MVME2400 Series
VME Processor Module

Installation and Use

V2400A/IH3

August 2001

© Copyright 1999, 2000, 2001 Motorola, Inc.

All rights reserved.

Printed in the United States of America.

Motorola® and the Motorola symbol are registered trademarks of Motorola, Inc.

PowerStack™, VMEmodule™, and VMEsystem™ are trademarks of Motorola, Inc.

PowerPC® is a registered trademark and AIX™, PowerPC 603™, and PowerPC 604™ are trademarks of International Business Machines Corporation and are used by Motorola, Inc. under license from International Business Machines Corporation.

SNAPHAT®, TIMEKEEPER®, and ZEROPOWER® are registered trademarks of STMicroelectronics.

All other products mentioned in this document are trademarks or registered trademarks of their respective holders.

Safety Summary

The following general safety precautions must be observed during all phases of operation, service, and repair of this equipment. Failure to comply with these precautions or with specific warnings elsewhere in this manual could result in personal injury or damage to the equipment.

The safety precautions listed below represent warnings of certain dangers of which Motorola is aware. You, as the user of the product, should follow these warnings and all other safety precautions necessary for the safe operation of the equipment in your operating environment.

Ground the Instrument.

To minimize shock hazard, the equipment chassis and enclosure must be connected to an electrical ground. If the equipment is supplied with a three-conductor AC power cable, the power cable must be plugged into an approved three-contact electrical outlet, with the grounding wire (green/yellow) reliably connected to an electrical ground (safety ground) at the power outlet. The power jack and mating plug of the power cable meet International Electrotechnical Commission (IEC) safety standards and local electrical regulatory codes.

Do Not Operate in an Explosive Atmosphere.

Do not operate the equipment in any explosive atmosphere such as in the presence of flammable gases or fumes. Operation of any electrical equipment in such an environment could result in an explosion and cause injury or damage.

Keep Away From Live Circuits Inside the Equipment.

Operating personnel must not remove equipment covers. Only Factory Authorized Service Personnel or other qualified service personnel may remove equipment covers for internal subassembly or component replacement or any internal adjustment. Service personnel should not replace components with power cable connected. Under certain conditions, dangerous voltages may exist even with the power cable removed. To avoid injuries, such personnel should always disconnect power and discharge circuits before touching components.

Use Caution When Exposing or Handling a CRT.

Breakage of a Cathode-Ray Tube (CRT) causes a high-velocity scattering of glass fragments (implosion). To prevent CRT implosion, do not handle the CRT and avoid rough handling or jarring of the equipment. Handling of a CRT should be done only by qualified service personnel using approved safety mask and gloves.

Do Not Substitute Parts or Modify Equipment.

Do not install substitute parts or perform any unauthorized modification of the equipment. Contact your local Motorola representative for service and repair to ensure that all safety features are maintained.

Observe Warnings in Manual.

Warnings, such as the example below, precede potentially dangerous procedures throughout this manual. Instructions contained in the warnings must be followed. You should also employ all other safety precautions which you deem necessary for the operation of the equipment in your operating environment.



Warning

To prevent serious injury or death from dangerous voltages, use extreme caution when handling, testing, and adjusting this equipment and its components.

Flammability

All Motorola PWBs (printed wiring boards) are manufactured with a flammability rating of 94V-0 by UL-recognized manufacturers.

EMI Caution



Caution

This equipment generates, uses and can radiate electromagnetic energy. It may cause or be susceptible to electromagnetic interference (EMI) if not installed and used with adequate EMI protection.

Lithium Battery Caution

This product contains a lithium battery to power the clock and calendar circuitry.



Caution

Danger of explosion if battery is replaced incorrectly. Replace battery only with the same or equivalent type recommended by the equipment manufacturer. Dispose of used batteries according to the manufacturer's instructions.



Attention

Il y a danger d'explosion s'il y a remplacement incorrect de la batterie. Remplacer uniquement avec une batterie du même type ou d'un type équivalent recommandé par le constructeur. Mettre au rebut les batteries usagées conformément aux instructions du fabricant.



Vorsicht

Explosionsgefahr bei unsachgemäßem Austausch der Batterie. Ersatznur durch denselben oder einen vom Hersteller empfohlenen Typ. Entsorgung gebrauchter Batterien nach Angaben des Herstellers.

CE Notice (European Community)

Motorola Computer Group products with the CE marking comply with the EMC Directive (89/336/EEC). Compliance with this directive implies conformity to the following European Norms:

> EN55022 “Limits and Methods of Measurement of Radio Interference Characteristics of Information Technology Equipment”; this product tested to Equipment Class B
>
> EN55024 “Information technology equipment—Immunity characteristics—Limits and methods of measurement”

Board products are tested in a representative system to show compliance with the above mentioned requirements. A proper installation in a CE-marked system will maintain the required EMC performance.

In accordance with European Community directives, a “Declaration of Conformity” has been made and is available on request. Please contact your sales representative.

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 Computer Group web site. 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.
Computer Group
2900 South Diablo Way
Tempe, Arizona 85282

Contents

About This Manual

Summary of Changes xvi
Overview of Contents xvi
Comments and Suggestions xvii
Conventions Used in This Manual xviii

CHAPTER 1 Hardware Preparation and Installation

Introduction 1-1
Description 1-1
- MVME240x Module 1-2
- PMCspan Expansion Mezzanine 1-2
- PCI Mezzanine Cards (PMCs) 1-3
- VMEsystem Enclosure 1-3
- System Console Terminal 1-3

Overview of Start-Up Procedures 1-4
Unpacking Instructions 1-6
Hardware Configuration 1-6
MVME2400 Base Board Preparation 1-7
- Flash Bank Selection (J8) 1-9
- System Controller Selection (J9) 1-9
- Software-Readable Header (SRH) Switch (S3) 1-10

PMC Preparation 1-11
PMCspan Preparation 1-11
System Console Terminal Preparation 1-11
Hardware Installation 1-12
- PMC Module Installation 1-12
- Primary PMCspan Installation 1-14
- Secondary PMCspan Installation 1-17
- MVME240x Installation 1-20

System Considerations 1-22

CHAPTER 2 Operating Instructions

Overview 2-1
Applying Power 2-1

Switches 2-3
ABT (S1) 2-3
RST (S2) 2-3
Front Panel Indicators (DS1 – DS4) 2-4
BFL (DS1) 2-4
CPU (DS2) 2-4
PMC2 (DS3) 2-4
PMC1 (DS4) 2-4
10/100BaseT Port 2-5
DEBUG Port 2-5
PMC Slots 2-7
PCI MEZZANINE CARD (PMC Slot 1) 2-7
PCI MEZZANINE CARD (PMC Slot 2) 2-7
PMCspan 2-8

CHAPTER 3 Functional Description

Introduction 3-1
Features 3-1
General Description 3-3
Block Diagram 3-3
MPC750 Processor 3-5
L2 Cache 3-5
Hawk System Memory Controller (SMC)/PCI Host Bridge (PHB) ASIC 3-6
PCI Bus Latency 3-7
PPC Bus Latency 3-9
Assumptions 3-11
Clock Ratios and Operating Frequencies 3-11
PPC60x Originated 3-11
PCI Originated 3-12
SDRAM Memory 3-12
SDRAM Latency 3-13
Flash Memory 3-16
ROM/Flash Performance 3-17
Ethernet Interface 3-19
PCI Mezzanine Card (PMC) Interface 3-20
PMC Slot 1 (Single-Width PMC) 3-21
PMC Slot 2 (Single-Width PMC) 3-21
PMC Slots 1 and 2 (Double-Width PMC) 3-22
PCI Expansion 3-22
VMEbus Interface 3-22

Asynchronous Debug Port 3-23
PCI-ISA Bridge (PIB) Controller 3-23
Real-Time Clock/NVRAM/Timer Function 3-24
PCI Host Bridge (PHB) 3-25
Interrupt Controller (MPIC) 3-25
Programmable Timers 3-26
Interval Timers 3-26
16/32-Bit Timers 3-26

CHAPTER 4 Programming Details

Introduction 4-1
Memory Maps 4-1
Processor Bus Memory Map 4-2
Default Processor Memory Map 4-2
PCI Local Bus Memory Map 4-3
VMEbus Memory Map 4-3
Programming Considerations 4-4
PCI Arbitration 4-4
Interrupt Handling 4-6
DMA Channels 4-8
Sources of Reset 4-8
Endian Issues 4-10
Processor/Memory Domain 4-10
PCI Domain 4-10
VMEbus Domain 4-11

CHAPTER 5 PPCBug

PPCBug Overview 5-1
PPCBug Basics 5-1
Memory Requirements 5-3
PPCBug Implementation 5-3
MPU, Hardware, and Firmware Initialization 5-3
Using PPCBug 5-5
Debugger Commands 5-6
Diagnostic Tests 5-11

CHAPTER 6 Environment Modification

Overview 6-1

CNFG – Configure Board Information Block 6-2
ENV – Set Environment 6-3
Configuring the PPCBug Parameters 6-3
Configuring the VMEbus Interface 6-13

APPENDIX A Specifications

Specifications A-1
Cooling Requirements A-3
EMC Regulatory Compliance A-4

APPENDIX B Connector Pin Assignments

Introduction B-1
Pin Assignments B-1
VMEbus Connector – P1 ..... B-2
VMEbus Connector – P2 ..... B-3
Serial Port Connector – DEBUG (J2) B-5
Ethernet Connector – 10BaseT (J3) B-5
CPU Debug Connector – J1 ..... B-6
PCI Expansion Connector – J6 ..... B-11
PCI Mezzanine Card Connectors – J11 through J14 ..... B-14
PCI Mezzanine Card Connectors – J21 through J24 ..... B-17

APPENDIX C Troubleshooting

Solving Startup Problems C-1

APPENDIX D Related Documentation

Motorola Computer Group Documents D-1
Manufacturers' Documents D-2
Related Specifications D-4

List of Figures

Figure 1-1. MVME2400 Switches, Headers, Connectors, Fuses, LEDs 1-8
Figure 1-2. Software-Readable Header.....1-10
Figure 1-3. Typical Single-width PMC Module Placement on MVME240x 1-14
Figure 1-4. PMCspan-002 Installation on an MVME240x 1-16
Figure 1-5. PMCspan-010 Installation on a PMCspan-002/MVME240x 1-18
Figure 2-1. PPCBug Firmware System Startup 2-2
Figure 2-2. MVME240x DEBUG Port Configuration 2-6
Figure 3-1. MVME240x Block Diagram.....3-4
Figure 3-2. Timing Definitions for PPC Bus to SDRAM Access.....3-15
Figure 4-1. VMEbus Master Mapping.....4-5
Figure 4-2. MVME240x Interrupt Architecture 4-7

