexers and Demultiplexers.

The IO-920 also houses an I/O connector through which MDI lines from the Dual Gigabit Ethernet controller are brought into the IO-920. Through multiplexers and demultiplexers,

- Signals from two ports of the on-board Quad PHY could be routed to the PSB.
- Signals from the two ports of the baseboard Dual MAC (82546GB) could be routed to the PSB.

PMC Module

The CPCI-9120 provides one VITA32-compliant Processor PMC site on to which additional I/O or processing power may be added. The mounted Processor PMC may be configured in Monarch or Non-monarch mode.

Note: The PMC site on IXP/CPCI-9120 shall support 3V-only PrPMCs.

Note: The total power consumption (from +12V, -12V, +3.3V and +5V) of the PMC must not exceed 35 Watts. The power consumption from 5V and 3.3V shall not exceed 17.5Watts and 18W respectively.

Installation and Removal Procedure

For details on how to install and remove PMC, see “Installing PMC” and see “Removing PMC” page -53.

LA1-920/8M

LA1-920/8M is a QDR SRAM memory expansion mezzanine that is connected to Channel 1 of the IXP2400 memory controller. It contains 8 MB of QDR II SSRAM, and is expandable to 16 MB. This interface operates at 200 MHz.

Switch Settings

Single Throw Single pole switches are provided on the IXP/CPCI-9120 to

- Accord Monarch status to the PrPMC
- Force BMC or PM mode of operation for the IPMI controller
- Disable reset of the system from the Front panel
- Provide write-protection to BootROM Flash and User Flash

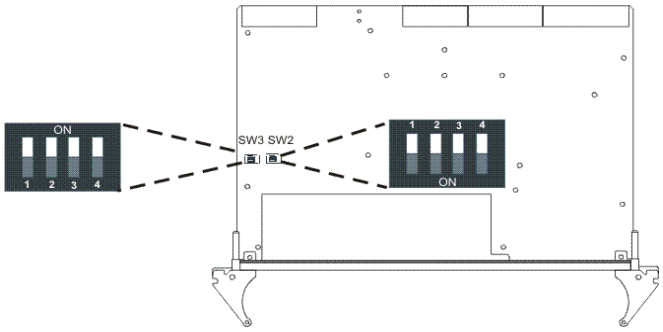


Figure 2: *Location of Switches*

The board is delivered with the DIP switches set to ON position for SW2 and OFF for SW3.

Table 7: *Switch settings*

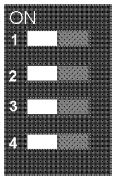
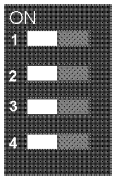
Switch	Position	Description
SW2		
	1	OFF: User Flash Write Disabled ON: User Flash Write Enabled
	2	OFF: BootROM Flash Write Disabled ON: BootROM Flash Write Enabled
	3	Reserved Reserved
	4	Reserved Reserved

Table 7: *Switch settings (cont.)*

Switch	Position	Description
SW3		
	1	OFF: Front Panel Reset enable
		ON: Front Panel Reset disable
	2	OFF: PrPMC Monarch disable
		ON: PrPMC Monarch enable
	3	OFF: Auto-detect (BMC or PM mode)
		ON: Force BMC mode
	4	OFF: Depends on Switch position 3
		ON: Force PM

Note: Positions 3 and 4 of SW3 are used together as indicated in Table 6.

Table 8: *IPMI Mode Selection*

Position 3	Position 4	Mode selected
OFF	OFF	Auto Detect
ON	OFF	Force BMC
OFF	ON	Force PM
ON	ON	Force PM

Installing PMC

To install a PMC, follow these steps:

1. Switch off power.
2. Remove IXP/CPCI-9120 from the system according to the removal procedure steps (see “Installation in a Non-Powered System” page -54).
3. Locate the PMC slot on IXP/CPCI-9120 into which you are installing the PMC.
4. Remove the PMC slot filler from front panel of IXP/CPCI-9120.

5. Insert the bezel of the PMC into the cutout on the front panel as shown in Figure 3.

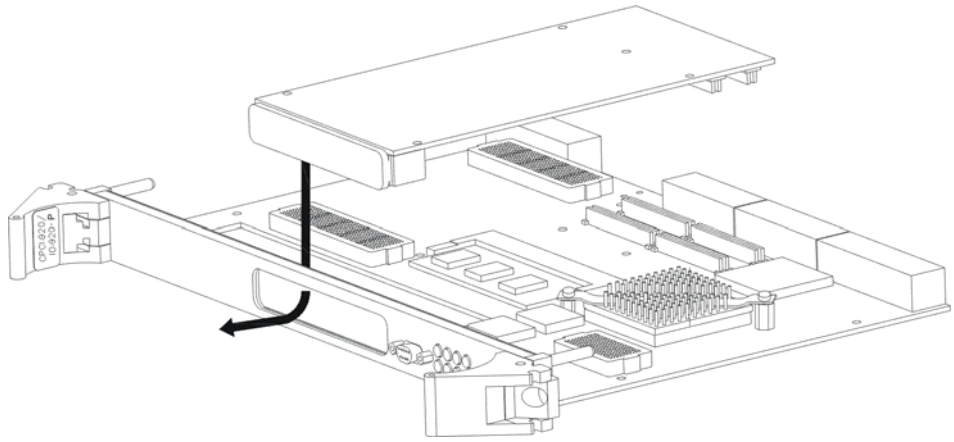


Figure 3: *Inserting the Bezel of the PMC into the Cutout on the Front Panel*

6. Engage the connectors as shown in Figure 4 making sure they are seated completely.

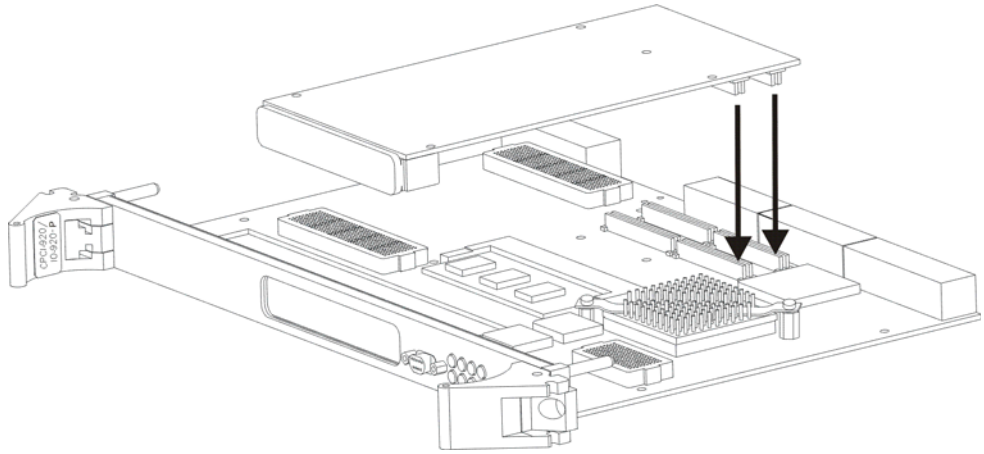


Figure 4: *Engaging the Connectors*

7. Secure the PMC using screws provided. Ensure that you screw the PMC from the rear side of using IXP/CPCI-9120 through clearance holes as in Figure 5.

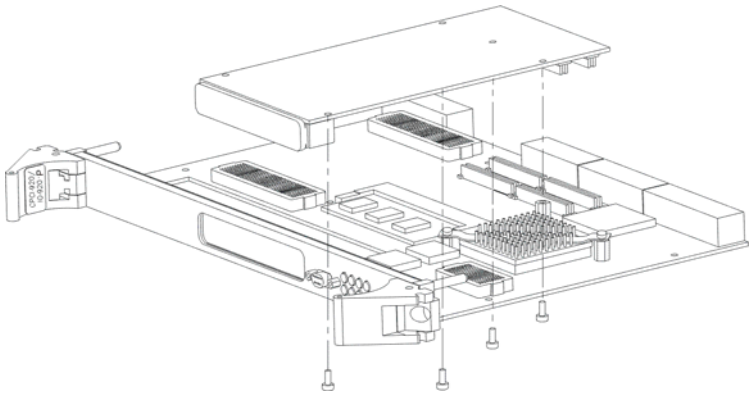


Figure 5: *Securing the PMC using Screws provided*

Installation of the PMC on IXP/CPCI-9120 is now complete. A view of the IXP/CPCI-9120 with the installed PMC is shown in Figure 6.

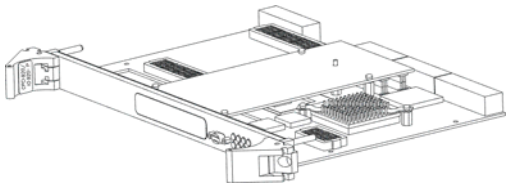


Figure 6: *IXP/CPCI-9120 with the installed PMC*

Note: Ensure that you cover the unused FrontPanel cutouts with filler panels.

Removing PMC

To remove the PMC from IXP/CPCI-9120 follow the steps detailed here:

1. Turn off system power.
2. Remove IXP/CPCI-9120 from the system as per the removal procedure detailed in section “Installation in a Non-Powered System”.
3. Carefully unscrew the PMC from the bottom of the IXP/CPCI-9120.
4. Carefully remove the PMC module from IXP/CPCI-9120 by disengaging PMC connectors.
5. Cover empty slot of IXP/CPCI-9120 with PMC filler slot.

IXP/CPCI-9120 Board Installation

The IXP/CPCI-9120 provides hot-swap support and hence may be installed in or removed from a powered system. This section details installation of IXP/CPCI-9120 in a non-powered system and in a powered system supporting hot-swap.

Installation in a Non-Powered System

Installation Procedure

In order to install the board in a non-powered system, proceed as follows:

Caution **Before touching integrated circuits, make sure you are working in an ESD-safe environment.**



1. Turn off system power.
2. Check switch settings. (See “Switch Settings”).
3. Insert board into the system by placing the top and bottom edges of the board in the card guides of the chassis as shown in Figure 7. Depending on the intended function, insert the board either into a System Slot or into a Peripheral Slot. A System Slot is marked with a triangle on the chassis cage, a Peripheral Slot is marked with a circle on the chassis cage.

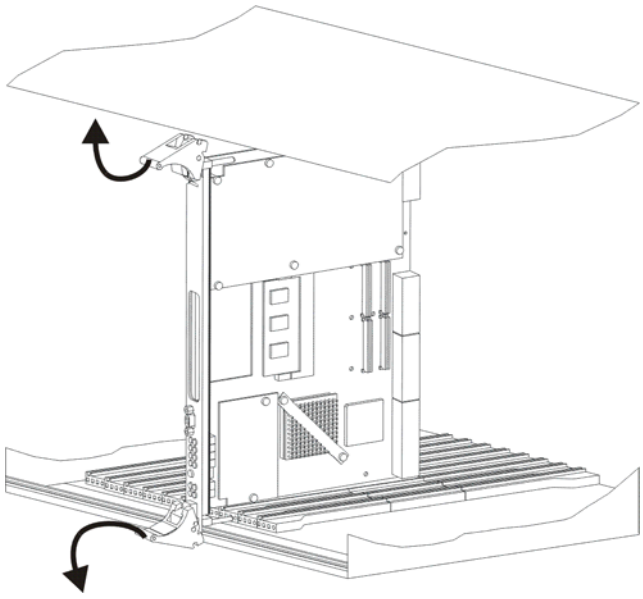


Figure 7: *Top and bottom Injector/Ejector Handles in the outward position*

4. Check that the top and bottom injector/ejector handles are in the outward position as shown by arrows in Figure 7.

5. Slide the board into the chassis until you feel resistance (approximately 1/4 inch short of full insertion). The arrow in Figure 8 shows you the direction of required movement.

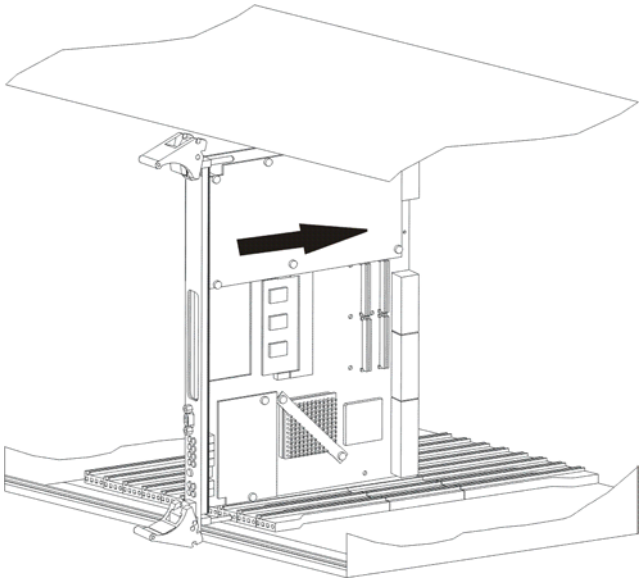


Figure 8: *Installing Board -- Slide Board into Chassis*

6. Simultaneously move the top and bottom injector/ejector handles to the inward position.
7. Verify that the board is seated properly.
8. Tighten the handle screws that secure the board to the chassis. A zoomed view of the screw on the injector/ejector handle is shown in Figure 9.

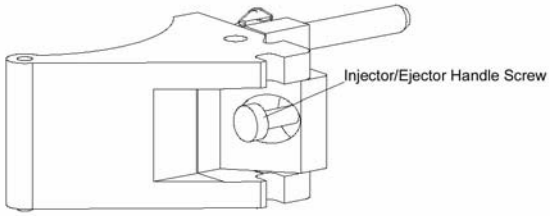


Figure 9: *Location of Screw on Injector/Ejector Handle*

Plug-in interface cables to the front panel connectors if applicable.

9. Turn on system power.

Removal Procedure

In order to remove a board from a non-powered system, proceed as follows:

1. Turn off system power.
2. Unfasten screws of front panel until board is detached from rack frame.
3. Press the red button on the ejector/injector handles.
4. Move the ejector/injector handles outwards to disconnect board from backplane.
5. Remove board from slot rails.

Installation in a Powered System Supporting Hot Swap

IXP/CPCI-9120 supports “full hot swap”.

The basic purpose of “hot-swap support” is to allow a board to be installed and removed in a powered system without adversely affecting system operation. With hot-swap support, defective boards can be repaired and systems can be reconfigured without disrupting system operation and with minimum operator intervention.

Caution 

Never hot-swap the IXP/CPCI-9120 unless a hot-swap or high availability platform is used and the documentation of all installed boards and the system documentation explicitly include appropriate guidelines. Live insertion/extraction of the IXP/CPCI-9120 into/from a System slot is not supported.

Control Elements

The board provides the following elements as man-machine interface:

- Hot-swap switch
- Blue hot-swap LED

Integrated in the lower injector/ejector handle of the front panel, the hot-swap switch allows you to indicate the intention to remove the board. The hot-swap switch is turned on when you move the lower injector/ejector handle inwards.

When the blue hot-swap LED glows, it indicates that you can safely remove the board.

Installation Procedure

Before installing IXP/CPCI-9120, observe the following:

Caution 

Only install the board under hot-swap conditions if the board is to be operated in slots for which hot swap is explicitly permitted by the system documentation (in most hot-swap systems, only Peripheral Slots are hot-swappable).

Before touching integrated circuits, make sure that you are working in an ESD-safe environment.

The installation process depends upon the system that the board is to be installed.

Installation in Basic Hot Swap System

1. Ensure that the board has the desired accessories and all switches are appropriately configured.
2. Ensure that IXP/CPCI-9120 is inserted into a slot that does not have an RTB already installed.
3. Insert board into powered system by placing the top and bottom edges of the board in the card guides of the chassis as shown in Figure 8.
4. Check that the top and bottom injector/ejector handles are in the outward position as shown by arrows in Figure 7.
5. Slide the board into the chassis until you feel resistance (approximately 1/4 inch short of full insertion). The arrow in Figure 8 shows you the direction of required movement.
6. Simultaneously move the top and bottom injector/ejector handles to the inward position.
7. Verify that the board is seated properly.
8. Tighten the captive screws that secure the board to the chassis. A zoomed view of the screw on the injector/ejector handle is shown in Figure 9.
9. Perform the software connection according to system documentation.

Installation in Full Hot-Swap System

In order to install the board in a full hot-swap system proceed as follows:

1. Ensure that the board has the desired accessories and that all switches are appropriately configured.
2. Ensure that IXP/CPCI-9120 is inserted into a slot that does not have an RTB already installed.
3. Insert board into powered system as per the steps 3 to 8 detailed in “Installation in Basic Hot Swap System”.

Removal Procedure

Before removing the board, observe the following:

Caution 

Never extract the IXP/CPCI-9120 unless a hot-swap or high availability platform is used and the documentation of all installed boards and the system documentation explicitly include appropriate guidelines. Live insertion/extraction of the IXP/CPCI-9120 into/from a System slot is not supported.

- **Before touching integrated circuits, make sure that you are working in an ESD-safe environment.**

Removal from Basic Hot-Swap System

To remove the board from a basic hot-swap system, proceed as follows:

1. Carry out software disconnection.
2. Wait until software disconnection process has been completed.
3. Press red buttons on ejector/injector handles to indicate board removal.
4. Wait till blue hot-swap LED is illuminated.
5. Move ejector/injector handles outwards.
6. Remove board from powered system.

Removal from Full Hot-Swap System

To remove the board from a full hot-swap system, proceed as follows:

1. Press red buttons on ejector/injector handles to indicate board removal.
2. Wait until blue hot-swap LED is illuminated.
3. Move ejector/injector handles outwards. Remove board from powered system.

3

Controls, Indicators and Connectors

Front Panel

An IEEE 1101.10-compliant front panel is used on the IXP/CPCI-9120. The front panels are equipped with an EMC gasket and two ejector/injector handles.

The front panel accommodates the following:

- Hot swap status LED (Blue LED) placed above the lower ejector.
- One bi-color LED for power status
- Two bi-color LEDs per gigabit ethernet port to indicate status and speed (total eight).
- Two general purpose/user-defined bi-color LEDs.
- Pinhole reset switch
- Serial port in a D-type subminiature connector (MD-9)
- Cutouts for one PMC Module.