List of Tables

Table 1-1. PMCspan Models 1-3
Table 1-2. Start-Up Overview 1-4
Table 1-3. Jumper Settings 1-7
Table 3-1. MVME240x Features 3-1
Table 3-2. Power Requirements 3-5
Table 3-3. PCI Originated Latency Matrix 3-7
Table 3-4. PCI Originated Bandwidth Matrix 3-8
Table 3-5. PPC60x Originated Latency Matrix 3-9
Table 3-6. PPC60x Originated Bandwidth Matrix 3-10
Table 3-7. Clock Ratios and Operating Frequencies 3-11
Table 3-8. 60x Bus to SDRAM Access Timing (100 MHz/PC100 SDRAMs) 3-13
Table 3-9. PPC Bus to ROM/Flash Access Timing (120ns @ 100 MHz) 3-17
Table 3-10. PPC Bus to ROM/Flash Access Timing (80ns @ 100 MHz) 3-18
Table 3-11. PPC Bus to ROM/Flash Access Timing (50ns @ 100MHz) 3-18
Table 3-12. PPC Bus to ROM/Flash Access Timing (30ns @ 100 MHz) 3-19
Table 4-1. Processor Default View of the Memory Map 4-2
Table 4-2. PCI Arbitration Assignments 4-6
Table 4-3. Classes of Reset and Effectiveness 4-9
Table 5-1. Debugger Commands 5-7
Table 5-2. Diagnostic Test Groups 5-12
Table A-1. Specifications A-1
Table B-1. P1 VMEbus Connector Pin Assignments B-2
Table B-2. P2 Connector Pin Assignment B-3
Table B-3. DEBUG (J2)Connector Pin Assignments B-5
Table B-4. 10/100 BASET (J3) Connector Pin Assignments B-5
Table B-5. Debug Connector Pin Assignments B-6
Table B-6. J6 - PCI Expansion Connector Pin Assignments B-11
Table B-7. J11 - J12 PMC1 Connector Pin Assignments B-14
Table B-8. J13 - J14 PMC1 Connector Pin Assignments B-15
Table B-9. J21 and J22 PMC2 Connector Pin Assignments B-17
Table B-10. J23 and J24 PMC2 Connector Pin Assignments B-18
Table C-1. Troubleshooting MVME240x Modules C-1
Table D-1. Motorola Computer Group Documents D-1
Table D-2. Manufacturers' Documents D-2
Table D-3. Related Specifications D-4

About This Manual

The *MVME2400 Series VME Processor Modules Installation and Use* manual provides information to install and use your MVME2400 Series VME Processor Modules.

As of the publication date, the information presented in this manual applies to the following MVME2400 series models:

Model Number	Description
MVME2401-1	233 MHz MPC750, 32MB ECC SDRAM
MVME2401-3	233 MHz MPC750, 64MB ECC SDRAM
MVME2403-1	233 MHz MPC750, 32MB ECC SDRAM
MVME2403-3	233 MHz MPC750, 32MB ECC SDRAM
MVME2431-1	350 MHz MPC750, 32MB ECC SDRAM
MVME2431-3	350 MHz MPC750, 32MB ECC SDRAM
MVME2432-1	350 MHz MPC750, 64MB ECC SDRAM
MVME2432-3	350 MHz MPC750, 64MB ECC SDRAM
MVME2433-1	350 MHz MPC750, 128MB ECC SDRAM
MVME2433-3	350 MHz MPC750, 128MB ECC SDRAM
MVME2434-1	350 MHz MPC750, 256MB ECC SDRAM
MVME2434-3	350 MHz MPC750, 256MB ECC SDRAM
MVME2400-0321	450 MHz MPC750, 32MB ECC SDRAM
MVME2400-0323	450 MHz MPC750, 32MB ECC SDRAM
MVME2400-0331	450 MHz MPC750, 64MB ECC SDRAM
MVME2400-0333	450 MHz MPC750, 64MB ECC SDRAM
MVME2400-0341	450 MHz MPC750, 128MB ECC SDRAM
MVME2400-0343	450 MHz MPC750, 128MB ECC SDRAM
MVME2400-0351	450 MHz MPC750, 256MB ECC SDRAM
MVME2400-0353	450 MHz MPC750, 256MB ECC SDRAM
MVME2400-0361	450 MHz MPC750, 512MB ECC SDRAM
MVME2400-0363	450 MHz MPC750, 512MB ECC SDRAM

Summary of Changes

This is the third edition of the *Installation and Use* manual. It supersedes the March 2000 edition and incorporates the following updates.

Date	Changes
August 2001	All data referring to the VME CSR Bit Set Register (VCSR_SET) and VME CSR Bit Clear Register (VCSR_CLR) has been deleted. These registers of the Universe II are unavailable for implementation as intended by the MVME materials and the Universe II User Manual.
March 2000	Addition of the 450 MHz product configurations and a general review of the manual's accuracy and content.

Overview of Contents

Chapter 1, *Hardware Preparation and Installation*, provides a brief description of the MVME2400 Series VME Processor Module along with instructions for preparing and installing the hardware.

Chapter 2, *Operating Instructions*, provides operating instructions for the MVME2400, including information about powering up the system, and functionality of the switches, status indicators, and I/O ports on the front panels of the MVME2400 and PMCspan modules.

Chapter 3, *Functional Description*, provides a functional description of the MVME2400, including an overview of the product and a detailed description of several blocks of circuitry.

Chapter 4, *Programming Details*, provides information useful in programming the MVME2400, including a description of memory maps, control and status registers, PCI arbitration, interrupt handling, sources of reset, and big/little-endian issues.

Chapter 5, *PPCBug*, describes the basics of the PPCBug and its architecture, along with the monitor (interactive command portion of the firmware), and gives information on using the PPCBug and the special commands.

Chapter 6, *Environment Modification*, contains information about the **CNFG** and **ENV** commands. These two commands are used to change configuration information and command parameters interactively.

Appendix A, *Specifications*, lists the general specifications for the MVME2400 VME processor module.

Appendix B, *Connector Pin Assignments*, provides the pin assignments for the interconnect signals for the MVME2400.

Appendix C, *Troubleshooting*, provides simple troubleshooting tips for your MVME2400 VME Processor Modules.

Appendix D, *Related Documentation*, includes all documentation related to the MVME2400.

Comments and Suggestions

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

Motorola Computer Group
Reader Comments DW164
2900 S. Diablo Way
Tempe, Arizona 85282

You can also submit comments to the following e-mail address:
reader-comments@mcg.mot.com

In all your correspondence, please list your name, position, and company. Be sure to include the title and part number of the manual and tell how you used it. Then tell us your feelings about its strengths and weaknesses and any recommendations for improvements.

Conventions Used in This Manual

The following typographical conventions are used in this document:

Unless otherwise specified, all address references are in hexadecimal. An asterisk (*) following the signal name for signals which are level significant denotes that the signal is true or valid when the signal is low. An asterisk (*) following the signal name for signals which are edge significant denotes that the actions initiated by that signal occur on high to low transition.

$	dollar	specifies a hexadecimal number
&	ampersand	specifies a decimal number
%	percent	specifies a binary number

bold

is used for user input that you type just as it appears; it is also used for commands, options and arguments to commands, and names of programs, directories and files.

italic

is used for names of variables to which you assign values. Italic is also used for comments in screen displays and examples, and to introduce new terms.

`courier`

is used for system output (for example, screen displays, reports), examples, and system prompts.

<**Enter**>, <**Return**> or <**CR**>

<**CR**> represents the carriage return or Enter key.

CTRL

represents the Control key. Execute control characters by pressing the Ctrl key and the letter simultaneously, for example, **Ctrl-d**.

In this manual, *assertion* and *negation* are used to specify forcing a signal to a particular state. In particular, *assertion* and *assert* refer to a signal that is active or true; *negation* and *negate* indicate a signal that is inactive or false. These terms are used independently of the voltage level (high or low) that they represent.

Data and address sizes are defined as follows:

- A *byte* is eight bits, numbered 0 through 7, with bit 0 being the least significant.
- A *word* is 16 bits, numbered 0 through 15, with bit 0 being the least significant.
- A *longword* is 32 bits, numbered 0 through 31, with bit 0 being the least significant.

The terms *control bit, status bit, true,* and *false* are used extensively in this document. The term *control bit* is used to describe a bit in a register that can be set and cleared under software control. The term *true* is used to indicate that a bit is in the state that enables the function it controls. The term *false* is used to indicate that the bit is in the state that disables the function it controls. In all tables, the terms *0* and *1* are used to describe the actual value that should be written to the bit, or the value that it yields when read. The term *status bit* is used to describe a bit in a register that reflects a specific condition. The status bit can be read by software to determine operational or exception conditions.

Hardware Preparation and Installation

1

Introduction

This chapter provides a brief description of the MVME2400 Series VME Processor Module. It also provides instructions for preparing and installing the hardware.Unless otherwise specified, the designation “MVME240x” refers to all models of the MVME2400 series modules.

Description

The MVME240x is a PCI Mezzanine Card (PMC) carrier board. It is based on the PowerPC™ 750 microprocessor, MPC750.

Two front panel cutouts provide access to PMC I/O. One double-width or two single-width PMCs can be installed directly on the MVME240x. Optionally, one or two PMCspan PCI expansion mezzanine modules can be added to provide the capability of up to four additional PMC modules.

Two RJ-45 connectors on the front panel provide the interface to 10/100BaseT Ethernet, and to a debug serial port.

The following list is of equipment that is appropriate for use in an MVME240x system:

- PMCspan PCI expansion mezzanine module
- Peripheral Component Interconnect (PCI) Mezzanine Cards (PMC)s
- VMEsystem enclosure
- System console terminal
- Disk drives (and/or other I/O) and controllers
- Operating system (and/or application software)

MVME240x Module

The MVME240x is a powerful, low-cost embedded VME controller and intelligent PMC carrier board. It includes support circuitry such as ECC SDRAM, PROM/Flash memory, and bridges to the Industry Standard Architecture (ISA) bus and the VMEbus. The unit's PMC carrier architecture allows flexible configuration options and easy upgrades. It is also designed to support one or two PMCs, plus one or two optional PCI expansion mezzanine modules that each support up to two PMCs. The unit occupies a single VMEmodule slot (except when optional PCI expansion mezzanine modules are also used).

The MVME240x interfaces to the VMEbus via the P1 and P2 connectors, which use the new 5-row 160-pin connectors as specified in the proposed VME64 Extension Standard. It also draws +5V, +12V, and –12V power from the VMEbus backplane through these two connectors. The +3.3V and 2.5V power, used for the PCI bridge chip and possibly for the PMC mezzanine, is derived onboard from the +5V power.

Support for two IEEE P1386.1 PCI mezzanine cards is provided via eight 64-pin SMT connectors. Front panel openings are provided on the board for the two PMC slots.

In addition, there are 64 pins of I/O from PMC slot 1 and 46 pins of I/O from PMC slot 2 that are routed to P2. The two PMC slots may contain two single-wide PMCs or one double-wide PMC. There are also two RJ-45 connectors on the front panel: one for the Ethernet 10BaseT/100BaseTX interface, and one for the async serial debug port. The front panel also includes reset and abort switches and status LEDs.

PMCspan Expansion Mezzanine

An optional PCI expansion mezzanine module or PMC carrier board, PMCspan, provides the capability of adding two additional PMCs. Two PMCspans can be stacked on an MVME240x, providing four additional

PMC slots, for a total of six slots including the two onboard the MVME240x. The following table lists the PMCspan models that are available for use with the MVME240x.