The board is shipped with a filler panel attached to the PMC cutout.

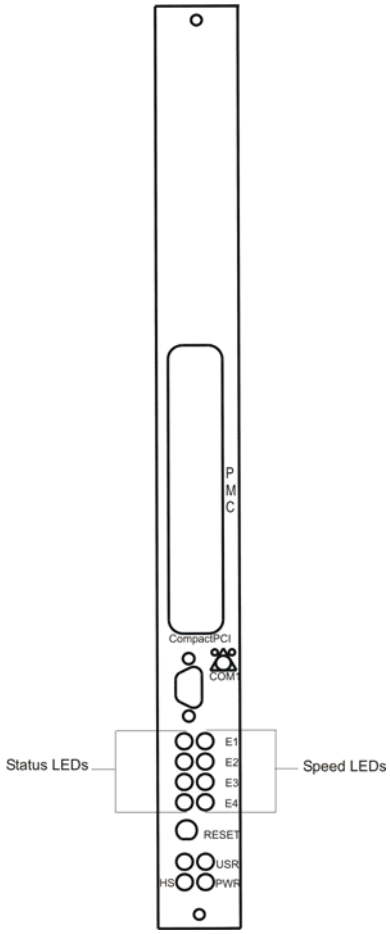


Figure 10: *Front Panel of IXP/CPCI-9120*

For *Description of IO Interfaces, Switches and LEDs on Front Panel,* see following page.

Table 9: *Description of IO Interfaces, Switches and LEDs*

PMC	Cut-out for PMC module
COM1	D-type subminiature D-type connector
E1	Ethernet Port1 Bi-color LED
E2	Ethernet Port2 Bi-color LED

Table 9: *Description of IO Interfaces, Switches and LEDs*

E3	Ethernet Port3 Bi-color LED
E4	Ethernet Port4 Bi-color LED
RESET	Reset Switch
USR	User defined Bi-color LED
HS	Hot swap status LED
PWR	Power Status Bi-color LED

LEDs

The following indicators are provided on the front panel of the IXP/CPCI-9120:

Table 10: *Description of Front Panel LEDs*

LED	Description
Hot Swap	
Blue LED	Hot Swap status LED. When illuminated, indicates that it is safe to extract the board from a live backplane.
Power	
Bi-color LED	Red: Power unstable Yellow: Power stable
User LEDs	
Bi-color LED1	User specific
Bi-color LED2	User specific
Ethernet Speed LEDs	
LED off	10 Mbps link
Blink Green	100 Mbps, half-duplex
Green	100 Mbps, Full-duplex
Yellow	1000 Mbps
Link Status LEDs	
LED off	Link not OK
Blink Green	Activity

Table 10: *Description of Front Panel LEDs*

LED	Description
Green	Link OK
Yellow	Collision

Keys

Front Panel Reset

A pinhole reset switch is provided on the front panel. Pressing this switch causes all devices on the IXP/CPCI-9120 to reset except IBMU. This reset will be active as long as the reset switch is held in the active position.

Resets from the front panel can be disabled by means of switch settings (see “Description of Front Panel LEDs” page -65).

Connectors

D-Type Subminiature Connector

A D-type subminiature connector is used to facilitate debug and software development. Modem controls are not supported on this connector.

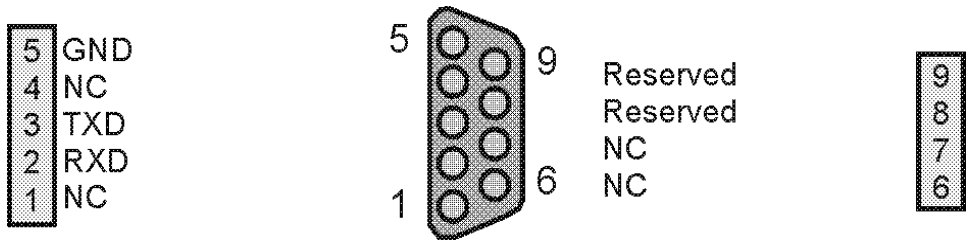


Figure 11: *D Subminiature Connector Pinout*

On-Board Connectors

IXP/CPCI-9120 has the following on-board connectors:

- CompactPCI connectors
- PrPMC connectors
- MSF mezanine connector
- LA-1 mezzanine connector

CompactPCI Connectors

CompactPCI connectors J1, J2, J3, and J5 are present.

J1 and J2

The J1 and J2 CompactPCI connectors implement the CompactPCI 64 bit connector pinouts as specified by the CompactPCI specification. These pinouts are hence not documented here.

J3

	A	B	C	A B C D E	D	E	
19	GND	GND	GND		GND	GND	19
18	PSB0_MDI[0]+	PSB0_MDI[0]-	GND		PSB0_MDI[2]+	PSB0_MDI[2]-	18
17	PSB0_MDI[1]+	PSB0_MDI[1]-	GND		PSB0_MDI[3]+	PSB0_MDI[3]-	17
16	PSB1_MDI[0]+	PSB1_MDI[0]-	GND		PSB1_MDI[2]+	PSB1_MDI[2]-	16
15	PSB1_MDI[1]+	PSB1_MDI[1]-	GND		PSB1_MDI[3]+	PSB1_MDI[3]-	15
14	GND	GND	GND		GND	GND	14
13	PORT2_SPEED+	PORT2_SPEED-	GND		NC	GP_RXD	13
12	GND	GND	GND		NC	GP_TXD	12
11	PORT1_STATUS+	PORT1_STATUS-	GND		NC	NC	11
10	GND	GND	NC		PORT0_STATUS-	GND	10
9	PORT1_SPEED+	PORT1_SPEED-	NC		GND	PORT0_STATUS+	9
8	GND	GND	NC		PORT0_SPEED-	GND	8
7	82546GB0_LED_LINK100#	82546GB1_LED_LINK100#	NC		GND	PORT0_SPEED+	7
6	GND	GND	MISC3		PORT3_STATUS-	GND	6
5	82546GB0_LED_LINK1000#	82546GB1_LED_LINK1000#	MISC2		GND	PORT3_STATUS+	5
4	GND	GND	MISC1		PORT3_SPEED-	GND	4
3	82546GB0_LED_LINK#	82546GB1_LED_LINK#	MISC0		GND	PORT3_SPEED+	3
2	GND	GND	PSB_RTB_SEL1		PORT2_STATUS-	GND	2
1	82546GB0_LED_ACTIVITY#	82546GB1_LED_ACTIVITY#	PSB_RTB_SEL0		GND	PORT2_STATUS+	1

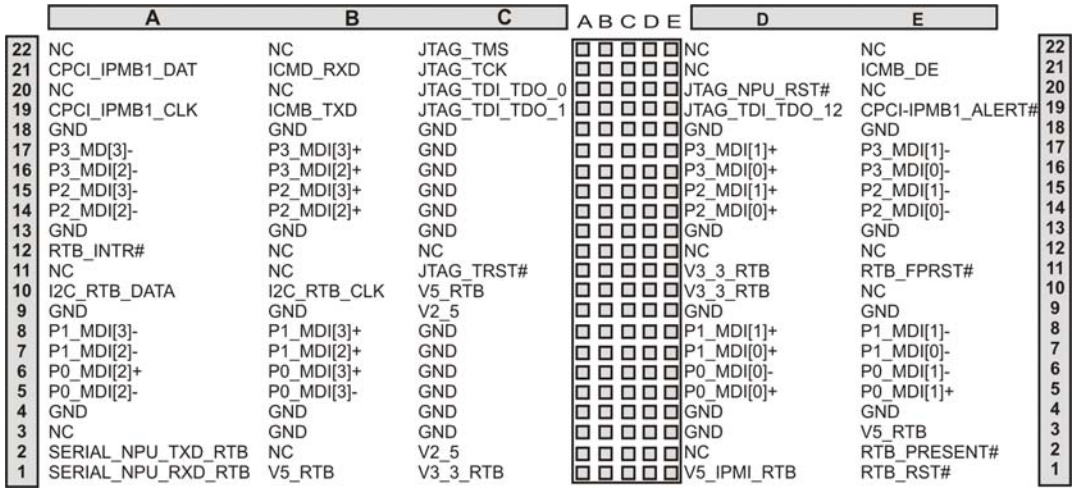
Figure 12: *J3 Connector Pinout (rows A-E)*

Note: All pins of Row F are taken to GND.

J4

Note: J4 connector is not mounted.

J5



	A	B	C	D	E	
22	NC	NC	JTAG_TMS	NC	NC	22
21	CPCI_IPMB1_DAT	ICMD_RXD	JTAG_TCK	NC	ICMB_DE	21
20	NC	NC	JTAG_TDI_TDO_0	JTAG_NPU_RST#	NC	20
19	CPCI_IPMB1_CLK	ICMB_TXD	JTAG_TDI_TDO_1	JTAG_TDI_TDO_12	CPCI-IPMB1_ALERT#	19
18	GND	GND	GND	GND	GND	18
17	P3_MD[3]-	P3_MDI[3]+	GND	P3_MDI[1]+	P3_MDI[1]-	17
16	P3_MDI[2]-	P3_MDI[2]+	GND	P3_MDI[0]+	P3_MDI[0]-	16
15	P2_MDI[3]-	P2_MDI[3]+	GND	P2_MDI[1]+	P2_MDI[1]-	15
14	P2_MDI[2]-	P2_MDI[2]+	GND	P2_MDI[0]+	P2_MDI[0]-	14
13	GND	GND	GND	GND	GND	13
12	RTB_INTR#	NC	NC	NC	NC	12
11	NC	NC	JTAG_TRST#	V3_3_RTB	RTB_FPRST#	11
10	I2C_RTB_DATA	I2C_RTB_CLK	V5_RTB	V3_3_RTB	NC	10
9	GND	GND	V2_5	GND	GND	9
8	P1_MDI[3]-	P1_MDI[3]+	GND	P1_MDI[1]+	P1_MDI[1]-	8
7	P1_MDI[2]-	P1_MDI[2]+	GND	P1_MDI[0]+	P1_MDI[0]-	7
6	P0_MDI[2]+	P0_MDI[3]+	GND	P0_MDI[0]-	P0_MDI[1]-	6
5	P0_MDI[2]-	P0_MDI[3]-	GND	P0_MDI[0]+	P0_MDI[1]+	5
4	GND	GND	GND	GND	GND	4
3	NC	GND	GND	GND	V5_RTB	3
2	SERIAL_NPU_TXD_RTB	NC	V2_5	NC	RTB_PRESENT#	2
1	SERIAL_NPU_RXD_RTB	V5_RTB	V3_3_RTB	V5_IPMI_RTB	RTB_RST#	1

A B C D E

Figure 13: *J5 Connector Pinout (rows A-E)*

Note: All pins of Row F are taken to GND.

PrPMC Connectors

PMC slot connectors include the following:

- Jn1 and Jn2 for the 32-bit PCI bus signals
- Jn3 for 64-bit PCI bus signals
- Jn4 for User I/O signals

Local bus signal pins on Jn1, Jn2 and Jn3 are assigned as specified by MMSC of IEEE in Draft Standard Physical and Environmental Layers for PCI Mezzanine Cards: PMC P1386.1/Draft 2.4. Therefore, these pinouts have not been documented.

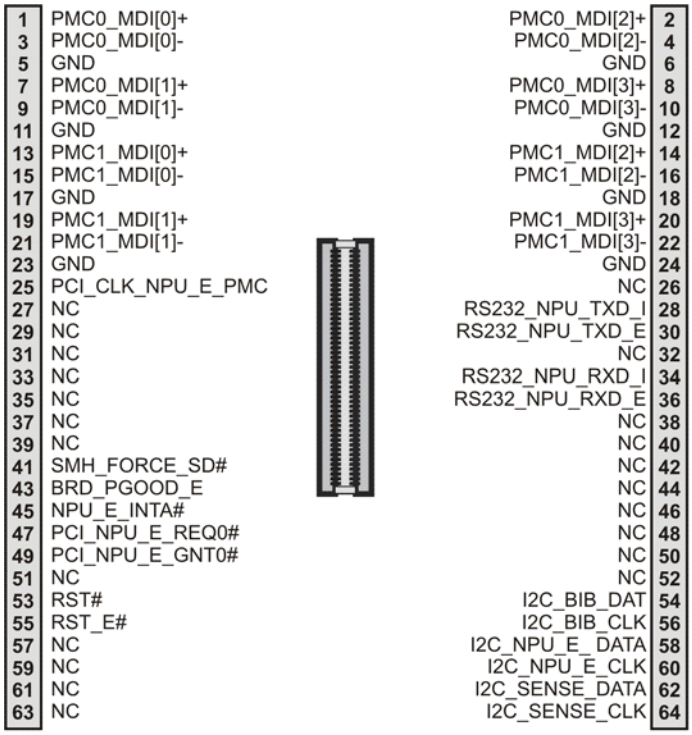


Figure 14: *Jn4 Connector Pinout*

MSF Mezzanine Connector

Two 400-pin MEG-Array® connectors from FCI are used to connect the baseboard with the MSF mezzanine card. One connector contains all MSF, C-bus, LED signals and Power signals. MDI signals, I2C, JTAG and SlowPort signals are routed through the second 400-pin connector.

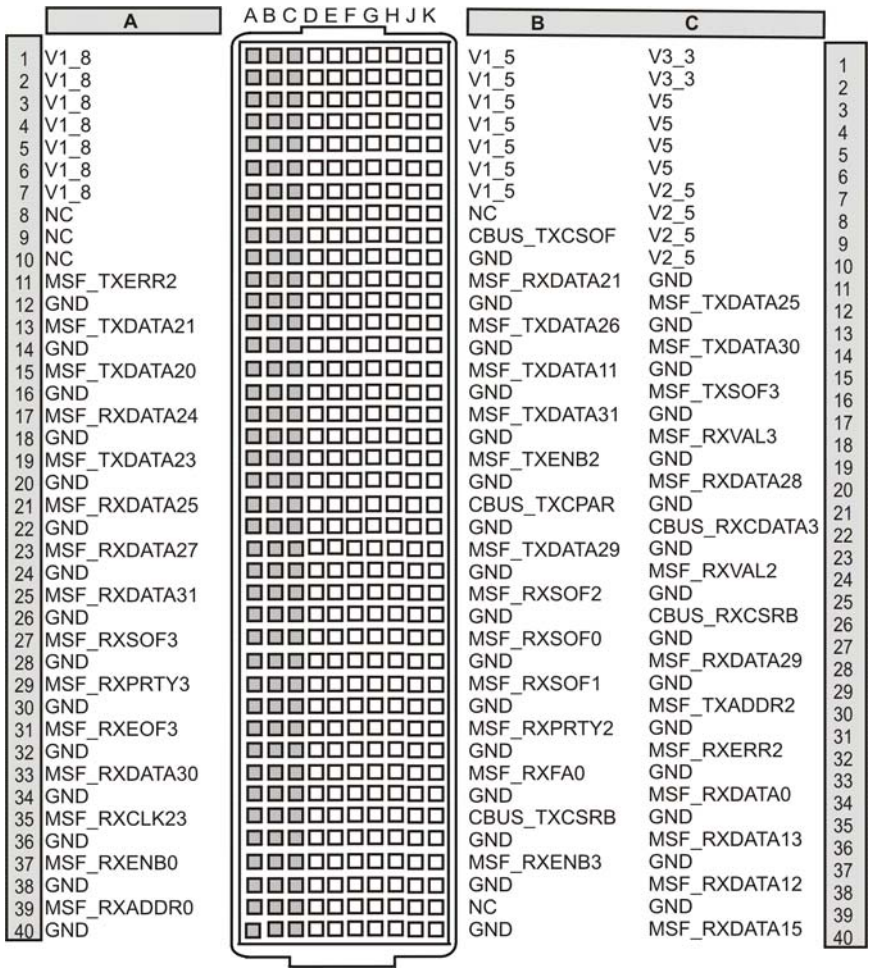


Figure 15: *MSF Mezzanine Connector Pinout (rows A-C).*

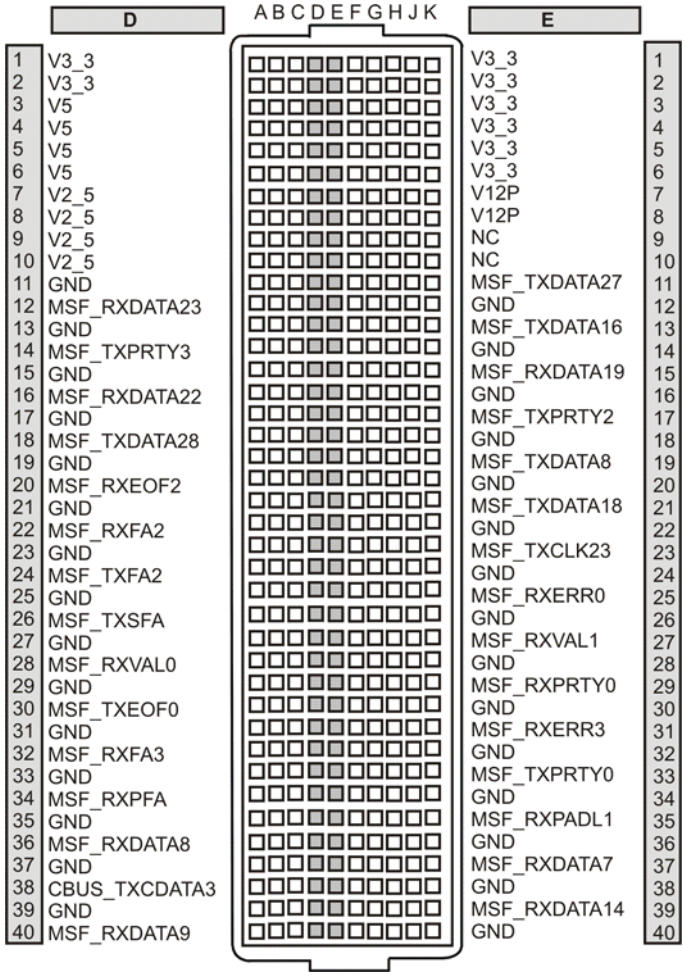


Figure 16: *MSF Mezzanine Connector Pinout (rows D-E)*

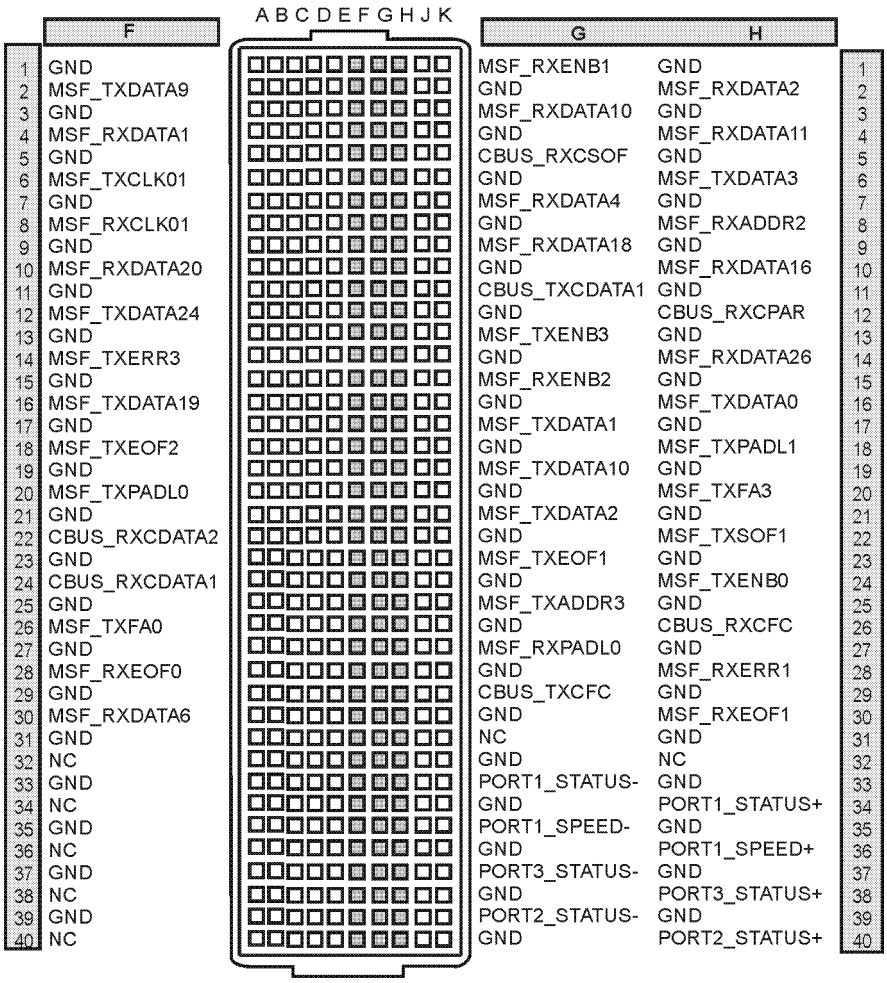


Figure 17: *MSF Mezzanine Connector Pinout (rows F-H)*

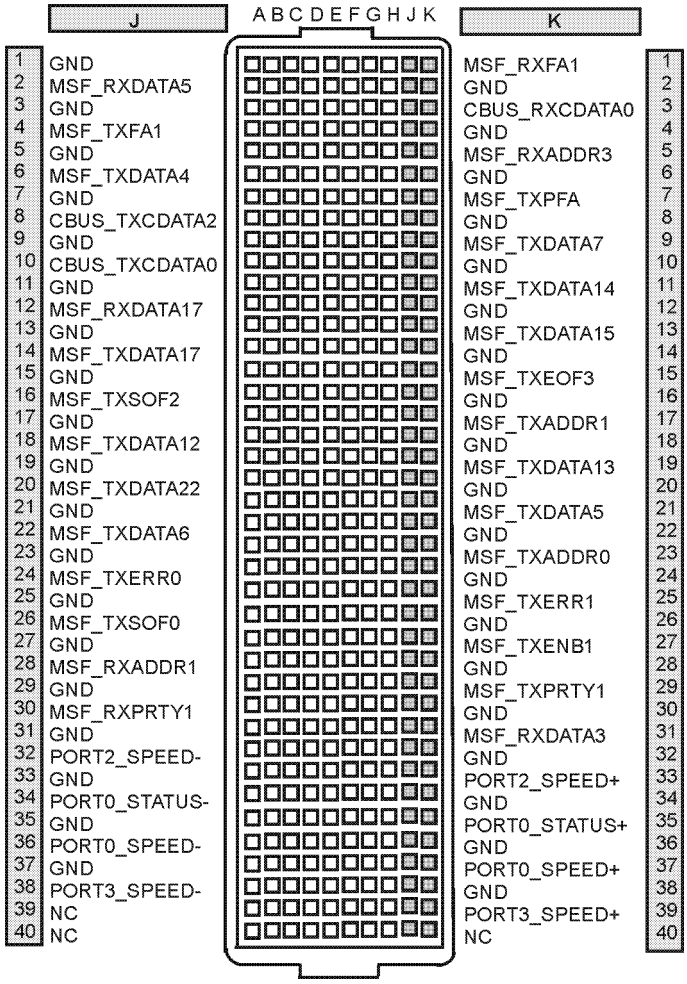


Figure 18: *MSF Mezzanine Connector Pinout (rows J-K)*

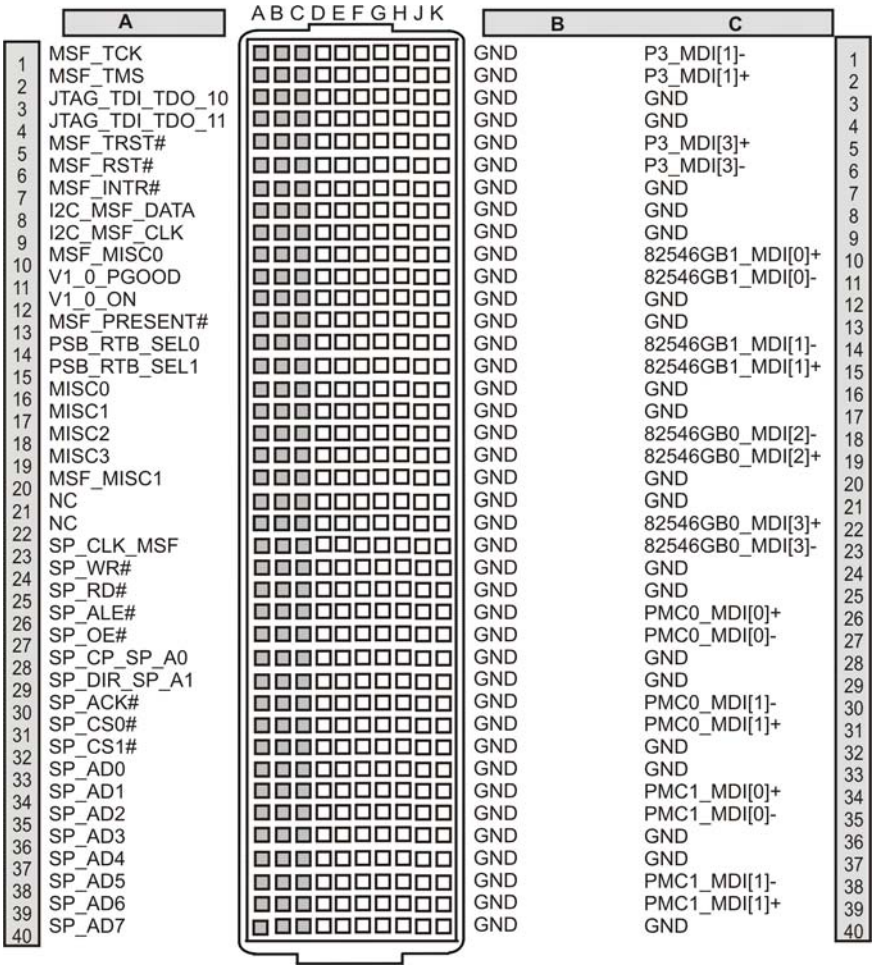


Figure 19: *MSF Mezzanine Connector Pinout (MDI, SlowPort, I2C and JTAG) rows A-C*

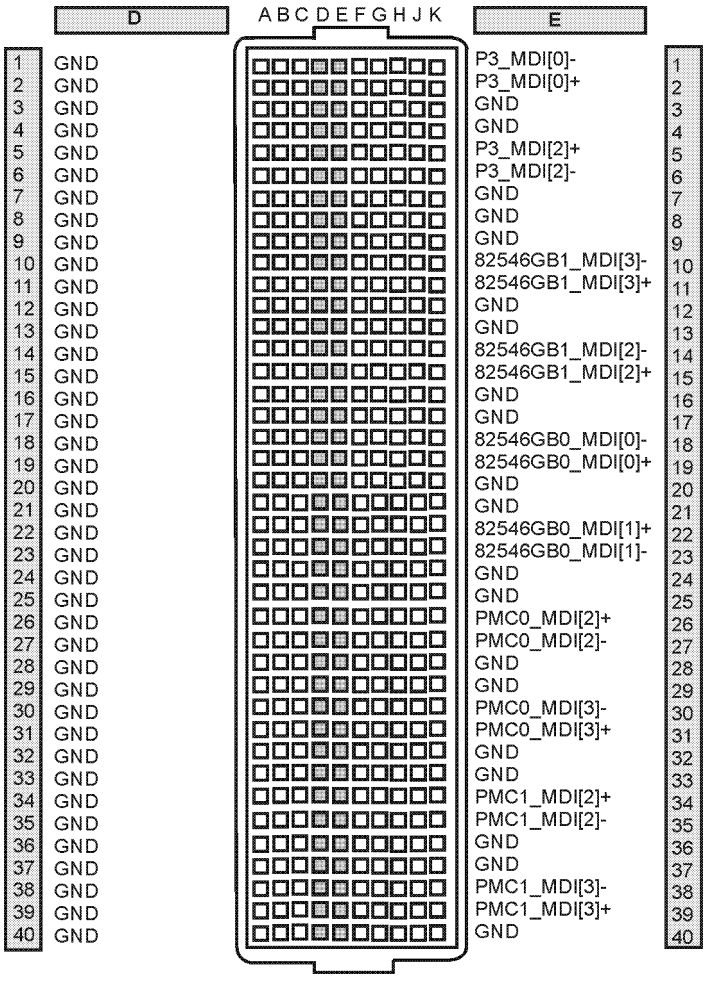


Figure 20: *MSF Mezzanine Connector Pinout (MDI, SlowPort, I2C and JTAG) rows D-E*

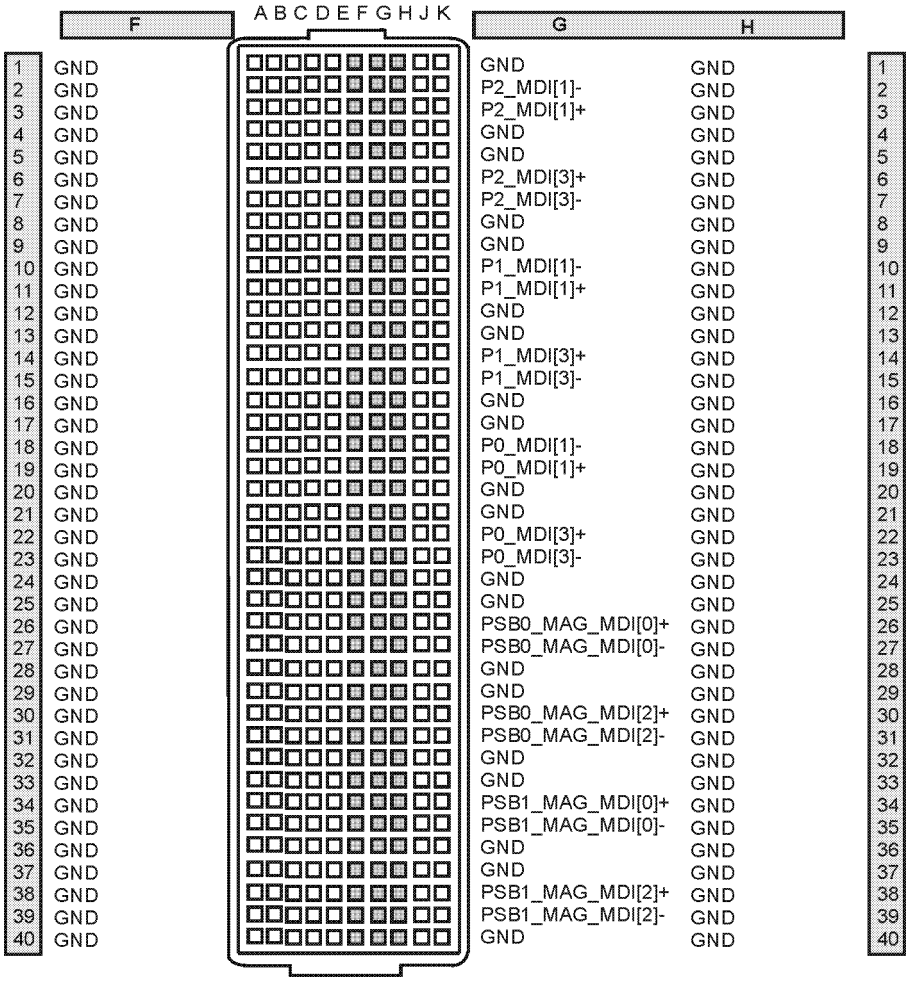


Figure 21: *MSF Mezzanine Connector Pinout (MDI, SlowPort, I2C and JTAG) rows F-H*

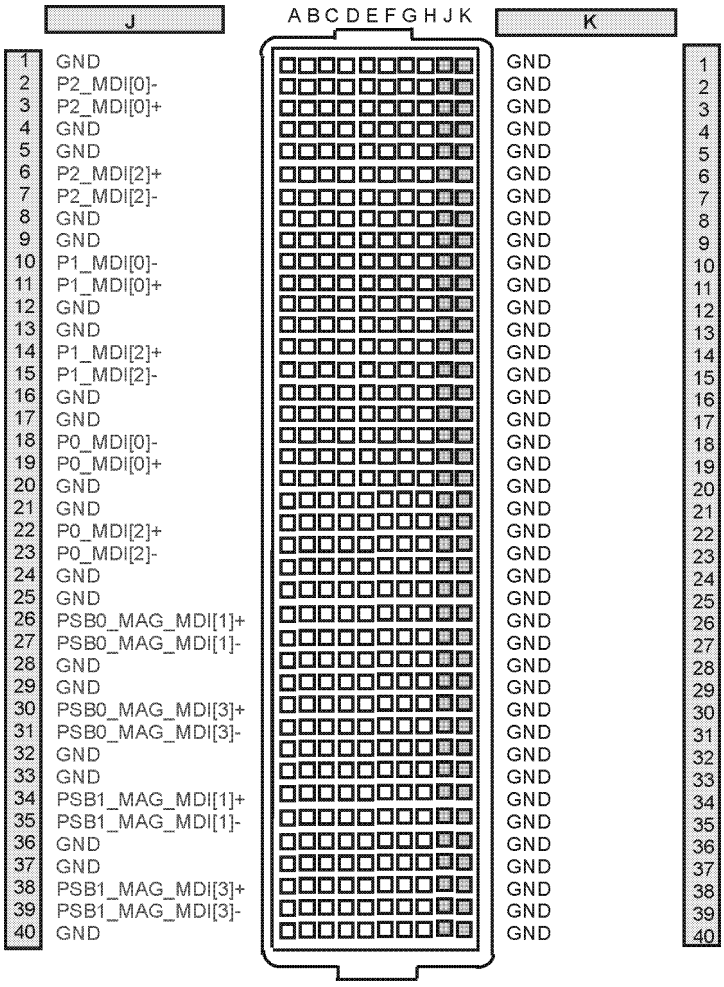


Figure 22: *MSF Mezzanine Connector Pinout (MDI, SlowPort, I2C and JTAG) rows J-K*

Note: Rows B, D, F, H and K are Ground.

LA-1 Mezzanine Connector

A 200-pin MEG-Array® connector from FCI is used to connect the QDR II Channel 1 interface of the IXP2400 to the LA-1 mezzanine card.