Table 1-1. PMCspan Models

Expansion Module	Description
PMCSPAN-002	Primary PCI expansion mezzanine module. Allows two PMC modules for the MVME240x. Includes 32-bit PCI bridge.
PMCSPAN-010	Secondary PCI expansion mezzanine module. Allows two additional PMC modules for the MVME240x. Does not include 32-bit PCI bridge; requires a PMCSPAN-002.

PCI Mezzanine Cards (PMCs)

The PMC slots on the MVME240x board are IEEE P1386.1 compliant. P2 I/O-based PMCs that follow the PMC committee recommendation for PCI I/O when using the 5-row VME64 extension connector will be pin-out compatible with the MVME240x.

The MVME240x board supports both front panel I/O and rear panel P2 I/O through either PMC slot 1 or PMC slot 2. 64 pins of I/O from slot 1 and 46 pins of I/O from slot 2 are routed directly to P2.

VMEsystem Enclosure

Your MVME240x board must be installed in a VMEsystem chassis with both P1 and P2 backplane connections. It requires a single slot, except when PMCspan carrier boards are used. Allow one extra slot for each PMCspan.

System Console Terminal

In normal operation, connection of a debug console terminal is required only if you intend to use the MVME240x's debug firmware, PPCBug, interactively. An RJ-45 connector is provided on the front panel of the MVME240x for this purpose.

Overview of Start-Up Procedures

The following table lists the things you will need to do before you can use this board, and tells where to find the information you need to perform each step. Read this chapter in its entirety along with all Caution and Warning notes before beginning.

Table 1-2. Start-Up Overview

What you need to do...	Refer to...
Unpack the hardware.	*Unpacking Instructions* on page 1-6
Set jumpers on the MVME240x module.	*Hardware Configuration* on page 1-6
Prepare the PMCs.	*PMC Preparation* on page 1-11
	For additional information on PMCs, refer to the PMC manuals provided with these cards.
Prepare the PMCspan module(s).	*PMCspan Preparation* on page 1-11
	For additional information on PMCspan, refer to the PMCspan *PMC Adapter Carrier Module Installation and Use manual*, listed in Appendix D, *Related Documentation*.
Prepare any other optional devices or equipment you'll be using.	For more information on optional devices and equipment, refer to the documentation provided with that equipment.
Install the PMCs on the MVME240x module.	*PMC Module Installation* on page 1-12
	PMC Slots on page 2-7
	For additional information on PMCs, refer to the PMC manuals provided with these cards.
Install the primary PMCspan module (if used).	*Primary PMCspan Installation* on page 1-14
	For additional information on PMCspan, refer to the PMCspan *PMC Adapter Carrier Module Installation and Use manual*, listed in Appendix D, *Related Documentation*.
Install the secondary PMCspan module (if used) (Cont'd).	*Secondary PMCspan Installation* on page 1-17
	For additional information on PMCspan, refer to the PMCspan *PMC Adapter Carrier Module Installation and Use manual*, listed in Appendix D, *Related Documentation*.

Table 1-2. Start-Up Overview (Continued)

What you need to do...	Refer to...
Install the MVME240x module in a chassis.	*MVME240x Installation* on page 1-20
Connect a console terminal.	*System Considerations* on page 1-22
	DEBUG Port on page 2-5
Connect any other optional devices or equipment you will be using.	Appendix B, *Connector Pin Assignments*
	For more information on optional devices and equipment, refer to the documentation provided with that equipment.
Power up the system.	*Applying Power* on page 2-1
	Front Panel Indicators (DS1 – DS4) on page 2-4
	If any problems occur, refer to *Diagnostic Tests* on page 5-11.
	You may also wish to obtain the *PPCBug Diagnostics Manual*, listed in Appendix D, *Related Documentation*.
Examine the environmental parameters and make any changes needed.	*ENV – Set Environment* on page 6-3
	You may also wish to obtain the *PPCBug Firmware Package User's Manual*, listed in Appendix D, *Related Documentation*.
Program the MVME240x module and PMCs as needed for your applications.	*MVME2400 Base Board Preparation* on page 1-7
	Chapter 4, *Programming Details*
	For additional information on PMCs, refer to the PMC manuals provided with these cards.
	You may also wish to obtain the *MVME2400 Series VME Processor Module Programmer's Reference Guide*, listed in Appendix D, *Related Documentation*.

Unpacking Instructions

Note If the shipping carton is damaged upon receipt, request that the carrier's agent be present during the unpacking and inspection of the equipment.

Unpack the equipment from the shipping carton. Refer to the packing list and verify that all items are present. Save the packing material for storing and reshipping of equipment.



Caution Avoid touching areas of integrated circuitry; static discharge can damage these circuits.

Hardware Configuration

To produce the desired configuration and ensure proper operation of the MVME2400, you may need to carry out certain hardware modifications before installing the module.

The MVME2400 provides software control over most options: by setting bits in control registers after installing the module in a system, you can modify its configuration. The MVME2400 control registers are described in Chapter 3, *Functional Description*, and/or in the *MVME2400 Series VME Processor Module Programmer's Reference Guide* listed under Appendix D, *Related Documentation*.

Some options, however, are not software-programmable. Such options are controlled through manual installation or removal of header jumpers or interface modules on the base board.

MVME2400 Base Board Preparation

Figure 1-1 illustrates the placement of the switches, jumper headers, connectors, and LED indicators on the MVME240x. Manually configurable items on the base board are listed in the following table. Refer to the sections or figures listed along side the jumper function for more information.

Table 1-3. Jumper Settings

Jumper	Function
J8	*Flash Bank Selection (J8)* on page 1-9
J9	*System Controller Selection (J9)* on page 1-9
S3	*Software-Readable Header (SRH) Switch (S3)* on page 1-10

The MVME240x has been factory tested and is shipped with the configurations described in the following sections. The MVME240x factory-installed debug monitor, PPCBug, operates with those factory settings.

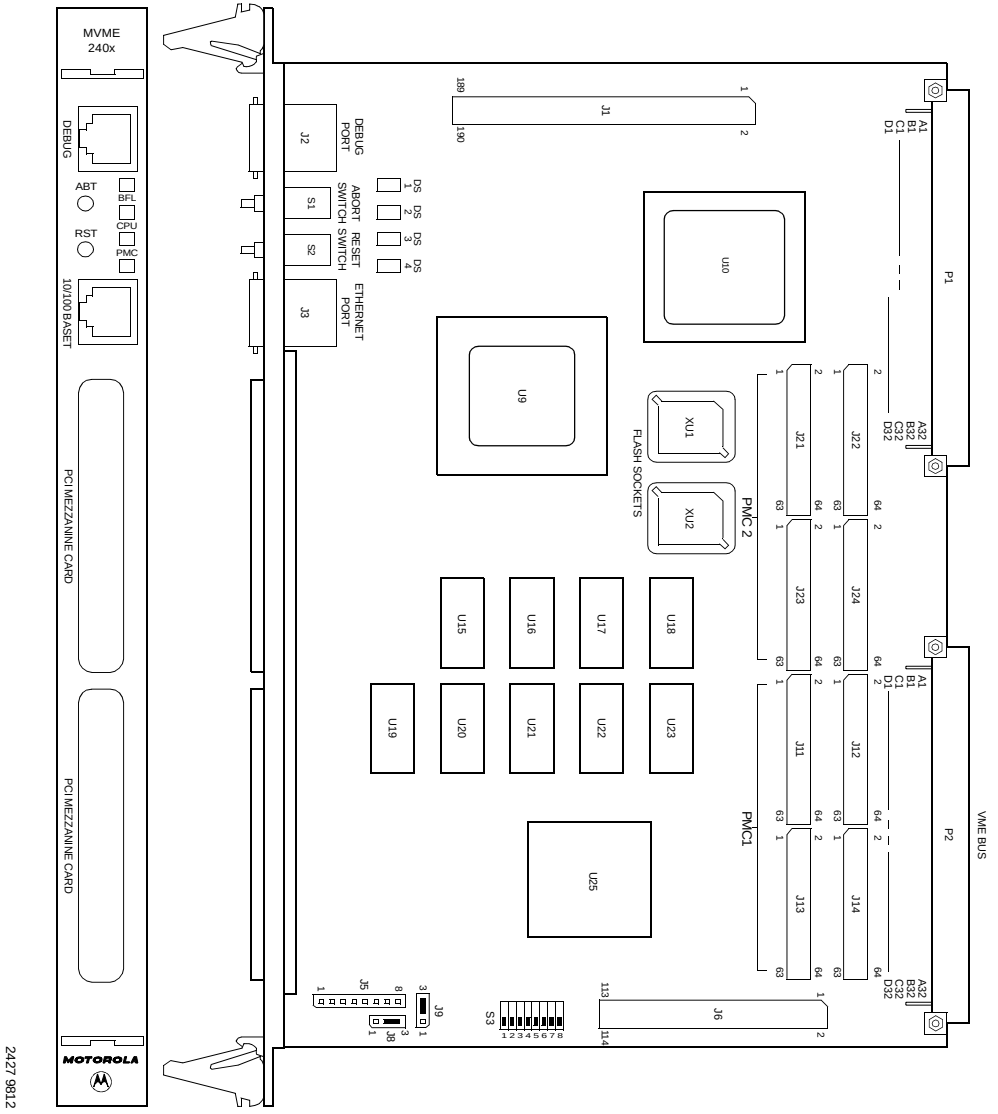
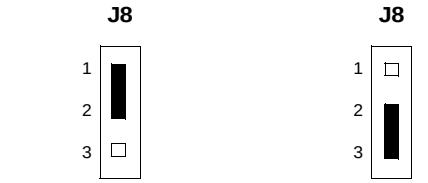


Figure 1-1. MVME2400 Switches, Headers, Connectors, Fuses, LEDs

Flash Bank Selection (J8)

Bank B consists of 1MB of 8-bit Flash memory in two 32-pin PLCC 8-bit sockets.

Bank A consists of four 16-bit devices that are populated with 16Mbit Flash devices (8MB). A jumper header, J8, associated with the first set of four Flash devices provides a total of 64KB of hardware-protected boot block. Only 32-bit writes are supported for this bank of Flash. The address of the reset vector is jumper-selectable. A jumper must be installed either between J8 pins 1 and 2 for Bank A factory configuration, or between J8 pins 2 and 3 for Bank B. When the jumper is installed, the SMC (System Memory Controller) of the Hawk ASIC maps 0xFFF00100 to the Bank B sockets.



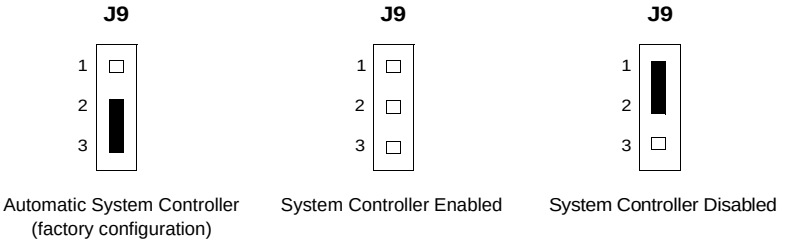
Bank A (factory configuration) Bank B

System Controller Selection (J9)

The MVME240x is factory-configured in automatic system controller mode; that is, a jumper is installed across pins 2 and 3 of header J9. This means that the MVME240x determines if it is system controller at system power-up or reset by its position on the bus; if it is in slot 1 on the VME system, it configures itself as the system controller.