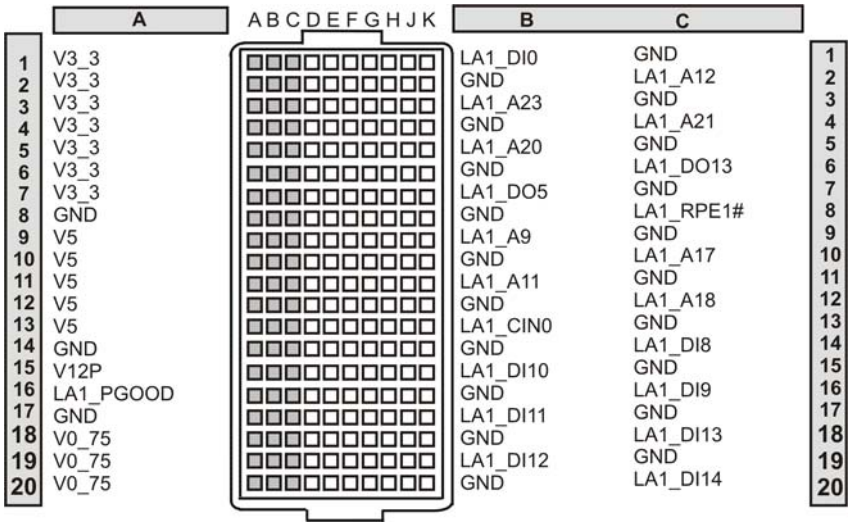


Figure 23: *LA-1 Pinout rows A-C*

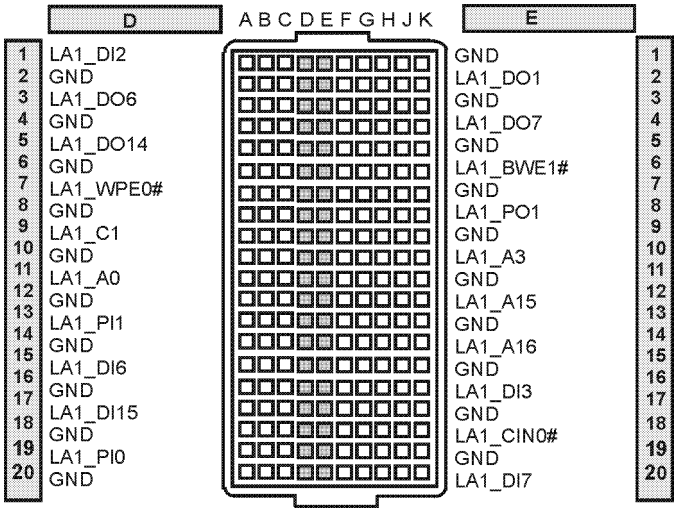


Figure 24: *LA-1 Pinout rows D-E*

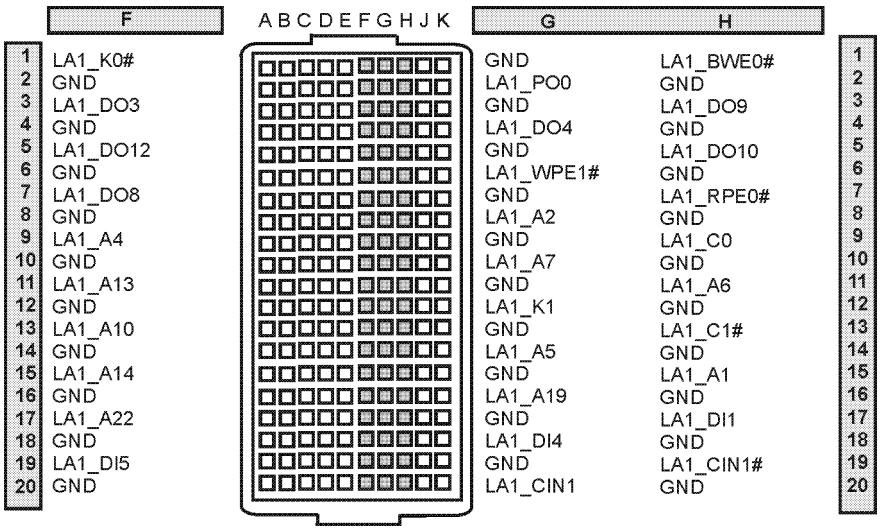


Figure 25: *LA-1 Pinout rows F-H*

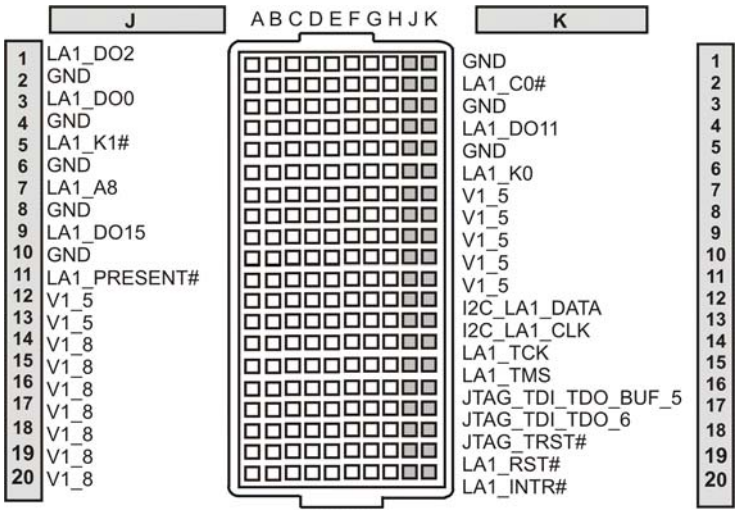


Figure 26: *LA-1 Pinout rows J-K*

4

Devices’ features and Data Paths

Block Diagram

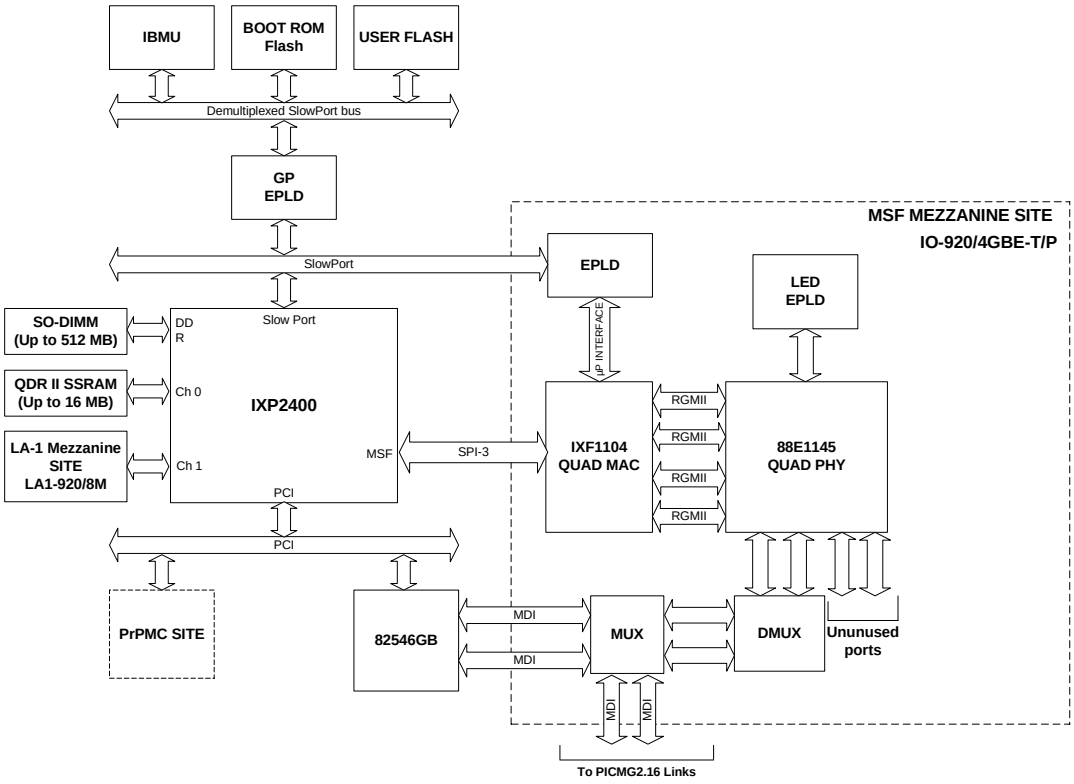


Figure 27: *Single Slot Complex Functional Block Diagram*

The IXP2400

The IXP/CPCI-9120 is based on the Intel IXP2400 Network Processor (Sausalito).

- Half Duplex OC-48 / 2.5 Gbps Network Processor
- Integrated Intel XScale Core that operates at frequencies of 400 MHz and 600 MHz.
- Eight fully programmable multi-threaded microengines for packet forwarding and traffic management on a single chip, supporting 5.4 giga-operations per second.

- Two unidirectional 32-bit LVTTL media interfaces (Rx and Tx) programmable to be SPI-3, UTOPIA 1/2/3 or CSIXL1. Each path is configured for 4x8 bit, 2x16 bit, 1x32 bit or combinations of 8 and 16 bit data paths.
- DDR SDRAM interface with ECC support.
- Two 32-bit QDR SRAM interfaces. This interface is LA-1 compliant and can therefore interface with co-processors.
- 64-bit, 66 MHz PCI interface, Version 2.2
- 1356-Ball FCBGA2 package

PCI Controller

The PCI Controller provides 64-bit, 66-MHz-capable PCI Rev. 2.2 interface. It is also compatible to 32-bit and/or 33 MHz PCI devices. The PCI interface on IXP/CPCI-9120 is clocked at 66 MHz.

DDR Controller

The DDR Controller contains the mechanism that allows the Xscale, microengines and PCI units to access the DDR SDRAM connected to the IXP2400.
IXP/CPCI-9120 supports non-stacked, non-buffered SO-DIMMs for DDR SDRAM. The DDR interface operates at 150 MHz on the IXP/CPCI-9120.

QDR SRAM Interface

The IXP2400 has two independent SRAM controllers, each of which supports pipelined QDR and QDR II SSRAM and/or a coprocessor that adheres to QDR signaling.
One of the two QDR SRAM interfaces of the IXP2400 is used to provide 8 MB of on-board QDR II SSRAM, while the other is connected to an expansion slot.
On IXP/CPCI-9120, the QDR interfaces of IXP2400 operate at 200 MHz.

Media and Switch Fabric Interface

PHY Modes Supported

The Media and Switch Fabric (MSF) Interface connects the IXP2400 to a physical layer device (PHY) and/or a Switch Fabric Interface. MSF consists of the following external interfaces:

- Receive and transmit interfaces, each of which can be individually configured for either UTOPIA (Level 1, 2, and 3), POS-PHY (Level 2 and 3) or CSIX protocols.
- A Flow Control Interface, which provides a point-to-point connection used to pass CSIX-L1-B flow control C-Frames either between two IXP2400 network processors or between an IXP2400 and a CSIX-L1-B switch fabric.
- Each 32-bit interface can be subdivided into 8- or 16-bit channel combinations. The MSF interface uses 3.3V LVTTL (low-voltage transistor-transistor logic) signaling. While the

CSIX standard is a source-synchronous bus, the IXP2400 uses a common-clocking scheme for compatibility with the other protocols.

The MSF interface on IXP/CPCI-9120 is connected to a mezzanine card.

Serial Port

The IXP2400 contains a standard RS-232–compatible universal asynchronous receiver/transmitter (UART), which can be used for communication with a debugger or maintenance console.

Note:Modem controls are not supported.

PCI Bus

IXP/CPCI-9120 has a local PCI Bus.

Gigabit Ethernet Controller

IXP/CPCI-9120 accommodates one 82546GB Dual Port Gigabit Ethernet Controller for Control plane processing. The 82546GB provides two PICMG 2.16 links on a Packet Switching Backplane (PSB).

Processor PMC

The IXP/CPCI-9120 provides one VITA 32–compliant Processor PMC site on to which additional I/O or processing power may be added. This site is connected to the IXP2400 via the local PCI bus.
Processor PMCs when mounted on this site may be configured in monarch or non-monarch mode. When configured as the monarch, the Processor PMC provides enumeration for the PCI bus.

Note: The PMC site on IXP/CPCI-9120 shall support 3V-only PrPMCs.

3.3 V key shall be mounted to prevent inadvertent installation of a non-compliant PMC.

Note: The total power consumption (from +12V, -12V, +3.3V and +5V) of the PMC must not exceed 35 Watts. The power consumption from 5 V and 3.3 V shall not exceed 17.5 Watts and 18 W respectively.

PCI Bus Signaling on the CompactPCI bus

IXP/CPCI-9120 supports universal signaling, thereby allowing it to be inserted into 3.3 V or 5 V backplanes.

Hot Swap Support

IXP/CPCI-9120 supports full hot swap as described in PICMG 2.1 R2.0 (CompactPCI Hot Swap Specification).

Note: Live insertion or extraction of a system slot board is not defined by the PICMG 2.1 R2.0 (CompactPCI Hot Swap Specification), and therefore is not supported on the IXP/CPCI-9120.

Packet Switching Backplane

IXP/CPCI-9120 functions as a dual node board in a packet switching backplane. The Intel® 82546GB Dual Port Gigabit Ethernet Controller described earlier provides the dual PICMG 2.16 links.

Programmable devices

BootROM Flash

The IXP/CPCI-9120 provides 4 MB of BootROM Flash. Support for up to 8 MB of BootROM Flash is provided on the IXP/CPCI-9120.

User Flash

An Intel StrataFlash is used to provide 16 MB of User Flash. Up to 32 MB of User Flash can be accommodated.

IBMU

System management functionality as described in PCIMG 2.9 R1.0 is achieved through the Motorola GmbH IBMU building block (R1.2).
IBMU is a functional unit that provides remote sensing and remote management functionality according to the IPMI specification v1.0.

IBMU also includes a supply voltage generation, a watchdog and an interface for on-board supply voltage and NPU temperature measurement.

IBMU runs totally independent of the IXP2400, its main task being monitoring and logging of events.

IBMU shall be fully functional even when the IXP/CPCI-9120 is not powered.
IBMU is mapped on to IXP2400's SlowPort area. Communication between the IPMI controller and the IXP2400 is achieved through the General Purpose EPLD, which translates slowport signals into X-bus and vice versa.

ICMB

ICMB is an RS-485-based bus used to connect the BMCs of two or more chassis. The ICMB port of IXP/CPCI-9120 is located on the RTB. Connections from the IPMI controller are routed through CompactPCI connectors to the RTB.

5

Daughter Card Details

Media Switch Fabric mezzanine

The MSF mezzanine connects to the MSF interface of the IXP2400. Several flavors are planned, thereby providing added expansion and flexibility.

IO-920

IO-920 MSF Mezzanine card is designed for use in Single Slot complex board. The IO-920 accommodates an Intel® IXF1104 Four Port Gigabit Ethernet Media Access Controller (Vallejo), a Marvell 88E1145 Quad Gigabit Ethernet Transceiver, multiplexers and demulti-lpexers.

Quad Gigabit Ethernet MAC

The Intel® IXF1104 is a 4 x 1 Gigabit Ethernet MAC device that provides a 2.4 to 4.0 Gbps network processor interface to four individual 10/100/1000 Mbps full-duplex and 10/100 half-duplex-capable Ethernet Media Access Controllers (MACs). The interface to the network processor is supported through a System Packet Interface Phase 3 (SPI3) media interface, and the PHY interfaces are Serializer/Deserializer (SerDes) with GBIC support, Gigabit Media Independent Interface, (GMII) or Reduced GMII (RGMII) selected on a per-port basis.
Signals from the SPI-3 interface of the IXF1104 are routed to the MSF interface of the IXP2400 on the baseboard through a 400-pin connector.

Quad Gigabit Ethernet Transceiver

The 88E1145 belongs to Marvell's Alaska™ Quad Family of single-chip devices that contain four independent Gigabit Ethernet Transceivers on a single monolithic IC. Each transceiver performs all the physical layer functions for the 100BASE-TX and 1000BASE-T full or half-duplex Ethernet on CAT5 twisted pair cable, and 10BASE-T full or half-duplex Ethernet on CAT 3,4 and 5 cable. Some of its features are given below:

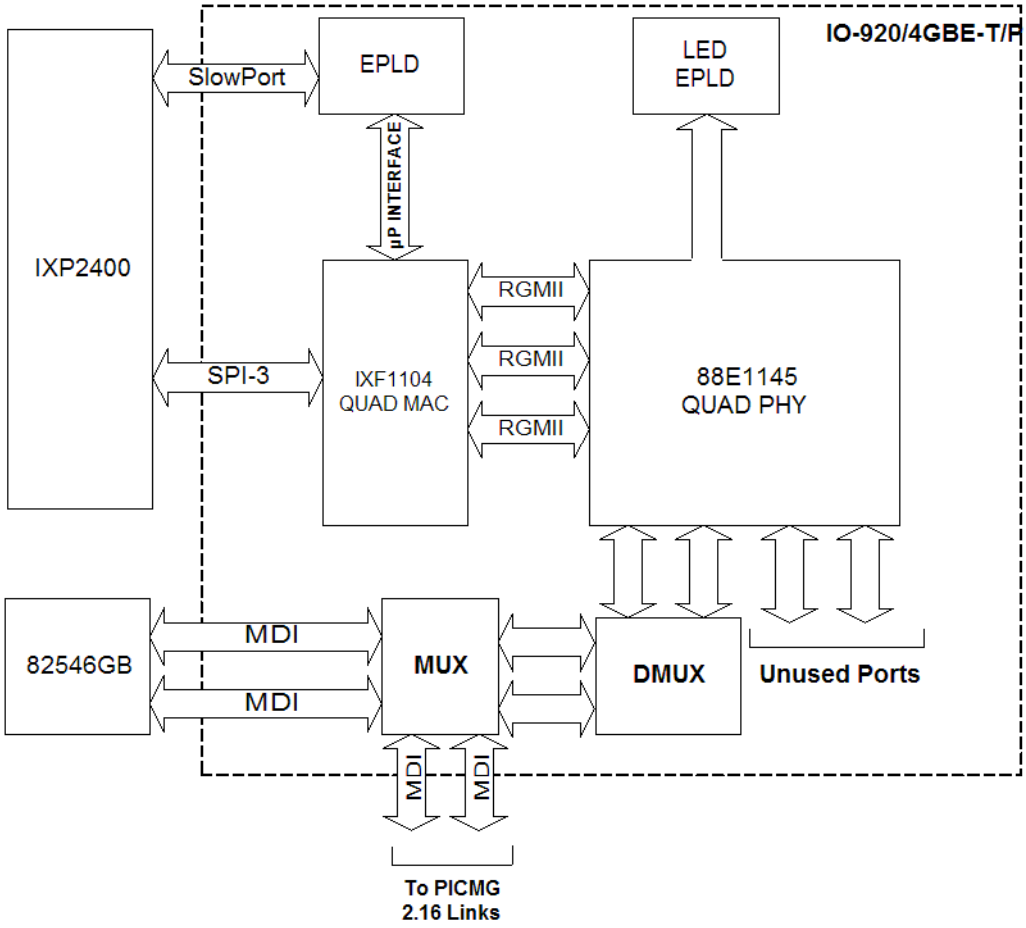
- 10/100/1000BASE-T IEEE 802.3 compliant
- Supports GMII, TBI, RGII, RTBI, and SGMII interfaces.
- Integrated 1.25 GHz SERDES for 1000BASE-X fiber applications
- CRC checker, packet counter
- Packet generation
- Software programmable LED modes

IO-920 Variants

The IO-920 also houses an I/O connector through which MDI lines from the dual gigabit ethernet controller are brought into the IO-920. Through multiplexers and demultiplexers, signals from the on-board PHY and the baseboard Dual MAC (82546GB) can either be routed to the PSB . The muxes are controlled from slowport through EPLD.

Two 1000Base-T PSB (Single Slot Complex PSB Ethernet variant)

In this variant two of the MDI outputs are routed to the PSB.



LA-1/QDR mezzanine

The IXP2400 has two independent SRAM controllers, each of which supports pipelined QDR SSRAM and/or a coprocessor that adheres to QDR signaling.
One of the two QDR SRAM interfaces of the IXP2400 is used to provide 8 MB of on-board QDR II SRAM, while the other is connected to an expansion slot through a 200-pin MEG-Array® connector.

LA1-920/8M

LA1-920/8M is a QDR SRAM memory expansion mezzanine that is connected to Channel 1 of the IXP2400 memory controller. It contains 8 MB of QDR II SSRAM, and is expandable to 16 MB. This interface operates at 200 MHz.

I2C Interface

The IXP2400 detects the presence and type of the LA-1 module by interrogating the SPD device on the LA-1 module. The information contained in the SPD on the LA-1 module may be used to configure the IXP2400's QDR II Ch1 interface to meet the modules operating requirements. The SPD device on the LA-1 module resides on the I2Cbus of the IXP2400. 2K Serial EEPROMs, internally organized with 32 pages of 8 bytes each, shall be used to provide board-specific information.

6

Maps and Registers

PCI memory map

Table 11: *PCI Memory Map*

PCI device	Description	IDSEL	Device number
82546GB	Dual LAN controller	17	1
PrPMC	First PCI agent of the PrPMC	18	2
PrPMC	Second PCI agent of the PrPMC	19	3
IXP2400	NPU	22	6

SlowPort map

Table 12: *SlowPort Map*

Device	SlowPort Address	Description
Flash		
BootROM Flash	C400 0000 – C5FF FFFF	
User Flash	C400 0000 – C5FF FFFF	
General purpose EPLD		
Variant ID	C600 0000	Indicates base board type.
Version No.	C600 0001	EPLD version number
Board Revision No.	C600 0002	Indicates the PCB revision
User Flash select	C600 0003	Provides a means of selecting between User flash and BootROM Flash
CompactPCI register	C600 0010	Provides basic CompactPCI information
PrPMC register	C600 0020	Provides basic PrPMC information
System specific register	C600 0030	Contains information on mezzanine boards and RTB installed
User LED control register	C600 0050	
KCS - SMIC Data 0 Register KSD0	C600 0060	Register in the IPMI Controller, VSC215.
KCS - SMIC Status 0 Register KSS0	C600 0061	Register in the IPMI Controller, VSC215.
KCS - SMIC Data 1 Register KSD1	C600 0070	Register in the IPMI Controller, VSC215.
KCS - SMIC Status 1 Register KSS1	C600 0071	Register in the IPMI Controller, VSC215.
KCS - SMIC Data 2 Register KSD2	C600 0080	Register in the IPMI Controller, VSC215
KCS - SMIC Status 2 Register KSS2	C600 0081	Register in the IPMI Controller, VSC215
Watchdog Timer	C600 0090	External Watchdog Timer Refresh

Table 12: *SlowPort Map*

Device	SlowPort Address	Description
Watchdog Timer Interrupt Clear Register	C600 00A0	Clears the Watchdog Timer interrupt.
Watchdog Enable and Watchdog Reset Enable Register	C60000B0	Enable/Disable the watchdog and watchdog interrupt to reset the board
Watchdog Time Set Register	C60000C0	Select the refresh time of the watchdog Timer
Interrupt Mask Register 1	C700 0008	
Interrupt Mask Register 2	C700 0009	
Interrupt Mask Register 3	C700 000A	
Interrupt Status Register 1	C700 000C	
Interrupt Status Register 2	C700 000D	
Interrupt Status Register 3	C700 000E	
I2C Mux Reset Register	C700 000F	Resets the I2C MUX
MAC EPLD		
Microprocessor interface[1]	C600 4000 - C600 5FFF	Microprocessor Interface of IXF1104
Variant ID	C600 3010	MSF variant indicated here.
Version Number	C600 3011	EPLD version number
Board revision Number	C600 3012	
PHY reset Control register	C600 3020	Provides ability to reset the PHY
Reserved	C600 3030	Reserved.
Reserved	C600 3031	Reserved
Reserved	C600 3032	Reserved
Reserved	C600 3033	Reserved
MUX Control register	C600 3040	Routing of signals to RTB or PSB is controlled here.
MAC pause address register	C600 3050	Selection of the MAC port is done here for pause frame transmission.
MAC pause control register	C600 3060	Initiates the pause frame transmission in the selected ports

Table 12: *SlowPort Map*

Device	SlowPort Address	Description
Interrupt Mask Register	C700 000B	Provides the ability to mask interrupts generated by the PHY and MAC
Interrupt Status Register	C700 000F	Provides status of interrupts issued by the PHY and MAC

1. SlowPort addresses for the MAC are derived from the MAC offsets by multiplying the offset by 4 and adding the result to C600 4000. For example, the SlowPort address for offset 50C is C600 5430, derived as

C6004000 + (0x50C * 4) = C6004000 + 1430 = C6005430

I2C memory map

NPU I2C memory map

Table 13: *NPU I2C Memory Map*

Device	Description	I2C Address
RTC	Real Time Clock	A0
SO-DIMM SPD	SO-DIMM on IXP/CPCI-9120	A2
MSF EEPROM	EEPROM	A4
I2C MUX	Multiplexer for RTC	E0
LA-1 site	LA-1 on IXP/CPCI-9120	A8
PrPMC site	PrPMC site	AC

SENSE I2C memory map

Table 14: *SENSE I2C Memory Map*

Device	Description	I2C Address
LM87	Voltage/Temperature sensor on IXP/CPCI-9120	5A

BIB I2C Memory Map

Table 15: *BIB I2C Memory Map*

Device	Description	I2C Address
24LC128	Board BIB	A0
AT24C02	IBMU BIB	AE

Register Details

General Purpose EPLD

Variant ID

Address: C600 0000
Type: Read only
This register uniquely identifies the board as IXP/CPCI-9120 and also indicates the amount of on-board QDR II SSRAM installed.

Table 16: *Variant ID*

Bit	Description	Access
3:2	00: 8 MB on-board QDR II SSRAM	RO
	01: 16 MB on-board QDR II SSRAM	
	1X: RFU	
1:0	00: IXP/CPCI-9120	RO
	01: RFU	
	10: RFU	
	11: RFU	

Version Number

Address: C600 0001
Type: Read only
This register indicates the EPLD revision number.

Table 17: *General Purpose EPLD-Version Number*

Bit	Description	Access
3:0	EPLD revision number	RO

Board Revision Number

Address: C600 0002
Type: Read only
This register indicates the board revision number.

Table 18: *General Purpose EPLD-Board Revision Number*

Bit	Description	Access
3:0	Board revision number	RO

User Flash Select

Address: C600 0003
Type: Read/Write
User Flash and BootROM Flash is mapped in the same area of memory. This register enables either type to be selected.

Table 19: *User Flash Select*

Bit	Description	Access
0	0: Selects BootROM Flash	R/W
	1: Selects User Flash	
	Reset value: 0	

CompactPCI Register

Address: C600 0010
Type: Read/Write

This register provides status of signals related to CompactPCI and hot swap. It also provides control of the hot swap blue LED.

Table 20: *CompactPCI Register*

Bit	Description	Access
3	0: IXP/CPCI-9120 in System Slot	RO
	1: IXP/CPCI-9120 in Peripheral Slot	

Table 20: *CompactPCI Register*

Bit	Description	Access
2	0: PCI Present	RO
	1: PCI not present	
1	0: Hot Swap Eject Switch Open	RO
	1: Hot Swap Eject Switch Closed	
0	0: Blue LED Off	R/W
	1: Blue LED On	
	Reset value: 1	

PrPMC register

Address: C600 0020
Type: Read/Write

This register provides status of signals related to the Processor PMC site. It also provides control of the EREADY signal.

Table 21: *PrPMC Register*

Bit	Description	Access
1	EREADY control	
	Note that the EREADY pin of the EPLD is configured as an output when the PrPMC is made Monarch. Otherwise, it is an input on the EPLD.	
	EREADY (OUT): If Processor PMC is a Monarch	WO
	Reset value: 0	
	EREADY IN: If Processor PMC is a non-Monarch	RO
	0: Non-monarch processor has not yet finished its initialization.	
	1: Non-monarch processor has completed its initialization.	
0	0: Processor PMC is a Monarch	RO
	1: Processor PMC is a non-Monarch	

System specific register

Address: C600 0030

Type: Read Only
System-specific configurations are indicated here.

Table 22: *System Specific Register*

Bit	Description	Access
3	0: PrPMC Mezzanine Card Present	RO
	1: PrPMC Mezzanine Card Absent	
2	0: Reserved	RO
	1: Reserved	
1	0: MSF Mezzanine Card Present	RO
	1: MSF Mezzanine Card Absent	
0	0: LA1 Mezzanine Card Present	RO
	1: LA1 Mezzanine Card Absent	

User LED Control register

Address: C600 0050
Type: Read/Write
Used to control the bi-color user LEDs on the front panel. On reset/power-on, none of the LEDs glow. In order to obtain green or yellow color, the register must be written as below:

Table 23: *User LED Control Register*

Bit	Description	Access
3:2	00: User LED2 does not glow	R/W
	01: User LED2 glows green	
	10: User LED2 glows yellow	
	11: User LED2 does not glow	
1:0	00: User LED1 does not glow	R/W
	01: User LED1 glows green	
	10: User LED1 glows yellow	
	11: User LED1 does not glow	

Watchdog Timer Register

Address: C600 0090

Type: Write Only

This register must be accessed periodically (refreshed). Failure to access this register results in an interrupt. Failure to refresh this register after an interrupt has been issued will result in a reset of the entire system. Refresh periodicity should be within a second.

Table 24: *Watchdog Timer Register*

Bit	Description	Access
0	0,1: Access (read or write) to this register refreshes the Watch Dog Timer	W

Watchdog Timer Interrupt Clear Register

Address: C600 00A0

Type: Write Only

This register provides a means of clearing the Watchdog interrupt.

Table 25: *Watchdog Timer Interrupt Clear Register*

Bit	Description	Access
0	0,1: Access to this register clears the Watch Dog Timer Interrupt Reset value: 1	WO

Watchdog Enable and Watchdog Reset Enable Register

Address: C600 00B0

Type: Read/Write

This register provides a means to enable or disable the watchdog timer as well as to enable or disable the watchdog interrupt to reset the board.

Table 26: *Watchdog Enable and Watchdog Reset Enable Register*

Bit	Description	Access
0	0: Enable watchdog Timer 1: Disable watchdog Timer Reset Value: 1	RW
1	0: Enable watchdog timer Reset 1: Disable watchdog timer Reset Reset Value: 1	RW

Watchdog Time Set Register

Address: C600 00C0
Type: Read/Write
This register provides a means to select the refresh time of the watchdog timer.

Table 27: *Watchdog Time Set Register*

Bit	Description	Access
2:0	111: Refresh Time of 2 secs 110: Refresh Time of 3 secs 101: Refresh Time of 4 secs 100: Refresh Time of 5 secs 011: Refresh Time of 6 secs 010: Refresh Time of 7 secs 001: Refresh Time of 8 secs 000: Refresh Time of 9 secs Reset Value:111	RW

Interrupt Mask Register-1

Address: C700 0008
Type: R/W
Interrupts can be masked through the Interrupt mask registers. All interrupts are masked on reset/power-on.

Table 28: *Interrupt Mask Register-1*

Bit	Description	Access
7	0: Reserved 1: Reserved Reset value: 1	R/W
6	0: MSF Interrupt is not masked 1: MSF Interrupt is masked Reset value: 1	R/W
5	0: LA1 Interrupt is not masked 1: LA1 Interrupt is masked Reset value: 1	R/W
4	0: IPMI Interrupt-2 is not masked 1: IPMI Interrupt-2 is masked Reset value: 1	R/W
3	0: IPMI Interrupt-1 is not masked 1: IPMI Interrupt-1 is masked Reset value: 1	R/W

Table 28: *Interrupt Mask Register-1*

Bit	Description	Access
2	0: IPMI Interrupt-0 is not masked 1: IPMI Interrupt-0 is masked Reset value: 1	R/W
1	0: 82546 Interrupt-B is not masked 1: 82546 Interrupt-B is masked Reset value: 1	R/W
0	0: 82546 Interrupt-A is not masked 1: 82546 Interrupt-A is masked Reset value: 1	R/W

Interrupt Mask Register-2

Address: C700 0009
Type: R/W

Interrupts can be masked through the Interrupt mask registers. All interrupts are masked on reset/power-on.

Table 29: *Interrupt Mask Register-2*

Bit	Description	Access
7	0: PMC Interrupt-D is not masked 1: PMC Interrupt-D is masked Reset value: 1	R/W
6	0: PMC Interrupt-C is not masked 1: PMC Interrupt-C is masked	R/W
5	0: PMC Interrupt-B is not masked 1: PMC Interrupt-B is masked	R/W
4	0: PMC Interrupt-A is not masked 1: PMC Interrupt-A is masked	R/W
3	0: RFU 1: RFU	R/W
2	0: RFU 1: RFU	R/W

Table 29: *Interrupt Mask Register-2*

Bit	Description	Access
1	0: RFU 1: RFU	R/W
0	0: RFU 1: RFU Reset value: 1	R/W

Interrupt Mask Register-3

Address: C700 000A

Type: R/W

Interrupts can be masked through the Interrupt mask registers. All interrupts are masked on reset/power-on.

Table 30: *Interrupt Mask Register-3*

Bit	Description	Access
2	0: Voltage Monitor Interrupt is not masked 1: Voltage Monitor Interrupt is masked Reset value: 1	R/W
1	0: Watchdog Timer Interrupt is not masked 1: Watchdog Timer Interrupt is masked Reset value: 1	R/W
0	RFU Reset value: 1 This bit must always be masked on the IXP/CPCI-9120.	RFU

Interrupt Status Register-1

Address: C700 000C

Type: Read Only

This register provides Interrupt status. When interrupted, the NPU can interrogate the three Interrupt status registers to determine the interrupt-source.

Table 31: *Interrupt Status Register-1*

Bit	Description	Access
7	0: Reserved 1: Reserved	RO
6	0: MSF Interrupt is active 1: MSF Interrupt is not active	RO
5	0: LA1 Interrupt is active 1: LA1 Interrupt is not active	RO
4	0: IPMI Interrupt-2 is active 1: IPMI Interrupt-2 is not active	RO
3	0: IPMI Interrupt-1 is active 1: IPMI Interrupt-1 is not active	RO
2	0: IPMI Interrupt-0 is active 1: IPMI Interrupt-0 is not active	RO
1	0: 82546 Interrupt-B is active 1: 82546 Interrupt-B is not active	RO
0	0: 82546 Interrupt-A is active 1: 82546 Interrupt-A is not active	RO

Interrupt Status Register-2

Address: C700 000D
Type: Read Only
This register provides Interrupt status. When interrupted, the NPU can interrogate the three Interrupt status registers to determine the interrupt-source.

Table 32: *Interrupt Status Register-2*

Bit	Description	Access
7	0: PMC Interrupt-D is active 1: PMC Interrupt-D is not active	RO
6	0: PMC Interrupt-C is active 1: PMC Interrupt-C is not active	RO
5	0: PMC Interrupt-B is active 1: PMC Interrupt-B is not active	RO
4	0: PMC Interrupt-A is active: 1: PMC Interrupt-A is not active	RO

Table 32: *Interrupt Status Register-2*

Bit	Description	Access
3	RFU	
2	RFU	
1	RFU	
0	RFU	

Interrupt Status Register-3

Address: C700 000E
Type: Read Only
This register provides Interrupt status. When interrupted, the NPU can interrogate the three Interrupt status registers to determine the interrupt-source.

Table 33: *Interrupt Status Register-3*

Bit	Description	Access
3	0: Ejector handles are open 1: Ejector handles are closed.	RO
2	0: Voltage Monitor Interrupt is active 1: Voltage Monitor Interrupt is inactive	RO
1	0: Watchdog Timer Interrupt is active 1: Watchdog Timer Interrupt is inactive	RO
0	RFU	

I2C MUX Reset Register

Address: C700 000F
Type: Write Only
This register provides a means of resetting the 4-channel I2C switch, PCA9546. An active-LOW reset input allows the PCA9546 to recover from a situation where one of the down-stream I2C buses is stuck in a LOW state. Pulling the RESET pin LOW resets the I2C state machine and causes all the channels to be deselected, as does the internal power-on reset function.

Table 34: *I2C MUX Reset Register*

Bit	Description	Access
0	In order to reset the I2C MUX, this must be zero. Reset value: 1	WO

MAC EPLD-IO-920

Registers of the EPLD in IO-920 are described below.

Variant ID

Address: C600 3010
Type: Read only
This register provides information on the board variant.

Table 35: *MAC EPLD - IO-920-Variant ID*

Bit	Description	Access
3:0	0000: IO-920 (Front variant)	RO
	0001: IO-920 (Rear variant)	
	0010: RFU	
	0011: RFU	
	0100: 2.16 Variant	
	0101 to 1111: RFU	

Version Number

Address: C600 3011
Type: Read only
This register indicates the EPLD revision number.

Figure 28: *MAC EPLD – IO-920-Version Number*

Bit	Description	Access
3:0	EPLD Revision number for the current board	RO

Board Revision Number

Address: C600 3012
Type: Read only
This register indicates the board revision number.

Figure 29: *Board revision Number*

Bit	Description	Access
3:0	Board Revision number for the current board	RO

PHY Reset Control Register

Address: C600 3020
Type: Read/Write
This register provides a means of controlling PHY operation. The PHY can be held in reset until the OS is ready to initialize the PHY. This delayed reset is useful in cases where it is necessary to prevent link-pulses from being sent on the line before the OS has booted (Example: Rev C0 silicon of Marvell's 88E1145).