Remove the jumper from J9 if you intend to operate the MVME240x as system controller in all cases.

Install the jumper across pins 1 and 2 if the MVME240x will not to operate as system controller under any circumstances.



Software-Readable Header (SRH) Switch (S3)

Switch S3 is an eight pole single-throw switch with software readable switch settings. These settings can be read as a register at ISA I/O address $801 (hexadecimal). Each switch pole can be set to either logic 0 or logic 1. A logic 0 means the switch is in the “ON” position for that particular bit. A logic 1 means the switch is in the “OFF” position for that particular bit. SRH Register Bit 0 is associated with Pin 1 and Pin 16 of the SRH, and SRH Register Bit 7 is associated with Pin 8 and Pin 9 of the SRH. The SRH is a read-only register.

If Motorola’s PowerPC firmware, PPCBug, is being used, it reserves all bits, SRH0 to SRH7. If it is not being used, the switch can be used for other applications.

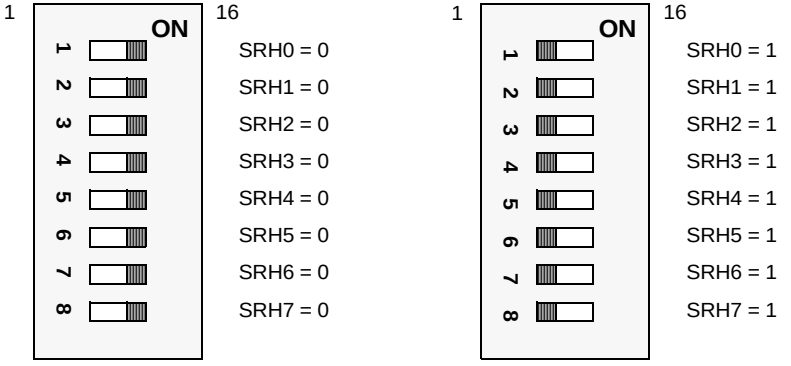


Figure 1-2. Software-Readable Header

PMC Preparation

For a discussion of any configurable items on the PMCs, refer to the user's manual for the particular PMCs.

PMCspan Preparation

You will need to use an additional slot in the VME chassis for each PMCspan expansion module you plan to use. Before installing a PMCspan on the MVME240x, you must install the selected PMCs on the PMCspan. Refer to the PMCspan *PMCAdapter Carrier Module Installation and Use* manual for instructions.

System Console Terminal Preparation

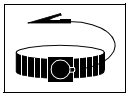
Ensure that the switches are set in the proper position for all bits on switch S3 of the MVME240x board as shown in Figure 1-2. This is necessary when the PPCBug firmware is used. Connect the terminal via a cable to the RJ-45 DEBUG connector J2. See Table B-3 on page B-5 for pin signal assignments. Set up the terminal as follows:

- Eight bits per character
- One stop bit per character
- Parity disabled (no parity)
- Baud rate = 9600 baud (default baud rate of the port at power-up); after power-up, you can reconfigure the baud rate with PPCBug's **PF** command

Hardware Installation

The following paragraphs discuss installing PMCs onto the MVME240x, installing PMCspan modules onto the MVME240x, installing the MVME240x into a VME chassis, and connecting an optional system console terminal.

Use ESD



Wrist Strap

Motorola strongly recommends that you use an antistatic wrist strap and a conductive foam pad when installing or upgrading a system. Electronic components, such as disk drives, computer boards, and memory modules, can be extremely sensitive to electrostatic discharge (ESD). After removing the component from its protective wrapper or from the system, place the component flat on a grounded, static-free surface (and, in the case of a board, component side up). Do not slide the component over any surface.

If an ESD station is not available, you can avoid damage resulting from ESD by wearing an antistatic wrist strap (available at electronics stores) that is attached to an active electrical ground. Note that a system chassis may not be grounded if it is unplugged.

PMC Module Installation

PCI mezzanine card (PMC) modules mount on top of the MVME240x module, and/or on a PMCspan. Refer to Figure 1-3 on page 1-14 and perform the following steps to install a PMC on your MVME240x module.

Note This procedure assumes that you have read the user's manual that came with your PMCs.

1. Attach an ESD strap to your wrist. Attach the other end of the ESD strap to the chassis as a ground. The ESD strap must be secured to your wrist and to ground throughout the procedure.
2. Perform an operating system shutdown. Turn the AC or DC power off and remove the AC cord or DC power lines from the system. Remove chassis or system cover(s) as necessary for access to the VME modules.



Inserting or removing modules with power applied may result in damage to module components.



Dangerous voltages, capable of causing death, are present in this equipment. Use extreme caution when handling, testing, and adjusting.

3. Carefully remove the MVME2400 from its VMEbus card slot and lay it flat, with connectors P1 and P2 facing you.



Avoid touching areas of integrated circuitry; static discharge can damage these circuits.

4. Remove the PCI filler plate from the selected PMC slot in the front panel of the MVME240x. If installing a double-width PMC, remove the filler plates from both PMC slots.
5. Slide the edge connector(s) of the PMC module into the front panel opening(s) from behind and place the PMC module on top of the MVME240x. The four connectors on the underside of the PMC module should then connect smoothly with the corresponding connectors for a single-width PMC (J11/J12/J13/J14 or J21/J22/J23/J24, all eight for a double-width PMC) on the MVME240x.

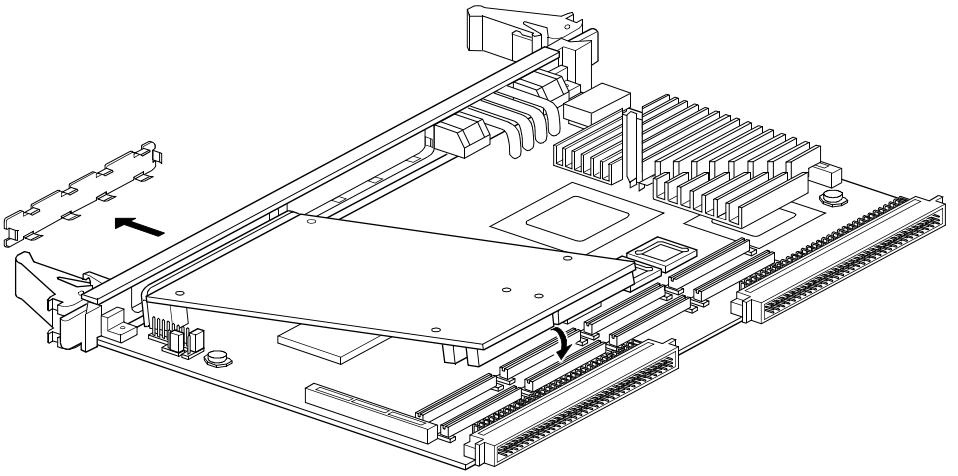


Figure 1-3. Typical Single-width PMC Module Placement on MVME240x

6. Insert the two short Phillips screws through the holes at the forward corners of the PMC module, into the standoffs on the MVME240x. Tighten the screws.
7. If installing two single-width PMCs, repeat the above procedure for the second PMC.

Primary PMCspan Installation

To install a PMCspan-002 PCI expansion module on your MVME240x, refer to Figure 1-4 on page 1-16 and perform the following steps:

Note This procedure assumes that you have read the user's manual that was furnished with the PMCspan, and that you have installed the selected PMCs on the PMCspan according to the instructions given in the PMCspan and PMC manuals.

1. Attach an ESD strap to your wrist. Attach the other end of the ESD strap to the chassis as a ground. The ESD strap must be secured to your wrist and to ground throughout the procedure.
2. Perform an operating system shutdown. Turn the AC or DC power off and remove the AC cord or DC power lines from the system. Remove chassis or system cover(s) as necessary for access to the VME modules.


Caution

Inserting or removing modules with power applied may result in damage to module components.


Warning

Dangerous voltages, capable of causing death, are present in this equipment. Use extreme caution when handling, testing, and adjusting.

3. Carefully remove the MVME2700 from its VMEbus card slot and lay it flat, with connectors P1 and P2 facing you.


Caution

Avoid touching areas of integrated circuitry; static discharge can damage these circuits.

4. Attach the four standoffs to the MVME240x module. For each standoff:
   - Insert the threaded end into the standoff hole at each corner of the VME processor module.
   - Thread the locking nuts onto the standoff tips.
   - Tighten the nuts with a box-end wrench or a pair of needle nose pliers.