Table 36: *PHY Reset Control Register*

Bit	Description	Access
0	0: Quad PHY Reset is asserted 1: Quad PHY Reset is de-asserted Reset value: 0	R/W

MUX Control Register

Address: C600 3040
Type: Read/Write
This register controls the flow of Ethernet signals between the RTB and PSB.

Figure 30: *Mux Control Register*

Bit	Description	Access
1:0	01: Port0 and Port2 of the Quad PHY are routed to RTB Port0 and Port1 of the 82546 are routed to PSB 10: Port0 and Port2 of the Quad PHY are routed to PSB 00,11: Reserved Reset value: 01	R/W

MAC PAUSE Address Register

Address: C600 3050
Type: Read/Write
This register provides a means of selecting the port on to which XON or XOFF packets must be transmitted.

Table 37: *MAC PAUSE Address Register*

Bit	Description	Access
2:0	000: XON Packet, on all IXF1104 Ports 001: XOFF Packet, on Port-0 010: XOFF Packet, on Port-1 011: XOFF Packet, on Port-2 100: XOFF Packet, on Port-3 110-101: Reserved 111: XOFF Packet, on all Ports Reset value: 000	RW

MAC PAUSE Control Register

Address: C600 3060
Type: Read/Write
After the port has been selected using the MAC PAUSE Address Register, transmission of the XON/XOFF packets is initiated using this register.

Table 38: *MAC PAUSE Control Register*

Bit	Description	Access
0	This bit is to be set (1) and then reset (0) once the address has been set in the MAC Pause address register. This initiates the transmission of the PAUSE FRAME. Reset value: 0	RW

Interrupt Mask Register

Address: C700 000B
Type: Read/Write
Interrupts generated from the PHY may be masked through this register.

Table 39: *Interrupt Mask Register*

Bit	Description	Access
3	0: PHY Port-3 Interrupt is not masked 1: PHY Port-3 Interrupt is masked Reset value: 1	R/W
2	0: PHY Port-2 Interrupt is not masked 1: PHY Port-2 Interrupt is masked Reset value: 1	R/W
1	0: PHY Port-1 Interrupt is not masked 1: PHY Port-1 Interrupt is not active Reset value: 1	R/W
0	0: PHY Port-0 Interrupt is not masked 1: PHY Port-0 Interrupt is masked Reset value: 1	R/W

Interrupt Status Register

Address: C700 000F

Type: Read Only

This register provides Interrupt status of interrupts generated by the PHY. When interrupted, the NPU can interrogate this register to determine the interrupt-source.

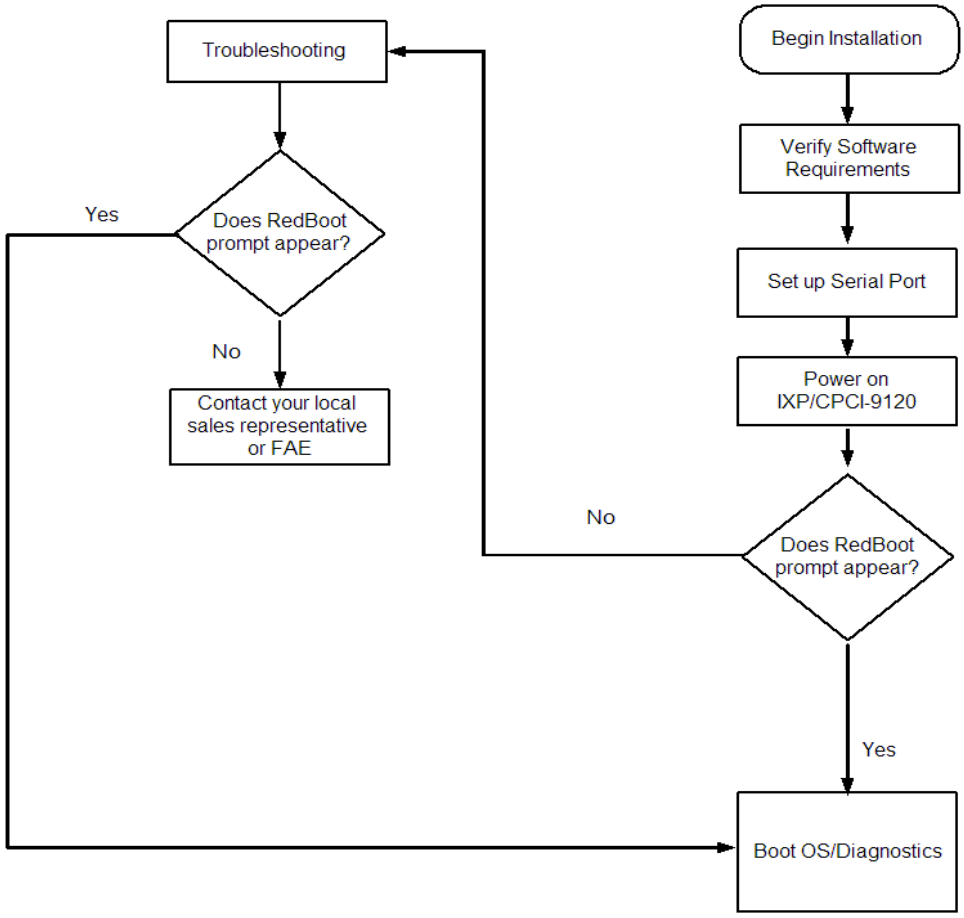
Table 40: *Interrupt Status Register*

Bit	Description	Access
3	0: PHY Port-3 Interrupt is active 1: PHY Port-3 Interrupt is not active	RO
2	0: PHY Port-2 Interrupt is active 1: PHY Port-2 Interrupt is not active	RO
1	0: PHY Port-1 Interrupt is active 1: PHY Port-1 Interrupt is not active	RO
0	0: PHY Port-0 Interrupt is active 1: PHY Port-0 Interrupt is not active	RO

7

RedBoot

Action Plan


Troubleshooting
Begin Installation
Verify Software Requirements
Yes
Does RedBoot prompt appear?
Set up Serial Port
No
Power on IXP/CPCI-9120
Contact your local sales representative or FAE
No
Does RedBoot prompt appear?
Yes
Boot OS/Diagnostics

Introduction

This chapter details the RedBoot firmware that runs on the IXP 2400 processor on the IXP/CPCI-9120 board. This chapter also details the commands that are supported by the firmware.
The main purpose of RedBoot for IXP/CPCI-9120 is to boot VxWorks, Diagnostics, and Linux. The RedBoot is also used to write an OS image to a determined location on the Flash. Currently RedBoot runs at 600 MHz. Certain parameters can be stored in Flash and are retained after reboot. These parameters can be changed using the RedBoot commands.
The serial interface setting is 9600 bps with 8-N-1 and no flow control.

Building Images

1. Create a folder RedBoot and untar the source file frcCPCI920_RedBoot_SRC_Rel_<x_x>.tar into RedBoot folder.
2. Verify whether a directory named SOURCES has been created.
3. The tool chain for building RedBoot can be downloaded from the following link:

http://public.planetmirror.com/pub/redhat-gnupro/dev_tools/010413/
http://sunsite.rediris.es/sites/download.intel.nl/design/intelxscale/dev_tools/010413/
The file to be downloaded is i686-pc-linux-gnulibc2.1-x-xscale-elf.tar.Z.

4. Uncompress the file i686-pc-linux-gnulibc2.1-x-xscale-elf.tar.Z.
5. Untar the file i686-pc-linux-gnulibc2.1-x-xscale-elf.tar.
6. Verify whether a directory named H-i686-pc-linux-gnulibc2.1 has been created.
7. Add the directory H-i686-pc-linux-gnulibc2.1/bin/ to the PATH variable.
8. In the SOURCES directory,

   make the files RUNIT.sh, packages/swapbytes_x86-linux and ecosconfig files executable using **`chmod`** command.
9. Execute the RUNIT.sh script.

The Binaries will be created in the build/install/bin/ directory.

Loading Images from RedBoot

RedBoot

1. Execute **boot_flash** command to switch to boot flash.

```
RedBoot> boot_flash
```

2. Load the redboot.bin image to SDRAM using **load** command

```
RedBoot> load redboot.bin -b 0x400000
```

3. Unlock the boot flash from 0xC40000000 to 0xC4040000 using **fis unlock** command

```
RedBoot> fis unlock -f 0xc4000000 -l 0x40000
```

4. Program the redboot.bin image to the boot flash using **fis create** command at address 0xC4000000

```
RedBoot> fis create RedBoot -b 0x400000 -f 0xc4000000 -l 0x40000
```

Diagnostics

1. Execute **boot_flash** command to switch to boot flash.

```
RedBoot> boot_flash
```

2. Load the diag.bin image to SDRAM using **load** command

```
RedBoot> load diag.bin -b 0x400000
```

3. Program the diag image to the boot flash using **fis create** command at address 0xC4100000

```
RedBoot> fis create diag -b 0x400000 -f 0xc4100000 -l 0x100000 -r 0x400000 -e
0x400000
```

4. Execute **fis load diag** command to load it to RAM

```
RedBoot>fis load diag
```

5. Execute **go** command to execute diag.

```
RedBoot> go 0x400000
```

RedBoot Commands

This section details the commands available in RedBoot.

Common Commands

The general format of commands is:
command <options, parameters>
Numbers, such as a memory location, may be specified in either decimal or hexadecimal (requires a 0x prefix).

ping

SYNTAX

```
ping [-v] [-n <count>] [-l <length>] [-t <timeouts>] [-r <rate>][-i
<IP_addr>] -h IP_addr
```

DESCRIPTION

The **ping** command checks the connectivity of the local network by sending special (ICMP) packets to a specific host. These packets should be automatically returned by that host. The command will indicate how many of these round-trips were successfully completed.

Note: Make sure an ethernet interface is configured before using `ping`.

Arguments

-v Be verbose, displaying information about each packet sent.
-n <count> Controls the number of packets to be sent. Default is 10 if `-n` is not specified.
-t <timeout> How long to wait for the round-trip to complete, specified in milliseconds. Default is 1000ms (1 second).
-r <rate> How fast to deliver packets, i.e. time between successive sends. Default is 1000ms (1 second). Specifying **"-r 0"** will send packets as quickly as possible.
-l <length> Each packet contains some amount of payload data. This option specifies the length of that data. The default is 64 and the value is restricted to the range 64 .. 1400.
-i <local IP> This allows the ping command to override its local network address. While this is not recommended procedure, it can help diagnose some situations, for example where BOOTP is not working properly.
-h <host IP> The address of the other device to contact.

Cache Settings

SYNTAX

cache [ON | OFF]

DESCRIPTION

This command is used to manipulate the caches on the processor. With no options, this command specifies the state of the system caches. When an option is given, the caches are turned off or on appropriately.

Checksum

SYNTAX

cksum -b <location> -l <length>

DESCRIPTION

Computes the POSIX (see “POSIX” page -18) checksum on a range of memory -either RAM or Flash. The value printed can be compared with the output from the Linux program ‘chksum’.

```
RedBoot>
```

Memory Dump

SYNTAX

dump -b <location> [-l <length>]

DESCRIPTION

This command displays the contents of memory in hexadecimal format. It is most useful for examining a segment of RAM or flash. Note that it could be detrimental if used on memory mapped hardware registers.

The memory is displayed at most sixteen bytes per line, first as the raw hex value, followed by an ASCII interpretation of the data.

```
RedBoot> du -b 0x100 -l 0x80
0x00000100: 3C60 0004 6063 2000 7C68 03A6 4E80 0020 |<`..`c .|h..N.. |
0x00000110: 0000 0000 0000 0000 0000 0000 0000 0000 |................|
0x00000120: 0000 0000 0000 0000 0000 0000 0000 0000 |................|
0x00000130: 0000 0000 0000 0000 0000 0000 0000 0000 |................|
0x00000140: 0000 0000 0000 0000 0000 0000 0000 0000 |................|
0x00000150: 0000 0000 0000 0000 0000 0000 0000 0000 |................|
0x00000160: 0000 0000 0000 0000 0000 0000 0000 0000 |................|
0x00000170: 0000 0000 0000 0000 0000 0000 0000 0000 |................|
```

Reset

SYNTAX

reset

DESCRIPTION

This command resets the system.

```
RedBoot>reset
```

boot_flash

SYNTAX

boot_flash

DESCRIPTION

This command is used to switch to boot flash

```
RedBoot>boot_flash
```

user_flash

SYNTAX

user_flash

DESCRIPTION

This command is used to switch to user flash

```
RedBoot>user_flash
```

help

SYNTAX

help [<topic>]

DESCRIPTION

This command is used for help.

Detect I2C devices

SYNTAX

i2cdetect

DESCRIPTION

This command detects and displays all I2C devices.

```
RedBoot> i2cdetect
Detecting I2C devices......
I2C MUX is present
PrPMC is not present
RTB is present
LA-1 MEZZANINE is not present
MSF MEZZANINE is present
SO-DIMM-SPD is present
RTC is present
RedBoot>
```

Write into I2C device

SYNTAX

`i2c_write [slave address] [offset] [data]`

DESCRIPTION

Write into I2C for the specified slave address and offset.

Data field should be a continous hexadecimal string to write into I2C device.

```
RedBoot> i2c_write 0xa4 0 555555
RedBoot>
```

Read from I2C device

SYNTAX

`i2c_read [slave address] [offset] [count]`

DESCRIPTION

Read from I2C for the specified slave address and offset

```
RedBoot> i2c_read 0xa4 0 6
55  55  55   0   0   0
RedBoot>
```

Read byte

SYNTAX

`inb [address]`

DESCRIPTION

This command reads the value of the byte present at [address]

```
RedBoot>inb 0xc6000003
0x1
RedBoot>
```

Read dword

SYNTAX

`inl [address]`

DESCRIPTION

This command reads the value of the long word at [address].

```
RedBoot>inl 0xc4000000
0xfefefefe
RedBoot>
```

Read word

SYNTAX

inw [address]

DESCRIPTION

This command reads the value of the word at [address].

```
RedBoot>inl 0xc4000000
0xfefe
RedBoot>
```

Read IPMI info

SYNTAX

ipmi [netfn] [command] [lun] <data>

DESCRIPTION

This command reads the IPMI info for the specified netfn, cmd, lun
Data field is optional and depends on IPMI command.
The following is an example of the GetDeviceID command.

```
RedBoot> ipmi 6 1 0
LUN/NetFn      : 0x0/0x7
Command        : 0x1
Completion Code: 0x0
0x1     0x81    0x99    0x7     0x1     0x2b    0x48    0xe     0x0     0xc     0
x20     0x0     0x0     0x0     0x0     0x0     0x0     0x0     0x0     0x0     0
x0

RedBoot>
```

Write Byte

SYNTAX

outb [address] [value]

DESCRIPTION

This command writes value to byte address.

```
RedBoot>outb 0xc6000000 0x0
```

Write dword

SYNTAX

outl [address] [value]

DESCRIPTION

Write value to dword address

```
RedBoot>outl 0xc6000003 0x0
```

Write Word

SYNTAX

outw [address] [value]

DESCRIPTION

This command writes value to word address.

```
RedBoot>outw 0xc6000000 0x0
```

Read PCI config registers

SYNTAX

pci_config_read [-b <bus>] [-d <device>] [-f <func>] [-r <reg>]

DESCRIPTION

This command reads the PCI Configuration Register for the specified values of the bus number, device number, function and the register number.

```
RedBoot> pci_config_read -b 0 -d 2 -f 0 -r 2
0x12298086
RedBoot>
```

Write PCI config Registers

SYNTAX

pci_config_write [-b <bus>] [-d <device>] [-f <func>] [-r <reg>] [-v <data>]

DESCRIPTION

This command writes data to the PCI Configuration Register for the specified values of the bus number, device number, function and the register number.

```
RedBoot> pci_config_write -b 0 -d 2 -f 0 -r 2 -v 12298086
RedBoot>
```

Display PCI device Info

SYNTAX

`pci_device_show [-b <bus>] [-d <device>] [-f <func>]`

DESCRIPTION

This command shows all configuration registers for the PCI device for the given bus number, device number and function number.

```
RedBoot> pci_device_show -b 0 -d 2 -f 0
PCI Device Configuration Header:
(0x00) PCI device and vendor IDs:       0x12298086
(0x04) PCI status and command:          0x00000007
(0x08) PCI revision ID:                 0x00000008
(0x0c) PCI cache line size:             0x00
(0x0d) PCI latency timer:               0x00
(0x0e) PCI header type:                 0x00
(0x0f) PCI BIST:                        0x00
(0x10) PCI base address 0:              0x04040000
(0x14) PCI base address 1:              0x00001081
(0x18) PCI base address 2:              0x04100000
(0x1c) PCI base address 3:              0x00000000
(0x20) PCI base address 4:              0x00000000
(0x24) PCI base address 5:              0x00000000
(0x28) PCI CardBus CIS pointer:         0x00000000
(0x2c) PCI subsystem IDs:               0x00000000
(0x30) PCI expansion rom address:       0x00000000
(0x34) PCI Capabilities:                0x000000dc
(0x3c) PCI interrupt line:              0x00
(0x3d) PCI interrupt pin:               0x01
(0x3e) PCI minimum grant:               0x08
(0x3f) PCI maximum latency:             0x38
RedBoot>
```

Scan PCI

SYNTAX

`pci_scan`

DESCRIPTION

This command scans and diplays all PCI devices

```
RedBoot> pci_scan
... Bus = 0, Dev = 1, Fun = 0, Vendor = 8086, Device = 1079
... Bus = 0, Dev = 1, Fun = 1, Vendor = 8086, Device = 1079
... Bus = 0, Dev = 2, Fun = 0, Vendor = 8086, Device = 1229
... Bus = 0, Dev = 6, Fun = 0, Vendor = 8086, Device = 9001
RedBoot>
```

Program MAC Address

SYNTAX

`mac_prog [id1] [id2] [id3] [id4] [id5] [id6]`

DESCRIPTION

The command stores the MAC address of the backplane ethernet controller into the flash. The command should be followed by the program_eeprom command that stores the MAC address into the EEPROM of 82546.

```
RedBoot>mac_prog 0xbe 0xba 0xbe 0xba 0xbe 0xba
... Unlock from 0xc4040000-0xc4060000: .
... Erase from 0xc4040000-0xc4060000: .
... Program from 0x00054d78-0x00054d7e at 0xc4040000: .
... Lock from 0xc4040000-0xc4060000: .
RedBoot>
```