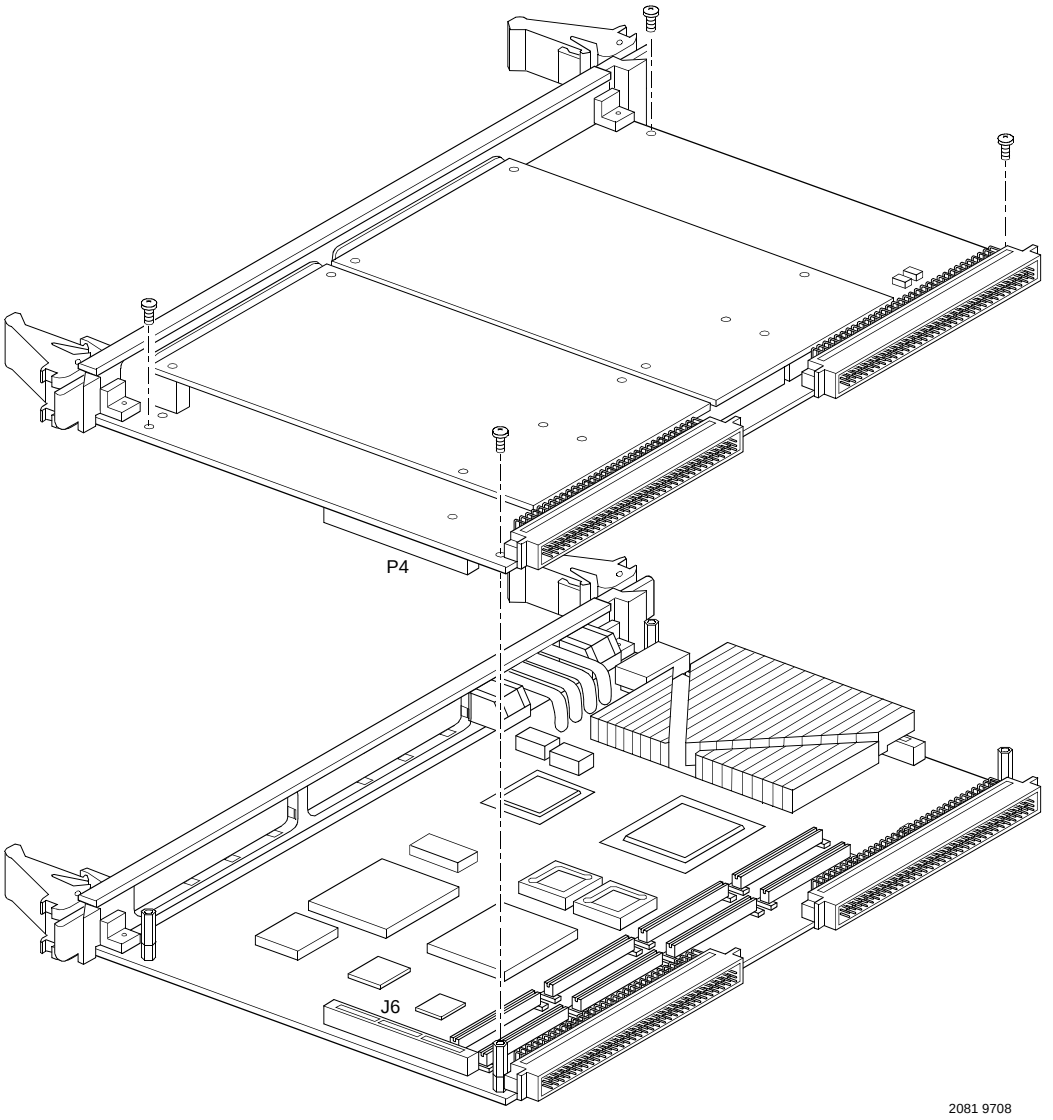


Figure 1-4. PMCspan-002 Installation on an MVME240x

5. Place the PMCspan on top of the MVME240x module. Align the mounting holes in each corner to the standoffs, and align PMCspan connector P4 with MVME240x connector J6.
6. Gently press the PMCspan and MVME240x together, making sure that P4 is fully seated in J6.
7. Insert the four short Phillips screws through the holes at the corners of the PMCspan and into the standoffs on the MVME240x module. Tighten the screws.

Note The screws have two different head diameters. Use the screws with the smaller heads on the standoffs next to VMEbus connectors P1 and P2.

Secondary PMCspan Installation

The PMCspan-010 PCI expansion module mounts on top of a PMCspan-002 PCI expansion module. To install a PMCspan-010 on your MVME240x, refer to Figure 1-5 and perform the following steps:

Note This procedure assumes that you have read the user's manual that was furnished with the PMCspan, and that you have installed the selected PMCs on the PMCspan according to the instructions given in the PMCspan and PMC manuals.

1. Attach an ESD strap to your wrist. Attach the other end of the ESD strap to the chassis as a ground. The ESD strap must be secured to your wrist and to ground throughout the procedure.
2. Perform an operating system shutdown. Turn the AC or DC power off and remove the AC cord or DC power lines from the system. Remove chassis or system cover(s) as necessary for access to the VME modules.

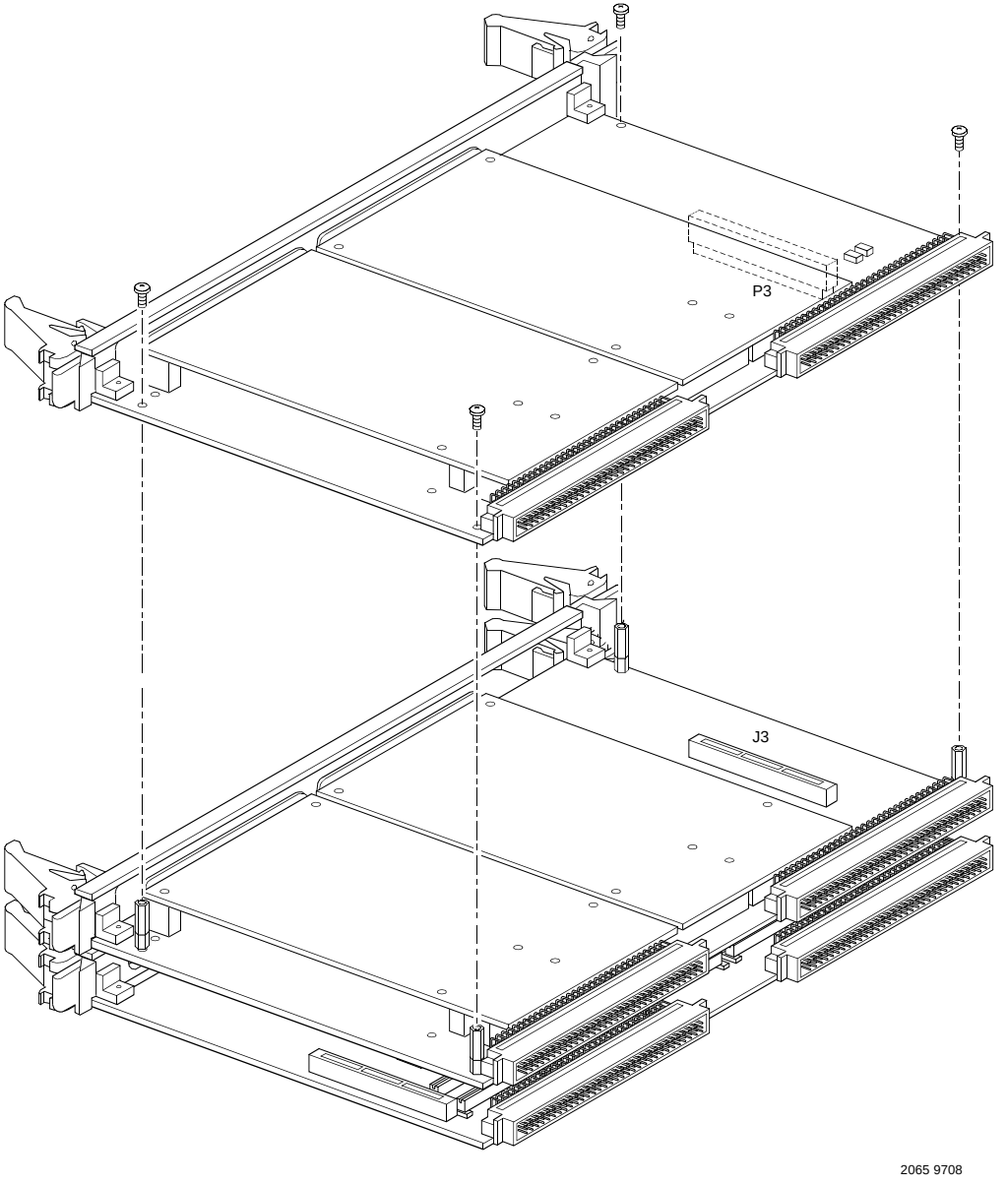


Figure 1-5. PMCspan-010 Installation on a PMCspan-002/MVME240x

Caution Inserting or removing modules with power applied may result in damage to module components.

Warning Dangerous voltages, capable of causing death, are present in this equipment. Use extreme caution when handling, testing, and adjusting.

3. If the Primary PMC Carrier Module/MVME240x assembly is already installed in the VME chassis, carefully remove the two-board assembly from the VMEbus card slots and lay it flat, with the P1 and P2 connectors facing you.

Caution Avoid touching areas of integrated circuitry; static discharge can damage these circuits.

4. Remove the four short Phillips screws from the standoffs in each corner of the primary PCI expansion module, PMCspan-002.
5. Attach the four standoffs to the PMCspan-002.
6. Place the PMCspan-010 on top of the PMCspan-002. Align the mounting holes in each corner to the standoffs, and align PMCspan-010 connector P3 with PMCspan-002 connector J3.
7. Gently press the two PMCspan modules together, making sure that P3 is fully seated in J3.
8. Insert the four short Phillips screws through the holes at the corners of PMCspan-010 and into the standoffs on the primary PMCspan-002. Tighten the screws.

Note The screws have two different head diameters. Use the screws with the smaller heads on the standoffs next to VMEbus connectors P1 and P2.

MVME240x Installation

Before installing the MVME240x into your VME chassis, ensure that the jumpers on the MVME240x J8, J9, and S3 switch are configured, as previously described. This procedure assumes that you have already installed the PMCspan(s) if desired, and any PMCs that you have selected.

Proceed as follows to install the MVME240x in the VME chassis:

1. Attach an ESD strap to your wrist. Attach the other end of the ESD strap to the chassis as a ground. The ESD strap must be secured to your wrist and to ground throughout the procedure.
2. Perform an operating system shutdown. Turn the AC or DC power off and remove the AC cord or DC power lines from the system. Remove chassis or system cover(s) as necessary for access to the VME modules.



Caution

Inserting or removing modules with power applied may result in damage to module components.



Warning

Dangerous voltages, capable of causing death, are present in this equipment. Use extreme caution when handling, testing, and adjusting.

3. Remove the filler panel from the card slot where you are going to install the MVME240x. If you have installed one or more PMCspan PCI expansion modules onto your MVME240x, you will need to remove filler panels from one additional card slot for each PMCspan, above the card slot for the MVME240x.
   - If you intend to use the MVME240x as system controller, it must occupy the left-most card slot (slot 1). The system controller must be in slot 1 to correctly initiate the bus-grant daisy-chain and to ensure proper operation of the IACK daisy-chain driver.
   - If you do not intend to use the MVME240x as system controller, it can occupy any unused card slot.

Caution Avoid touching areas of integrated circuitry; static discharge can damage these circuits.

4. Slide the MVME240x (and PMCspans if used) into the selected card slot(s). Be sure the module or modules is/are seated properly in the P1 and P2 connectors on the backplane. Do not damage or bend connector pins.
5. Secure the MVME240x (and PMCspans if used) in the chassis with the screws provided, making good contact with the transverse mounting rails to minimize RF emissions.

Note Some VME backplanes (for example, those used in Motorola "Modular Chassis" systems) have an auto-jumpering feature for automatic propagation of the IACK and BG signals. Step 6 does not apply to such backplane designs.

6. On the chassis backplane, remove the INTERRUPT ACKNOWLEDGE (IACK) and BUS GRANT (BG) jumpers from the header for the card slot occupied by the MVME240x.
7. If you intend to use PPCBug interactively, connect the terminal that is to be used as the PPCBug system console to the DEBUG port on the front panel of the MVME240x.

   In normal operation the host CPU controls MVME240x operation via the VMEbus Universe registers.
8. Replace the chassis or system cover(s), cable peripherals to the panel connectors as appropriate, reconnect the system to the AC or DC power source, and turn the equipment power on.
9. The MVME240x's green **CPU** LED indicates activity as a set of confidence tests is run, and the debugger prompt `PPC1-Bug>` appears.

System Considerations

The MVME240x draws power from the VMEbus backplane connectors P1 and P2. P2 is also used for the upper 16 bits of data in 32-bit transfers, and for the upper eight address lines in extended addressing mode. The MVME240x may not function properly without its main board connected to VMEbus backplane connectors P1 and P2.

Whether the MVME240x operates as a VMEbus master or as a VMEbus slave, it is configured for 32 bits of address and 32 bits of data (A32/D32). However, it handles A16 or A24 devices in the address ranges indicated in Chapter 4, *Programming Details*. D8 and/or D16 devices in the system must be handled by the PowerPC processor software. Refer to the memory maps in Chapter 4, *Programming Details*.

The MVME240x contains shared onboard DRAM whose base address is software-selectable. Both the onboard processor and off-board VMEbus devices see this local DRAM at base physical address $00000000, as programmed by the PPCBug firmware. This may be changed via software to any other base address. Refer to the MVME240x *Programmer's Reference Guide* for more information.

If the MVME240x tries to access off-board resources in a nonexistent location and is not system controller, and if the system does not have a global bus timeout, the MVME240x waits forever for the VMEbus cycle to complete. This will cause the system to lock up. There is only one situation in which the system might lack this global bus timeout: when the MVME240x is not the system controller and there is no global bus timeout elsewhere in the system.

Multiple MVME240x boards may be installed in a single VME chassis. Each must have a unique Universe address, selected by setting jumpers on its J17 header, as described in *MVME2400 Base Board Preparation*. In general, hardware multiprocessor features are supported.

Other MPUs on the VMEbus can interrupt, disable, communicate with, and determine the operational status of the processor(s). One register of the Universe set includes four bits that function as location monitors to allow

one MVME240x processor to broadcast a signal to any other MVME240x processors. All eight registers are accessible from any local processor as well as from the VMEbus.

Operating Instructions 2

Overview

This chapter provides operating instructions for the MVME240x. This includes information about powering up the system, and functionality of the switches, status indicators, and I/O ports on the front panels of the MVME240x and PMCspan modules.

Applying Power

After you have verified that all necessary hardware preparation has been done, that all connections have been made correctly, and that the installation is complete, you can power up the system. The MPU, hardware, and firmware initialization process is performed by the PPCBug firmware power-up or system reset. The firmware initializes the devices on the MVME240x module in preparation for booting the operating system.

The firmware is shipped from the factory with an appropriate set of defaults. In most cases there is no need to modify the firmware configuration before you boot the operating system. Refer to Chapter 6, *Environment Modification* for further information about modifying defaults.

The following flowchart in Figure 2-1 shows the basic initialization process that takes place during MVME240x system start-ups.

For further information on PPCbug, refer to the following items:

- Chapter 5, *PPCBug*
- Appendix C, *Troubleshooting*
- PPCBug documentation listed in Appendix D, *Related Documentation*.

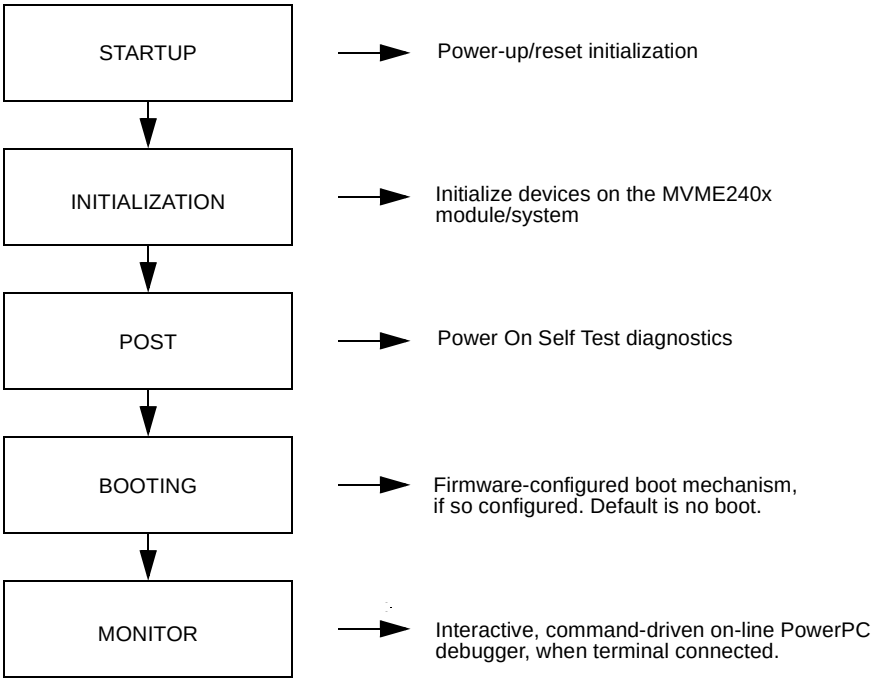


Figure 2-1. PPCBug Firmware System Startup

Switches

There are two switches (**ABT** and **RST**) located on the MVME240x front panel.

ABT (S1)

When activated by software, the Abort switch, **ABT**, can generate an interrupt signal from the base board to the processor at a user-programmable level. The interrupt is normally used to abort program execution and return control to the debugger firmware located in the MVME240x Flash memory. The interrupt signal reaches the processor module via ISA bus interrupt line IRQ8*. The signal is also available from the general purpose I/O port, which allows software to poll the Abort switch after an IRQ8* interrupt and verify that it has been pressed.

The interrupter connected to the **ABT** switch is an edge-sensitive circuit, filtered to remove switch bounce.

RST (S2)

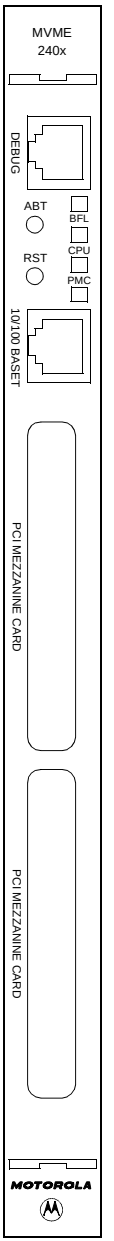
The Reset switch, **RST**, resets all onboard devices and causes HRESET* to be asserted in the MPC603 or MPC604. It also drives a SYSRESET* signal if the MVME240x VME processor module is the system controller.

The Universe ASIC includes both a global and a local reset driver. When the Universe operates as the VMEbus system controller, the reset driver provides a global system reset by asserting the VMEbus signal SYSRESET*. A SYSRESET* signal may be generated by the RESET switch, a power-up reset, a watchdog timeout, or by a control bit in the Miscellaneous Control Register (MISC_CTL) in the Universe ASIC. SYSRESET* remains asserted for at least 200 ms, as required by the VMEbus specification.

2

Front Panel Indicators (DS1 – DS4)

There are four LED (light-emitting diode) front panel indicators located on the MVME240x front panel: **BFL, CPU**, **PMC2**, and **PMC1**.



BFL (DS1)

The *yellow* **BFL** LED indicates board failure; lights when the BRDFAIL* signal line is active.

CPU (DS2)

The *green* **CPU** LED indicates CPU activity; lights when the DBB* (Data Bus Busy) signal line on the processor bus is active.

PMC2 (DS3)

The top *green* **PMC** LED indicates PCI activity; lights when the PCI bus grant to PMC2 signal line on the PCI bus is active. This indicates that a PMC installed on slot 2 is active.

PMC1 (DS4)

The bottom *green* **PMC** LED indicates PCI activity; lights when the PCI bus grant to PMC1 signal line on the PCI bus is active. This indicates that a PMC installed on slot 1 is active.

10/100BaseT Port

The RJ-45 port on the front panel of the MVME240x labeled **10/100 BASET** supplies the Ethernet LAN 10BaseT/100Base TX interface, implemented with a DEC 21140/21143 device.

Similarly, the Universe ASIC supplies an input signal and a control bit to initiate a local reset operation. By setting a control bit, software can maintain a board in a reset state, disabling a faulty board from participating in normal system operation. The local reset driver is enabled even when the Universe ASIC is not system controller. Local resets may be generated by the **RST** switch, a power-up reset, a watchdog timeout, a VMEbus SYSRESET*, or a control bit in the MISC_CTL register.

DEBUG Port

The RJ-45 port labeled **DEBUG** on the front panel of the MVME240x supplies the MVME240x serial communications interface, implemented via a UART PC16550 controller chip from National Semiconductor. It is asynchronous only. This serial port is configured for EIA-232-D DTE, as shown in Figure 2-2.

The **DEBUG** port may be used for connecting a terminal to the MVME240x to serve as the firmware console for the factory installed debugger, PPCBug. The port is configured as follows:

- 8 bits per character
- 1 stop bit per character
- Parity disabled (no parity)
- Baud rate = 9600 baud (default baud rate at power-up)

After power-up, the baud rate of the **DEBUG** port can be reconfigured by using the debugger's Port Format (**PF**) command. Refer to Chapter 5, *PPCBug* and Chapter 6, *Environment Modification* for information about the PPCBug.

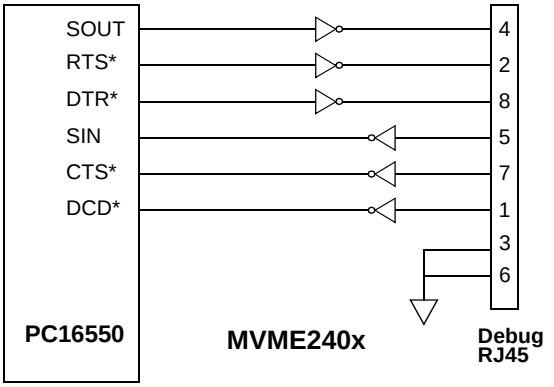
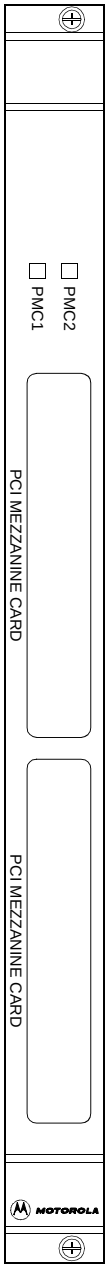


Figure 2-2. MVME240x DEBUG Port Configuration

PMC Slots



Two openings located on the front panel provide I/O expansion by allowing access to one or two 4-port single-wide or one 8-port double-wide PCI Mezzanine Card (PMC), connected to the PMC connectors on the MVME240x. Refer to Appendix B, *Connector Pin Assignments* for additional information on pin assignments for the PMC connectors.

Warning

Do not attempt to install any PMC boards without performing an operating system shutdown and following the procedures given in the user's manual for the particular PMC.

PCI MEZZANINE CARD (PMC Slot 1)

The right-most (lower) opening labeled **PCI MEZZANINE CARD** on the MVME240x front panel provides front panel I/O access to a PMC that is connected to the 64-pin connectors J11 through J14 on the MVME240x module. Connector J14 allows rear panel P2 I/O.

This slot is MVME240x Port 1.

PCI MEZZANINE CARD (PMC Slot 2)

The left-most (upper) opening labeled **PCI MEZZANINE CARD** on the MVME240x front panel provides front panel I/O access to a PMC that is connected to the 64-pin connectors J21 through J24 on the MVME240x module. Connector J24 allows rear panel P2 I/O.

This slot is MVME240x Port 2.

PMCspan

A PMCspan front panel is pictured on the previous page. The front panel is the same for all PMCspan models.

There are two PMC slots, labeled PCI MEZZANINE CARD, which support either two single-wide PMCs or one double-wide PMC.

The PMCspan board has two sets of three 32-bit connectors for PMC interface to a secondary PCI bus and a user-specific I/O. It also has a P1 connector and a 5-row P2 connector for power and VMEbus I/O.

The PMCspan has two green LEDs on its front panel, one for each PMC slot, labeled PMC2 and PMC1. Both LEDs are illuminated during reset. An individual LED is illuminated whenever a PMC has been granted bus mastership of the secondary PCI bus.