Program EEPROM

SYNTAX

program_eeprom

DESCRIPTION

This command programs the MAC address to 82546 EEPROM. Before executing this command, execute mac_prog command.

```
RedBoot> program_eeprom
BEBA BEBA BEBA FFFF FFFF FFFF FFFF FFFF FFFF FFFF 4603 1004 124B 1079 8086 3178
0000 1079 FFFF 0000 1800 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
0000 FFFF FFFF 0000 0800 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 0602
0120 4000 110F FFFF 0120 4000 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 8A22
RedBoot>
```

Read EEPROM

SYNTAX

read_eeprom

DESCRIPTION

This command reads 128 bytes of 82546 EEPROM.

```
RedBoot> read_eeprom
BEBA BEBA BEBA FFFF FFFF FFFF FFFF FFFF FFFF FFFF 4603 1004 124B 1079 8086 3178
0000 1079 FFFF 0000 1800 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
0000 FFFF FFFF 0000 0800 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 0602
0120 4000 110F FFFF 0120 4000 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 8A22
RedBoot>
```

Version Information

SYNTAX

version

DESCRIPTION

This command displays RedBoot version information.

```
RedBoot> version
Platform: CPCI-9120 (XScale)  BootMonitor Version 1.0
SDRAM(256M): 0x00000000 - 0x10000000, 0x00200000 - 0x0ffdf000 available
SRAM Channel 0(8M): 0x80000000 - 0x80800000
SRAM Channel 1(8M): 0x90000000 - 0x90800000
```

```
FLASH(16M): 0xc4000000 - 0xc5000000, 128 blocks of 0x00020000 bytes each.
Base Board Rev: 0  Base Board EPLD Rev: 2
IO-920 BOARD Rev: 10  IO-920 EPLD  Rev: 2
```

82546_backplane

SYNTAX

`82546_backplane`

DESCRIPTION

This command is used to have control plane connectivity to the back plane.

```
RedBoot> 82546_backplane
RedBoot>
```

ixf_backplane

SYNTAX

`ixf_backplane`

DESCRIPTION

This command is used to have data plane connectivity to the back plane.

```
RedBoot> ixf_backplane
RedBoot>
```

Download Process

load

SYNTAX

```
load {file} [-v] [-b location] [-r] [-m {[xmodem]|[ymodem]|[tftp]} ] [-h
host_IP_address]
```

DESCRIPTION

The load command is used to download data into the target system. Data can be loaded via a network connection, using either the TFTP protocol, or the console serial connection using the X/Y modem protocol. Files may also be loaded directly from local filesystems on disk.
Files to be downloaded may either be executable images in SREC (see “SREC” page -19) format or raw data.

Arguments

`file` The name of the file on the TFTP server.
`-v` Display a small spinner (indicator) while the download is in progress. This is just for feedback, especially during long loads. Note that the option has no effect when using a serial download method since it would interfere with the protocol.
`-b` Specify the location in memory to which the file should be loaded. Executable images normally load at the location to which the file was linked. This option allows the file to be loaded to a specific memory location, possibly overriding any assumed location.
`-r` Download raw data. Normally, the load command is used to load executable images into memory. This option allows for raw data to be loaded. If this option is given, `-b` will also be required.
`-m` The -m option is used to select the download method. The choices are:
xmodem, ymodem, zmodem serial download using standard protocols over the console serial port. When using this method, the file parameter is not required.
`tftp` Network based download using the TFTP protocol.
`-h` Used explicitly to name a host computer to contact for the download data. This works in TFTP mode only.

```
RedBoot> load redboot.bin -b 0x400000
```

Note: In this example, the `redboot.bin` file has been loaded into the RAM address 0x400000.

Flash Image System (FIS)

Executable images, as well as data, can be stored in flash in a simple file store. The **fis** command is used to manipulate and maintain flash images.

The available **fis** commands are:

SYNTAX

fis init [-f]

DESCRIPTION

This command is used to initialize the flash Image System (FIS). It should only be executed once, when RedBoot is first installed on the hardware. Subsequent executions will cause loss of data in the flash (previously-saved images will no longer be accessible).
If the **-f** option is specified, all blocks of flash memory will be erased as part of this process.

```
RedBoot> fis init -f
About to initialize [format] flash image system - are you sure (y/n)? n
```

SYNTAX

fis [-c] [-d] list

DESCRIPTION

This command lists the images currently available in the FIS. Certain images used by RedBoot have fixed names. Other images can be manipulated by the user.
If the **-c** option is specified, the image checksum is displayed instead of the Mem Addr field.
If the **-d** option is specified, the image datalength is displayed instead of the length [amount of flash used]. The datalength is the length of data within the allocated flash image actually being used for data.

```
RedBoot> fis list
Name              FLASH addr  Mem addr    Length      Entry point
RedBoot           0xC4000000  0x00300000  0x00040000  0x00000000
System Log        0xC43A0000  0xC43A0000  0x00020000  0x00000000
RedBoot config    0xC43DF000  0xC43DF000  0x00001000  0x00000000
FIS directory     0xC43E0000  0xC43E0000  0x00020000  0x00000000
vxRag             0xC4200000  0x00300000  0x00100000  0x00000000
red               0xC4040000  0x00300000  0x00040000  0x00000000


RedBoot> fis list
Name              FLASH addr  Mem addr    Length      Entry point
RedBoot           0xC4000000  0x00000000  0x00040000  0x00000000
System Log        0xC43A0000  0xC43A0000  0x00020000  0x00000000
RedBoot config    0xC43DF000  0xC43DF000  0x00001000  0x00000000
FIS directory     0xC43E0000  0xC43E0000  0x00020000  0x00000000
diag1             0xC4100000  0x00300000  0x00100000  0x00300000
vxRag             0xC4200000  0xC4200000  0x00100000  0x00000000
redboot           0xC4100000  0x00300000  0x00100000  0x00000000
```

SYNTAX

```
fis free
```

DESCRIPTION

This command shows which areas of the flash memory are currently not in use. 'In use' means that the block contains non-erased contents. Since it is possible to force an image to be loaded at a particular flash location, this command can be used to check whether that location is in use by any other image.

```
RedBoot> fis free
0xC4040000 .. 0xC4100000
0xC4200000 .. 0xC43A0000
```

Note: There is currently no cross-checking between actual flash contents and the image directory, which means that there could be a segment of flash which is not erased that does not correspond to a named image, or vice-versa.

SYNTAX

```
fis create -b <mem_base> -l <length> [-f <flash_addr>] [-e <entry_point>] [-r
<ram_addr>] [-s <data_length>] -n <name>
```

DESCRIPTION

This command creates an image in the FIS directory. The data for the image must exist in RAM memory before the copy. Typically, you would use the RedBoot load command to load an image into RAM and then the **`fis create`** command to write it to flash.

Arguments

`name`: The name of the file, as shown in the FIS directory.
`-b:` The location in RAM used to obtain the image. This is a required option.
`-l:` The length of the image. If the image already exists, then the length is inferred from when the image was previously created. If specified, and the image exists, it must match the original value.
`-f:` The location in flash for the image, which will be inferred for extant images if not specified. If this is not provided, the first freeVblock which is large enough will be used.
`-e:` The execution entry address. This is used if the starting address for an image is not known, or needs to be overridden.
`-r:` The location in RAM when the image is loaded via **`fis load`**. This only needs to be specified for images which will eventually loaded via **`fis load`**. Fixed images, such as RedBoot itself, will not need this.
`-s:` The length of the actual data to be written to flash. If not present then the image length (-1) value is assumed. If the value given by **`-s`** is less than -1, the remainder of the image in flash will be left in an erased state.

Note: By using this option it is possible to create a completely empty flash image, for example to reserve space for use by applications other than RedBoot.

-n: Only if **-n** is specified, the FIS directory is updated, and no data is copied from RAM to flash. This feature can be used to recreate the FIS entry if it has been destroyed.

```
RedBoot> fis create -b 0x400000 -r 0x400000 -l 0x100000 VxWorks -f 0xc4200000
Make sure flash is unlocked
... Erase from 0xc4200000-0xc4300000: ........
... Program from 0x00400000-0x00500000 at 0xc4200000: ........
... Unlock from 0xc43e0000-0xc4400000: .
... Erase from 0xc43e0000-0xc4400000: .
... Program from 0x0ffdf000-0x0ffff000 at 0xc43e0000: .
... Lock from 0xc43e0000-0xc4400000: .
If you are loading an existing file, then the fis create command will provide
some values automatically, such as the flash address and flash length.
RedBoot> fis create -b 0x400000 RedBoot
An image named 'RedBoot' exists - are you sure (y/n)? y
* CAUTION * about to program 'RedBoot'
            at 0xc4000000..0xc403ffff from 0x00400000 - are you sure (y/n)? y
Your action requires changes to BootMonitor blocks
Continue - are you sure (y/n)? y
Make sure flash is unlocked
... Erase from 0xc4000000-0xc4040000: ..
... Program from 0x00400000-0x00440000 at 0xc4000000: ..
... Lock from 0xc4000000-0xc4040000: ..
... Unlock from 0xc4fe0000-0xc5000000: .
... Erase from 0xc4fe0000-0xc5000000: .
... Program from 0x0ff5b000-0x0ff7b000 at 0xc4fe0000: .
... Lock from 0xc4fe0000-0xc5000000: .
```

SYNTAX

```
fis load [-b <memory load address>] [-c] -d <name>
```

DESCRIPTION

This command is used to transfer an image from flash memory to RAM. Once loaded, it may be executed using the **go** command. If **-b** is specified, then the image is copied from flash to the specified address in RAM. If **-b** is not specified, the image is copied from flash to the load address given when the image was created.

Arguments

name: The name of the file, as shown in the FIS directory

-b: Specify the location in memory to which the file should be loaded. Executable images normally load at the location to which the file was linked. This option allows the file to be loaded to a specific memory location, possibly overriding any assumed location.

-c: Compute and print the checksum of the image data after it has been loaded into memory.
-d: Decompress gzipped image while copying it from flash to RAM.

```
RedBoot> fis load Diag
RedBoot> go
```

SYNTAX

fis delete name

DESCRIPTION

This command removes an image from the FIS. The flash memory will be erased as part of the execution of this command, as well as removal of the name from the FIS directory.

```
RedBoot> fis delete vxWorks
Delete image 'vxWorks' - are you sure (y/n)? y
... Erase from 0xc4200000-0xc4300000: ........
... Unlock from 0xc43e0000-0xc4400000: .
... Erase from 0xc43e0000-0xc4400000: .
... Program from 0x0ffdf000-0x0ffff000 at 0xc43e0000: .
... Lock from 0xc43e0000-0xc4400000: .
```

SYNTAX

fis lock -f <flash_addr> -l <length>

DESCRIPTION

This command is used to write-protect (lock) a portion of flash memory, to prevent accidental overwriting of images. In order to make make any modifications to the flash, a matching unlock command must be issued. This command is optional and will only be provided on hardware which can support write-protection of the flash space.

```
RedBoot fis lock -f 0xC4040000 -l 0x20000
... Lock from 0xC4040000-0xC4060000: .
```

SYNTAX

fis unlock -f <flash_addr> -l <length>

DESCRIPTION

This command is used to unlock a portion of flash memory forcibly, allowing it to be updated. It must be issued for regions which have been locked before the FIS can reuse those portions of flash.

```
RedBoot fis unlock -f 0xC4040000 -l 0x20000
... Unlock from 0xC4040000-0xC4060000: .
```

SYNTAX

fis erase -f <flash_addr> -l <length>

DESCRIPTION

This command is used to erase a portion of flash memory forcibly. There is no cross-checking to ensure that the area being erased does not correspond to a loaded image.

```
RedBoot> fis erase -f 0xC4040000 -l 0x20000
... Erase from 0xC4040000-0xC4060000: .
```

SYNTAX

fis write -b <location> -l <length> -f <flash addr>

DESCRIPTION

Writes data from RAM at <location> to flash.

```
RedBoot> load -r -b 0x400000 -h 10.208.32.68 vxWorks.st_rom.bin
Raw file loaded 0x00400000-0x0049e530
RedBoot>
RedBoot> fis write -f 0xc4200000 -b 0x400000 -l 0x100000
* CAUTION * about to program FLASH
            at 0xc4200000..0xc42fffff from 0x00400000 - are you sure (y/n)? y
Make sure flash is unlocked
... Erase from 0xc4200000-0xc4300000: ........
... Program from 0x00400000-0x00500000 at 0xc4200000: ........
```

SYNTAX

fconfig <option> nickname [value]
nickname [value]

DESCRIPTION

Manages configuration kept in FLASH memory

A Reset has to be issued for the changed parameters to take effect.

Options:

-i - This option prompts the user whether to reset the configuration values to default state or not.

Example

```
RedBoot> fconfig -i
          Initialize non-volatile configuration - are you sure (y/n)? n
          Run script at boot: false
          Use BOOTP for network configuration: false
          Local IP address: 10.208.33.99
          Default server IP address: 10.208.33.50
          Network debug: false
          Default network device: i82559_fe0
          RedBoot>
          RedBoot> fconfig -i
          Initialize non-volatile configuration - are you sure (y/n)? y
          Run script at boot: false
          Use BOOTP for network configuration: true
          Network debug: false
          Default network device: i82546_gb0
```

-l - The configuration data is simply listed.

Example

```
RedBoot> fconfig -l
          Run script at boot: false
          Use BOOTP for network configuration: false
          Local IP address: 10.208.33.99
          Default server IP address: 10.208.33.50
          Network debug: false
          Default network device: i82546_gb2
          RedBoot>
```

-n - Displays nicknames of the parameter.

Example

```
RedBoot> fconfig -n
          boot_script: false
          bootp: false
          bootp_master_ip: 10.208.33.99
          bootp_server_ip: 10.208.33.50
          net_debug: false
          net_device: i82546_gb1
          RedBoot>
```

-f - Displays the full name.

Example

```
          RedBoot> fconfig -f
          Run script at boot: false
          Use BOOTP for network configuration: false
          Local IP address: 10.208.32.99
          Default server IP address: 10.208.33.50
          Network debug: false
          Default network device: i82546_gb0
          RedBoot>
```

nickname [value]- Nicknames provide a quick way to set a single entry using the format. If no value is supplied, the command will list and prompt for only that entry. If a value is supplied, then the entry will be set to that value. You will be prompted to write the new information into flash if any change was made.

Example

```
RedBoot> fconfig boot_script
                  boot_script: true f
                  Update RedBoot non-volatile configuration - are you sure (y/n)?
y
                  ... Unlock from 0xc43c0000-0xc43e0000: .
                  ... Erase from 0xc43c0000-0xc43e0000: .
                  ... Program from 0x00059d80-0x0005ad80 at 0xc43df000: .
                  ... Lock from 0xc43c0000-0xc43e0000: .
                  RedBoot>
                  RedBoot> fconfig bootp_master_ip 10.208.33.133
                  bootp_master_ip: 10.208.33.99 Setting to 10.208.33.133
```

```
                    Update RedBoot non-volatile configuration - are you sure (y/n)?
y
                    ... Unlock from 0xc43c0000-0xc43e0000: .
                    ... Erase from 0xc43c0000-0xc43e0000: .
                    ... Program from 0x00059d80-0x0005ad80 at 0xc43df000: .
                    ... Lock from 0xc43c0000-0xc43e0000: .
                    RedBoot>
```

Note: Provide IP address as per the available network configuration.

Executing Programs from RedBoot

Once an image has been loaded into memory, either via the load command or the **`fis load`** command, execution may be transferred to that image.

Note: Once the `go` command is executed, control cannot be transferred back to the Red Boot.

---

1. **`go`** - Execute a program
   The format of the **`go`** command is:
   `RedBoot>` **`go [-w time] [location]`**

Execution will begin at location if specified. Otherwise, the entry point of the last image loaded will be used.
The **`-w`** option gives the user time seconds before execution begins. The execution may be aborted by typing **`Ctrl+C`** on the console. This mode would typically be used in startup scripts.

2. **`exec`** - Execute a Linux kernel image

Arguments

```
[-w timeout]
[-b <load addr> [-l <length]]
[-r <ramdisk addr>
[-s <ramdisk length>]]
[-c "kernel command line"] [<entry_point>]
[-p <params address>]
```

8

Diagnostics

Introduction

Diagnostics allows detailed testing of various hardware interfaces and modules on the IXP/CPCI-9120. The IXP/CPCI-9120 boards are shipped with RedBoot and Diagnostics images in the boot flash.

Requirements

- Console or Windows-based PC with serial port.
- A terminal emulation progam, such as HyperTerminal that runs on Windows.

The settings on the terminal emulation program are given below:

- Bits per second (bit rate): 9600 bps
- Data bits: 8
- Parity: None
- Stop bits: 1
- Flow control: None

Setup

1. Install IXP/CPCI-9120 (see “IXP/CPCI-9120 Board Installation” page -54).
2. Connect the cable (ACC/CABLE/MD9-D9/2) (see “Ordering Information Excerpt: Non RoHS Part Number” page -37) to the PC or console through a null modem cable.

Procedure

Power-on the IXP/CPCI-9120.

From the RedBoot prompt, type the following:

```
fis load diag
go
```

Configuring the Platform

Executing Diagnostics

Diagnostics are executed under the control of a diagnostic command line interface as shown below

```
CPCI920_DIAG>
```

Run any of the diagnostics tests that are listed in the subsequent sections.

Board Diagnostics Commands

Note: All memory addresses are entered in hexadecimal format without the *0x* prefix. When specified without `addr` and `size`, it displays the contents of the last address when the command was issued.

Listed below are the different Board Diagnostics commands for IXP/CPCI-9120:
t/test

t/test

DESCRIPTION

Displays a list of test commands within the diagnostics image.

SYNTAX

```
test <interface/module>
```

PARAMETERS

`interface/module` is the interface or module that has to be tested.

Example:

The following command invokes the xscale scratchpad tests.

```
t xscale s p
```

help/h?