The right-most (lower) opening labeled **PCI MEZZANINE CARD** on the front panel is Port 1.

The left-most (upper) opening labeled **PCI MEZZANINE CARD** on the front panel is Port 2.

Functional Description 3

Introduction

This chapter provides a functional description of the MVME240x. This includes an overview of the product, followed by a detailed description of several blocks of circuitry. Figure 3-1 depicts a block diagram of the overall board architecture.

Detailed descriptions of other MVME240x blocks, including programmable registers in the ASICs and peripheral chips, can be found in the *MVME2400 Series VME Processor Module Programmer's Reference Guide*, listed in Appendix D, *Related Documentation*.

Features

The following table summarizes the features of the MVME240x.

Table 3-1. MVME240x Features

Feature	Description
Microprocessor	233 MHZ MPC750 PowerPC processor
	350 MHZ MPC750 PowerPC processor
	450 MHZ MPC750 PowerPC processor
Form factor	6U VMEbus
SDRAM	Double-Bit-Error detect, Single-Bit-Error correct across 72 bits 32MB, 64MB, or 128MB SDRAM
L2 Cache	Build-option for 1MB back side L2 Cache using late write or burst-mode SRAMS
Flash memory	Sockets for 1MB
	8MB Soldered on-board
Memory Controller	Hawk's SMC (System Memory Controller)
PCI Host Bridge	Hawk's PHB (PCI Host Bridge)
Interrupt Controller	Hawk's MPIC (Multi-Processor Interrupt Controller)

Table 3-1. MVME240x Features (Continued)

Feature	Description
PCI Interface	32/64-bit Data, 33 MHz operation
Real-time clock	8KB NVRAM with RTC and battery backup (SGS-Thomson M48T559)
Peripheral Support	One 16550-compatible async serial port routed to front panel RJ-45 10BaseT/100BaseTX Ethernet interface routed to front panel RJ-45
Switches	Reset (RST) and Abort (ABT)
Status LEDs	Four: Board fail (BFL), CPU, PMC (one for PMC slot 2, one for slot 1)
Timers	One 16-bit timer in W83C553 ISA bridge; four 32-bit timers in MPIC device
	Watchdog timer provided in SGS-Thomson M48T59
VME I/O	VMEbus P2 connector
PCI interface	Two IEEE P1386.1 PCI Mezzanine Card (PMC) slots for one double-width or two single-width PMCs
	Front panel and/or VMEbus P2 I/O on both PMC slots
	One 114-pin Mictor connector for optional PMCspan expansion module
VMEbus interface	VMEbus system controller functions
	64-bit PCI (Universe II)
	VMEbus-to-local-bus interface (A32/A24/A16, D64 (MBLT) D32//D16/D08 Master and Slave
	Local-bus-to-VMEbus interface (A16/A24/A32, D8/D16/D32)
	VMEbus interrupter
	VMEbus interrupt handler
	Global Control/Status Register (GCSR) for interprocessor communications
	DMA for fast local memory/VMEbus transfers (A16/A24/A32, D16/D32/D64)

General Description

The MVME240x is a VME processor module equipped with a PowerPC 604 RISC (MPC750) microprocessor.

As shown in the *Features* section, the MVME240x offers many standard features desirable in a computer system—including Ethernet and debug ports, Boot ROM, Flash memory, SDRAM, and interface for two PCI Mezzanine Cards (PMCs), contained in a one-slot VME package. Its flexible mezzanine architecture allows relatively easy upgrades of the I/O.

There are four standard buses on the MVME240x:

PowerPC Processor Bus	ISA Bus
PCI Local Bus	VMEbus

As shown in Figure 3-1, the PCI Bridge portion of the Hawk ASIC provides the interface from the Processor Bus to the PCI. A W83C553 PCI/ISA Bridge (PIB) Controller device performs the bridge function between PCI and ISA. The Universe ASIC device provides the interface between the PCI Local Bus and the VMEbus. Part of the Hawk ASIC is the ECC memory controller.

The Peripheral Component Interface (PCI) local bus is a key feature. In addition to the on-board local bus peripherals, the PCI bus supports an industry-standard mezzanine interface, IEEE P1386.1 PMC (PCI Mezzanine Card).

Block Diagram

Figure 3-1 is a block diagram of the MVME2400's overall architecture.

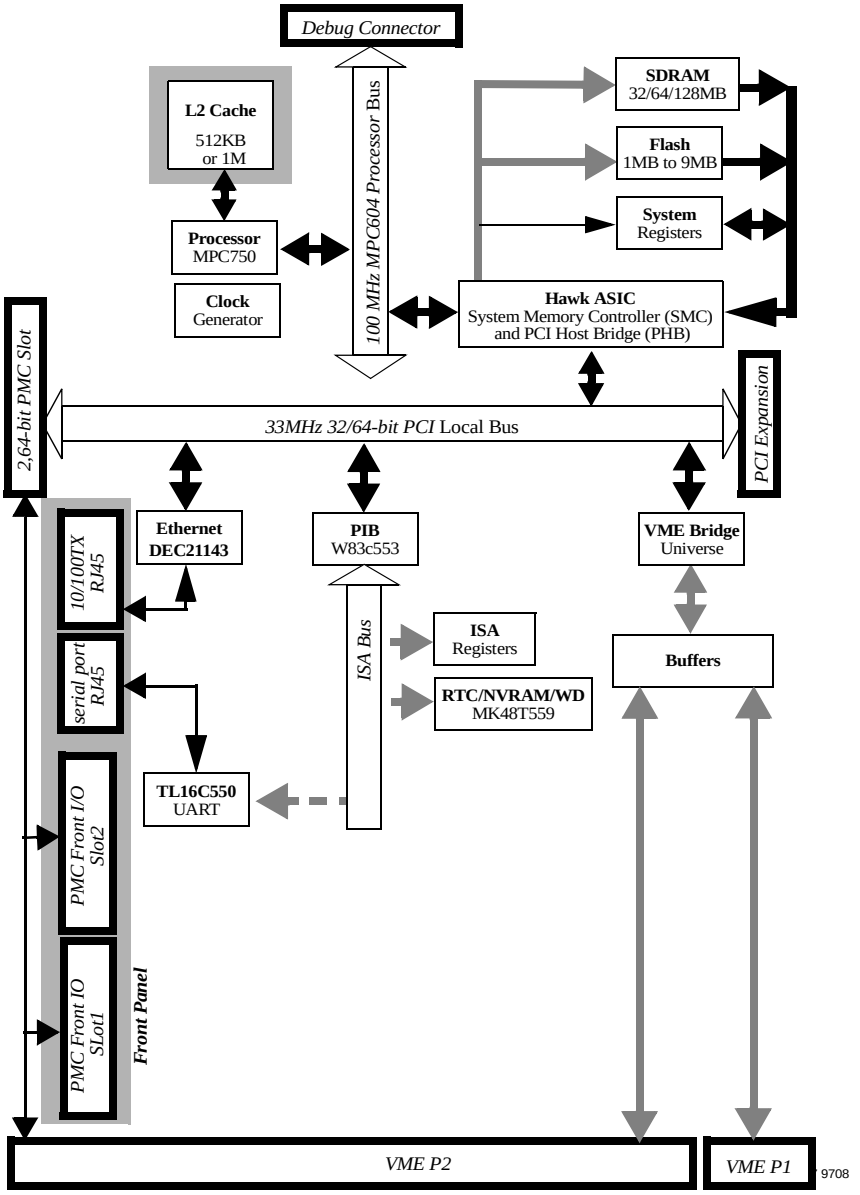


Figure 3-1. MVME240x Block Diagram

MPC750 Processor

The MVME240x can be ordered with a PowerPC 750 processor chip with 32MB to 512MB of ECC SDRAM, and up to 9MB of Flash memory.

The PowerPC 750 is a 64-bit processor with 32KB on-chip caches (32KB data cache and 32KB instruction cache).

The PHB bridge controller portion of the Hawk ASIC provides the bridge between the PowerPC microprocessor bus and the PCI local bus. Electrically, the Hawk is a 64-bit PCI connection. Four programmable map decoders in each direction provide flexible addressing between the PowerPC microprocessor bus and the PCI local bus.

The power requirements for the MVME240x are shown in Table 3-2.

Table 3-2. Power Requirements

Configuration	+5V Power	+12V and –12V Power
233 or 350 MHz 750	3.3 A typical 4.0 A maximum	PMC-dependent (Refer to Appendix A, *Specifications*)

L2 Cache

The MVME2400 SBC utilizes a back-door L2 cache structure via the MPC750 processor chip. The MCP750's L2 cache is implemented with an onchip 2-way set-associative tag memory and external direct-mapped synchronous SRAMs for data storage. The external SRAMs are accessed through a dedicated 72-bit wide (64 bits of data and 8 bits of parity) L2 cache port. The board is populated with 1MB of L2 cache SRAMs. The L2 cache can operate in copyback or writethru modes and supports system cache coherency through snooping. Parity generation and checking may be disabled by programming the MCP750 accordingly. Refer to the *MVME2400 Programmer's Reference Guide* for additional information.

Hawk System Memory Controller (SMC)/PCI Host Bridge (PHB) ASIC

The Hawk ASIC provides the bridge function between the MPC60x bus and the PCI Local Bus. It provides 32-bit addressing and 64-bit data. The 64-bit addressing (dual address cycle) is not supported. The Hawk supports various PowerPC processor external bus frequencies up to 100 MHz.

There are four programmable map decoders for each direction to provide flexible address mappings between the MPC and the PCI Local Bus. Refer to the *MVME2400 Programmer's Reference Guide* for additional information.

The Hawk ASIC also provides an MPIC Interrupt Controller to handle various interrupt sources. The interrupt sources are: Four MPIC Timer Interrupts, the interrupts from all PCI devices, the two software interrupts, and the ISA interrupts. The ISA interrupts actually are handled as a single 8259 interrupt at INT0.