DESCRIPTION

Prints the help screen for a given diagnostic command.

SYNTAX

```
h <command>
```

PARAMETERS

`command` is the command for which help information has to be displayed

Note: A specific command is optional. You can enter `h` to get help on all diagnostic commands.

Example:
For help on the fill command, type
`h fill`
The results one can expect from a working board are given below:

```
CPCI920_DIAG> h fill
fill: <low> <high> <val> (fill mem)
Arguments: <Hex><Hex><Hex>
```

banner

DESCRIPTION

Prints a banner containing basic hardware and software information.

SYNTAX

`Banner`
Example:
`Banner`

exit/reset

DESCRIPTION

reset command

SYNTAX

`Exit`
`Reset`
Example:
`Exit`
`Reset`

tf

DESCRIPTION

Enables test flags. Causes given device to be included or excluded from tests in the **`test all`** command.

SYNTAX

`tf <test_type_ID> <status>`

PARAMETERS

`test_type_id` test identifier. Can be any of the following:

- **0 - SRAM**
- **1 - SDRAM**
- **2 - UENG**
- **3 - TIMERS**
- **4 - PCI**

- 5 - XSCALE
- 6 - I2C
- 7 - GPIO
- 8 - UART
- 9 - MSF
- A - EPLD
- B - SPI3
- C - Flash
- F - All Test

- **status** test flag. Can be any of the following:
  - 0 - do not perform test
  - 1 - perform test

Example:

```
tf 4 1
```

p

DESCRIPTION

Displays the contents of memory that starts at address **addr** and is **size** bytes long.

SYNTAX

p/pb/pw/pd <addr> <size>

PARAMETERS

- **addr** starting address of a memory block
- **size** length of a memory block

Example:

```
p 800000 80
pb 500000
pw 300000
pd 000
```

c

DESCRIPTION

Interactively changes the contents of a specified memory address.

SYNTAX

c <addr>

PARAMETERS

addr memory address that will change

Example:

```
c 1000000
cb 210000
cw 300000
cd 500000
```

wrv

DESCRIPTION

Writes, reads and verifies to a specified memory address.

SYNTAX

```
wrv <location> <size>
```

PARAMETERS

- **location** - address of a memory block
- **size** - length of a memory block (decimal format)

Example:

```
wrv 800000 80
```

blockcopy

DESCRIPTION

Copies a block of memory between address **low** and **high** to address **dest**.

SYNTAX

```
blockcopy <low> <high> <dest>
```

PARAMETERS

- **low** - address in memory where the block begins
- **high** - address in memory where the block ends
- **dest** - address in memory where the block should be copied to

Example:

```
blockcopy 800000 800100 400000
```

fill

DESCRIPTION

Fills a block of memory with the specified value.

SYNTAX

```
fill <low> <high> <val>
```

PARAMETERS

- **low** - address in memory where the block begins
- **high** - address in memory where the block ends
- **val** - value that will be copied to every memory cell in the specified block

Example:

```
fill 800000 800100 41
```

mem

DESCRIPTION

Reads (r) or writes (w) a byte (b), word (w) or dword (d) from/to memory.

SYNTAX

```
mem[rw] [bwd]
memrb <addr> <rept> <silent>
memwb <addr> <data> <rept> <silent>
memrw <addr> <rept> <silent>
memww <addr> <data> <rept> <silent>
memrd <addr> <rept> <silent>
memwd <addr> <data> <rept> <silent>
```

PARAMETERS

- **addr -** memory address that will be written to/read from
- **rept -** repeat this on a loop. Can be set as follows:
  - **0** - perform a read or write one time only
  - **1** - perform a read or write on a loop until a key is pressed
- **data -** data to write to the memory address
- **silent -** do not trace any information on the console
  - 0 - display information on the console
  - 1 - run in silent mode

Example:
`memrb 1 0 0`

io

DESCRIPTION

Reads (r) or writes (w) a byte (b), word (w) or dword (d) from/to a PCI IO.

SYNTAX

```
io[rw] [bwd]
iorb <addr> <rept> <silent>
iowb <addr> <data> <rept>
iorw <addr> <rept> <silent>
ioww <addr> <data> <rept>
iord <addr> <rept> <silent>
iowd <addr> <data> <rept>
```

PARAMETERS

- **addr -** memory address that will be written to/read from
- **rept -** repeat this on a loop. Can be set as follows:

— **0** - perform a read or write one time only
— **1** - perform a read or write on a loop until a key is pressed

- **data -** data to write to the memory address
- **silent -** do not trace any information on the console
  - **0**- display information on the console
  - **1**- run in silent mode

Example:

```
iorb 1 0 0
```

cfg

DESCRIPTION

Reads (r) or writes (w) a byte (b), word (w) or dword (d) from/to a PCI device.

SYNTAX

```
cfg[rw] [bwd]
cfgrb <bus> <device> <function> <addr>
cfgwb <bus> <device> <function> <addr> <data>
cfgrw <bus> <device> <function> <addr>
cfgww <bus> <device> <function> <addr> <data>
cfgrd <bus> <device> <function> <addr>
cfgwd <bus> <device> <function> <addr> <data>
```

PARAMETERS

- **bus** - PCI bus number of the device
- **device** - device number of the device
- **function** - function number of the device
- **addr** - the n^{th} byte/word/dword required
- **data** - data to write to the memory address

Example:

```
cfgrb 1 0 0
```

mac

DESCRIPTION

Read all/read/write the MAC registers.

Note: MAC is IXF1104 in IO-920

SYNTAX

```
mac[r/ra/w]
Example:
macra
macr
macw
```

phy

DESCRIPTION

Read all/write the PHY connected to the MAC through the RGMII

SYNTAX

phyra
phyw
Example:
phyra
phyw

I2Cread

DESCRIPTION

Reads from an I2C device and displays the results. The valid I2C addresses are a0, a2, a4, a8, ae and ac.

SYNTAX

i2cread <slave_addr> <offset_addr> <count>

PARAMETERS

- **slave_addr -** address of the I2C device on the bus
- **offset_addr -** memory address inside the I2C device
- **count** - number of bytes read (decimal format)

Example:
i2cread A2 0 10

I2cwrite

DESCRIPTION

Writes string data into the memory of an I2C device. The valid addresses are a0, a2, a4, a6, a8, ae and ac.

Caution **Note: The device is capable of a limited number of writes. Use only as needed.**



SYNTAX

i2cwrite <slave_addr> <offset_addr> <data>

PARAMETERS

- **slave_addr -** address of the I2C device on the bus
- **offset_addr -** memory address inside the I2C device
- **data** - hexadecimal string to write to I2C device memory

Example:
i2write a4 30 123456789

don/doff

DESCRIPTION

Enable /Disable data cache.

SYNTAX

don - Enable data cache
doff - Disable data cache
Example:
don
doff

ion/ioff

DESCRIPTION

Enable /Disable instruction cache.

SYNTAX

ion - Enable instruction cache
ioff - Disable instruction cache
Example:
ion
ioff

createscript

DESCRIPTION

Allows a test script to be written and stored in flash and run.

Note: To finish the script, just enter a null string.

SYNTAX

createscript
Example:
createscript

viewscript

DESCRIPTION

Displays the test script stored in the flash.

SYNTAX

viewscript
Example:
viewScript

runscript

DESCRIPTION

Executes the test script stored in the flash for the number of times.

SYNTAX

```
runscript loopcount
```

PARAMETERS

loopcount - number of times to execute the script

t mem

DESCRIPTION

Performs memory tests. The tests cover all types of memory and various test methods such as walking ones or walking zeros.

SYNTAX

```
t mem <option> <test_option> <start_addr> <size> <data> <loops> <verbose>
```

PARAMETERS

- **option -** indicates the memory type to test. Can be set as follows:
  - **s -** SRAM memory test channel 0 and 1 (default)
  - **s0** - SRAM memory test channel 0
  - **s1 -** SRAM memory test channel 1
  - **d -** DRAM memory test
  - **a -** all memory types
  - **h -** help
- **test_option -** sets the test method to use. Can be set as follows:
  - **w1 -** walking one test
  - **w0 -** walking zero test
  - **w -** both walking tests
  - **c5 -** checker 0x5A5A5A5A test
  - **ca -** checker 0xA5A5A5A5 test
  - **c -** both checker test
  - **u -** unique test
  - **b -** address bus test
  - **p -** pattern test (<x:data> is required)
  - **i -** incremental test
  - **e -** ECC test (only applicable to DRAM)
  - **m -** March B test
  - **r** - Random Read test
  - **a -** all tests

- **start_addr -** memory address where the test should begin (relative to 2 MB)
- **size -** number of bytes to test starting from start_addr
- **data -** data used for patter test (mandatory for the pattern (p) test)
- **loops -** specifies the number of times a test is performed (default value is 1)
- **verbose -** sets the verbose level for the test. This setting is only valid during the initial execution of a looped command.

Example:

```
t mem s w 0 1000
```

t uart

DESCRIPTION

Performs UART tests. The test cover all types of UART in the system. User intervention is required to finish the test.

SYNTAX

```
t uart <command>
```

PARAMETERS

command - indicates a test method to perform. Can be set as follows:

- **d -** dump all UART readable registers
- **r -** UART register test
- **q -** UART non-FIFO polling test
- **j -** UART non-FIFO interrupt test
- **p -** UART FIFO polling test
- **i -** UART FIFO interrupt test
- **a -** all UART tests
- **h -** help

Example:

```
t uart 1
```

t xscale

DESCRIPTION

Performs XScale scratchpad and core component tests. The tests cover IRQs and internal scratch memory tests.

SYNTAX

```
t xscale <option>
```

PARAMETERS

option - indicates a test option to perform. Can be set as follows:

- **i -** XScale interrupt test
- **s -** XScale scratchpad test

  - **p** - XScale scratchpad pattern test
  - **r** - XScale scratch ring test
  - **a** - all tests
- **a** - all tests
- **h** - help

Example:

```
t xscale s p
```

t gpio

DESCRIPTION

Performs GPIO tests that check presence, registers, edge detection and level detection.

SYNTAX

t gpio <option>

PARAMETERS

option - indicates the test to perform. Can be set as follows:

- **d** - GPIO detection test
- **r** - GPIO register test
- **e** - GPIO edge detection test
- **l** - GPIO level detection test
- **a** - all GPIO tests
- **h** - help

Example:

```
t gpio l
```

t timers

DESCRIPTION

Performs timer tests. The tests cover three timers and watchdog timers.

Note: Watchdog Timer test causes the system to reset.

SYNTAX

t timers <option>

PARAMETERS

option- indicates a test option to perform. Can be set as follows:

- **g** - Generic Timer test
- **w** - Watchdog timer test
- **a** - all tests
- **h** - help

Example:

```
t timers g
```

t i2c

DESCRIPTION

Performs IXP2400 I2C bus tests. The I2C bus is a software-emulated bus that uses two GPIO lines. The bus is used to read the eeprom in the IO-920.

SYNTAX

```
t i2c <option> <offset> <data>
```

PARAMETERS

- **option** - indicates the test to perform. Can be set as follows:
  - **w** - I2C write-only command
  - **p** - I2C page write test
  - **a** - all tests
  - **h** - help
- **offset** - offset in the I2C device (serial eeprom IO-920)
- **data** - data to write into I2C device (0-FF)

Example:

```
t i2c p  0
```

t ueng

DESCRIPTION

Performs a microengine (uEngine) test.

SYNTAX

```
t ueng <option> <microengine> <loop>
```

PARAMETERS

- **option**- indicates a test method to perform. Can be set as follows:
  - **s** - context tests
  - **r** - register test
  - **c** - control store test
  - **t** - timers and counters test
  - **a** - all uEngine tests
  - **h** - help
- **microengine** - specifies the uEngine ID to test. 0-3 and 16-19. Entering 99 will test all uEngines.
- **loop** - specifies how many times the test is performed (default is 1)

Note: Argument "Microengine #" is 0-3, 16-19 and all defaults are microengines.

Example:
`t ueng c 1`

t pci

DESCRIPTION

Performs PCI scan. The test displays all the PCI devices present.

SYNTAX

`t pci`

Example:
`t pci`

t sf

DESCRIPTION

Performs MSF Register test.

SYNTAX

`t sf`

Example:
`tsf`

t flash

DESCRIPTION

Performs flash lock test.

SYNTAX

`t <flash>`

Example:
`t flash`

t epld

DESCRIPTION

Performs epld Register tests.

SYNTAX

`t epld <option>`

PARAMETERS

- **`option`** - indicates the test to perform. Can be set as follows:
  - **`d`** - slowport register default test
  - **`r`** - slowport register write-read test
  - **`a`** - all slowport tests

– **h** - help

Example:

```
t epld d
```

t nic

DESCRIPTION

Performs onboard 82546 tests

SYNTAX

t nic <option>

PARAMETERS

- **option** - indicates the test to perform. Can be set as follows:
  - **p** - Ethernet NIC 82546 Presence Test
  - **r** - Ethernet NIC 82546 Register Test
  - **l** - Ethernet NIC 82546 Link Test
  - **i** - Ethernet NIC 82546 Internal Loopback Test
  - **h** - help

Example:

```
t nic p
```

t dma

DESCRIPTION

Performs the PCI DMA tests on both the channels

SYNTAX

t dma

Example:

```
t dma
```

t ipmi

DESCRIPTION

Detects the presence of IPMI controller.

SYNTAX

t ipmi

Example:

```
t ipmi
```

t spi3

DESCRIPTION

Performs the spi3 interface test between IXDP2400 and IXF1104.

SYNTAX

`t spi3`

Example:

```
t spi3
```

A

Troubleshooting

Error List

A typical CompactPCI system is highly sophisticated. This chapter can be taken as a hint list for detecting erroneous system configurations and strange behaviors.It cannot replace a serious and sophisticated pre- and post- sales support during application development.

If it is not possible to fix a problem with the help of this chapter, contact your local sales representative or FAE for further support.

After Power-On

Problem	Possible Reason	Solution
RedBoot prompt fails to appear after reset or Power-on	Failure of communication on Serial port	Check serial port settings on the console (Baud rate, Flow control, Stop bits, Start bits) Check proper cables.

During Board Operation

Problem	Possible Reason	Solution
OS does not boot	Boot device is not partitioned according to used operating system	Check partition according to operating system's needs
	Wrong configuration of boot devices	Configure boot devices correctly
Front panel reset does not work	Function is disabled	Ensure that position 1 of SW3 is OFF.
PPMC installed does not behave as a monarch	Function is disabled	Ensure that position 2 of SW3 is ON.
RTB does not work	RTB defect	Replace RTB
	RTB installed on wrong slot position	Install RTB on adjacent slot position of the used CPU board.
	RTB not defined for the used peripheral or system board	Install RTB defined for the used peripheral or system board.
Rear panel reset does not work	Function is disabled	Ensure that position 1 of SW2 on the RTB is OFF. (Refer to Table 7.)

Problem	Possible Reason	Solution
Operating system runs unstable	Drivers are missing, faulty or do not match hardware	1. Check that all used hardware parts have a driver matching the hardware 2. Reinstall hardware drivers
Board runs unstable	Disregard of environmental requirements	Ensure that temperature inside the system stays within specified ranges. Improve cooling inside the system if necessary. Ensure that other environmental requirements, such as moisture or altitude, are satisfied.
	Drivers are missing, faulty or do not match hardware.	Reinstall appropriate drivers.
	Board defect	Replace board

Index

Symbols

6U 33

A

Accessories 43
IXP/CPCI-9120/256/8-8/16/4GBE-T/R / 5E 43

B

Bezel 51

C

CompactPCI Connectors 67
J1 67
J2 67
J3 67
Connectors 66
control elements 57
CPCI-9120 ..... 33, 58
full hot swap 57
Installation in a Non-Powered System ..54

D

Diagnostics
Board Commands 146
Flashing 159

E

Ejector/injector levers 55, 58
elements 57
EMC 23
ESD 54

F

Front Panel 63

I

I2C bus 86
IBMU 86
Injector/ejector levers 55, 58
Installation
CPCI-9120 54
CPCI-9120, hot swap 58
non-powered system 54
Unpack Kit 42
Installing PMC 50
IO-920 47
IO-920 Variants 92

K

Keys 66
Front panel Reset 66

L

LA-1 Mezzanine connector 77
LA-1/QDR mezzanine
LA1-920/8M 93
LED
hot swap 57

M

Maps 95
I2C memory map 101
NPU I2C memory map 101
PCI memory map 97
SENSE I2C memory map 102
BIB I2C Memory Map 102
IO-920 MAC-SFP I2C memory map 102
SlowPort map 98
Media Switch Fabric mezzanine 91
IO-920 91
MSF Mezzanine Connector 69

N

Non- Powered System
removal 57
Non-Powered System
Install 54
NPU I2C memory map 101

O

order numbers 36
Ordering Information 36, 37

P

Part Number
111013 43
PCI bus 85
PCI stacking adapter 93
Peripheral Slot 54
Product Nomenclature 36
Programmable devices 86
PrPMC Connectors 68

R

Register 103
Board Revision Number 104
General Purpose EPLD 103
Board Revision number 103
Interrupt Mask Register-1 107
Interrupt Mask Register-2 109
Interrupt Mask Register-3 110
Interrupt Status Register-1 110
Interrupt Status Register-2 111
Interrupt Status Register-3 112
PrPMC register 105
System specific register 105
User Flash Select 104
User LED Control register 106
Variant ID 103
Version Number 103
Watchdog Timer Register 107
GeneralPurpose EPLD
CompactPCI Register 104
MAC EPLD - IO 920
Board Revision Number 113
Interrupt Mask Register 115
Interrupt Status Register 116
MAC PAUSE Address Register 115
MAC PAUSE Control Register 115
PHY Reset Control Register 114
SFP Control Register 114
Variant ID 113
Version Number 113
MAC EPLD-IO-920 112
Removal
CPCI-9120, hot swap 59
PMC 53
Remove
CPCI-9120, non-powered 57
removing 59
RJ-45 Connector 25

S

SlowPort map 97
System Slot 54

W

Watchdog Timer Interrupt Clear Register .107

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