PCI Bus Latency

The following tables list the latency of PCI originated transactions and the bandwidth of originated transactions for five different clock ratios: 5:2, 3:2, 3:1, 2:1, and 1:1. The MVME2400 uses a 3:1 clock ratio:

Table 3-3. PCI Originated Latency Matrix

Transaction	32-bit PCI					64-bit PCI					Clock Ratio
	Beat 1	Beat 2	Beat 3	Beat 4	Total	Beat 1	Beat 2	Beat 3	Beat 4	Total	
Burst Read	9	1	1	1	12	9	1	1	1	12	5:2
Burst Write	3	1	1	1	6	3	1	1	1	6	
Single Read	9	-	-	-	9	9	-	-	-	9	
Single Write	3	-	-	-	3	3	-	-	-	3	
Burst Read	12	1	1	1	15	12	1	1	1	15	3:2
Burst Write	3	1	1	1	6	3	1	1	1	6	
Single Read	12	-	-	-	12	12	-	-	-	12	
Single Write	3	-	-	-	3	3	-	-	-	3	
Burst Read	9	1	1	1	12	9	1	1	1	12	3:1
Burst Write	3	1	1	1	6	3	1	1	1	6	
Single Read	9	-	-	-	9	-	-	-	-	-	
Single Write	3	-	-	-	3	-	-	-	-	-	
Burst Read	11	1	1	1	14	11	1	1	1	14	2:1
Burst Write	3	1	1	1	6	3	1	1	1	6	
Single Read	11	-	-	-	11	-	-	-	-	-	
Single Write	3	-	-	-	3	-	-	-	-	-	
Burst Read	16	1	1	1	19	16	1	1	1	19	1:1
Burst Write	3	1	1	1	6	3	1	1	1	6	
Single Read	16	-	-	-	16	-	-	-	-	-	
Single Write	3	-	-	-	3	-	-	-	-	-	

Table 3-4. PCI Originated Bandwidth Matrix

Transaction	First 2 Cache Lines		First 4 Cache Lines		First 6 Cache Lines		Continuous		Clock Ratio
	Clks	MBytes sec	Clks	MBytes sec	Clks	MBytes sec	Clks/ Line	MBytes sec	
64-bit Writes	10	213	18	237	26	246	4	266	5:2
64-bit Reads	16	133	24	178	32	200	4	266	
32-bit Writes	18	118	34	125	50	128	8	133	
32-bit Reads	24	89	40	107	56	114	8	133	
64-bit Writes	10	427	18	474	26	492	4	533	3:2
64-bit Reads	19	225	27	316	37	346	4	533	
32-bit Writes	18	237	34	251	50	256	8	267	
32-bit Reads	28	152	44	194	60	213	8	267	
64-bit Writes	10	213	18	237	26	246	4	266	3:1
64-bit Reads	16	133	24	178	32	200	4	266	
32-bit Writes	18	118	34	125	50	128	8	133	
32-bit Reads	24	89	40	107	56	114	8	133	
64-bit Writes	10	213	18	237	26	246	4	266	2:1
64-bit Reads	18	118	26	164	34	188	4	266	
32-bit Writes	18	118	34	125	50	128	8	133	
32-bit Reads	26	82	42	102	58	110	8	133	
64-bit Writes	10	427	18	474	30	427	5	427	1:1
64-bit Reads	23	186	34	251	46	278	5.5	388	
32-bit Writes	18	237	34	251	50	256	8	267	
32-bit Reads	31	138	47	182	63	203	8	267	

PPC Bus Latency

The following tables list the latency of PPC originated transactions and the bandwidth of originated transactions for five different clock ratios: 5:2, 3:2, 3:1, 2:1, and 1:1. The MVME2400 uses a 3:1 clock ratio:

Table 3-5. PPC60x Originated Latency Matrix

Transaction	32-bit PCI					64-bit PCI					Clock Ratio
	Beat 1	Beat 2	Beat 3	Beat 4	Total	Beat 1	Beat 2	Beat 3	Beat 4	Total	
Burst Read	40	1	1	1	43	29	1	1	1	32	5:2
Burst Write	5	1	1	1	8	5	1	1	1	8	
Single Read	22	-	-	-	22	-	-	-	-	-	
Single Write	5	-	-	-	5	-	-	-	-	-	
Burst Read	26	1	1	1	29	20	1	1	1	23	3:2
Burst Write	5	1	1	1	8	5	1	1	1	8	
Single Read	16	-	-	-	16	-	-	-	-	-	
Single Write	5	-	-	-	5	-	-	-	-	-	
Burst Read	45	1	1	1	48	33	1	1	1	36	3:1
Burst Write	5	1	1	1	8	5	1	1	1	8	
Single Read	24	-	-	-	24	-	-	-	-	-	
Single Write	5	-	-	-	5	-	-	-	-	-	
Burst Read	33	1	1	1	36	25	1	1	1	28	2:1
Burst Write	5	1	1	1	8	5	1	1	1	8	
Single Read	19	-	-	-	19	-	-	-	-	-	
Single Write	5	-	-	-	5	-	-	-	-	-	
Burst Read	20	1	1	1	23	16	1	1	1	19	1:1
Burst Write	5	1	1	1	8	5	1	1	1	8	
Single Read	13	-	-	-	13	-	-	-	-	-	
Single Write	5	-	-	-	5	-	-	-	-	-	

Table 3-6. PPC60x Originated Bandwidth Matrix

Transaction	First 2 Cache Lines		First 4 Cache Lines		First 6 Cache Lines		Continuous		Clock Ratio
	Clks	MBytes Sec	Clks	MBytes Sec	Clks	MBytes Sec	Clks/ Line	MBytes Sec	
64-bit Writes	14	381	58	184	108	148	25	107	5:2
64-bit Reads	-	-	-	-	-	-	32.5	82	
32-bit Writes	14	381	78	137	148	108	35	76	
32-bit Reads	-	-	-	-	-	-	42.5	63	
64-bit Writes	14	457	38	337	68	282	15	213	3:2
64-bit Reads	-	-	-	-	-	-	22.5	142	
32-bit Writes	14	457	50	256	92	209	21	152	
32-bit Reads	-	-	-	-	-	-	28.5	112	
64-bit Writes	14	457	67	191	127	151	30	107	3:1
64-bit Reads	-	-	-	-	-	-	36	89	
32-bit Writes	14	457	98	131	182	105	42	76	
32-bit Reads	-	-	-	-	-	-	48	67	
64-bit Writes	14	305	48	178	88	145	20	107	2:1
64-bit Reads	-	-	-	-	-	-	28	76	
32-bit Writes	14	305	64	133	120	107	28	76	
32-bit Reads	-	-	-	-	-	-	36	59	
64-bit Writes	14	305	29	294	49	261	10	213	1:1
64-bit Reads	-	-	-	-	-	-	18	118	
32-bit Writes	14	305	37	231	65	197	14	152	
32-bit Reads	-	-	-	-	-	-	22	97	

Assumptions

Certain assumptions have been made with regard to MVME2400 performance. Some things, which are assumed in making the previous tables, include the following:

Clock Ratios and Operating Frequencies

Performance is based on the appropriate clock ratio and corresponding operating frequency:

Table 3-7. Clock Ratios and Operating Frequencies

Ratio	PPC60x Clock (MHz)	PCI Clock (MHz)	SDRAM Speed (ns)
5:2	83	33	8
3:2	100	66	8
3:1	100	33	8
2:1	66	33	10
1:1	66	66	10

PPC60x Originated

- Count represents number of PPC60x bus clock cycles.
- Assumes write posting FIFO is initially empty.
- Does not include time taken to obtain grant for PPC60x bus. The count starts on the same clock period that TS_ is asserted.
- PPC60x bus is idle at the time of the start of the transaction. (that is, no pipelining effects).
- Cache aligned transfer, not critical word first.
- PCI medium responder with no zero states.
- One clock request/one clock grant PCI arbitration.
- Write posting enabled.

- Default FIFO threshold settings
- Single beat writes are aligned 32-bit transfer, always executed as 32-bit PCI.
- Clock counts represent best case alignment between PCI and PPC60x clock domains. An exception to this is continuous bandwidth which reflects the average affects of clock alignment.

PCI Originated

- Count represents number of PCI Bus clock cycles.
- Assumes write posting FIFO is initially empty
- L2 caching is not enabled, all transactions exclusively controlled by the SMC.
- Does not include time taken to obtain grant for PCI Bus. The count starts on the same clock period that FRAME_ is asserted.
- One clock request/one clock grant PPC60x bus arbitration.
- PPC60x bus traffic limited to PHB transactions only.
- Write posting and read ahead enabled.
- Default FIFO threshold settings.
- One cache line = 32 bytes.

SDRAM Memory

The MVME2400 SDRAM memory size can be 32MB, 64MB, or 128MB.

The SDRAM blocks are controlled by the Hawk ASIC which provides single-bit error correction and double-bit error detection. ECC is calculated over 72-bits.

The memory block size is dependant upon the SDRAM devices installed. Installing five 64Mbit (16-bit data) devices provides 32MB of memory. With 64Mbit (8bit data) devices, there are two blocks consisting of nine devices each that total 64MB per block. In this case, either block can be

populated for 64MB or 128MB of onboard memory. With 128Mbit (8-bit data) devices, the blocks can be populated for 128MB and 256MB. If 64Mbit (4-bit data) devices are installed, there is one block consisting of 18 devices that total 128MB. With 128Mbit (4-bit data) devices, the block contains 256MB. When populated, these blocks appear as Block A and Block B to the Hawk.

Refer to the *MVME2400 Series VME Processor Module Programmer's Reference Guide* for additional information and programming details.

SDRAM Latency

The following table shows the performance summary for SDRAM when operating at 100 MHz using PC100 SDRAM with a CAS_latency of 2. The figure on the next page defines the times that are specified in the table.

Table 3-8. 60x Bus to SDRAM Access Timing (100 MHz/PC100 SDRAMs)

ACCESS TYPE	Access Time (tB1-tB2-tB3-tB4)	Comments
4-Beat Read after idle, SDRAM Bank Inactive	10-1-1-1	
4-Beat Read after idle, SDRAM Bank Active - Page Miss	12-1-1-1	
4-Beat Read after idle, SDRAM Bank Active - Page Hit	7-1-1-1	
4-Beat Read after 4-Beat Read, SDRAM Bank Active - Page Miss	5-1-1-1	
4-Beat Read after 4-Beat Read, SDRAM Bank Active - Page Hit	2.5-1-1-1	2.5-1-1-1 is an average of 2-1-1-1 half of the time and 3-1-1-1 the other half.
4-Beat Write after idle, SDRAM Bank Active or Inactive	4-1-1-1	
4-Beat Write after 4-Beat Write, SDRAM Bank Active - Page Miss	6-1-1-1	

Table 3-8. 60x Bus to SDRAM Access Timing (100 MHz/PC100 SDRAMs)

ACCESS TYPE	Access Time (tB1-tB2-tB3-tB4)	Comments
4-Beat Write after 4-Beat Write, SDRAM Bank Active - Page Hit	3-1-1-1	3-1-1-1 for the second burst write after idle. 2-1-1-1 for subsequent burst writes.
1-Beat Read after idle, SDRAM Bank Inactive	10	
1-Beat Read after idle, SDRAM Bank Active - Page Miss	12	
1-Beat Read after idle, SDRAM Bank Active - Page Hit	7	
1-Beat Read after 1-Beat Read, SDRAM Bank Active - Page Miss	8	
1-Beat Read after 1-Beat Read, SDRAM Bank Active - Page Hit	5	
1-Beat Write after idle, SDRAM Bank Active or Inactive	5	
1-Beat Write after 1-Beat Write, SDRAM Bank Active - Page Miss	13	
1-Beat Write after 1-Beat Write, SDRAM Bank Active - Page Hit	8	

Notes

1. SDRAM speed attributes are programmed for the following: CAS_latency = 2, tRCD = 2 CLK Periods, tRP = 2 CLK Periods, tRAS = 5 CLK Periods, tRC = 7 CLK Periods, tDP = 2 CLK Periods, and the **swr_dpl** bit is set in the SDRAM Speed Attributes Register.

2. The Hawk is configured for “no external registers” on the SDRAM control signals.

3. tB1, tB2, tB3, and tB4 are specified in the following figure.

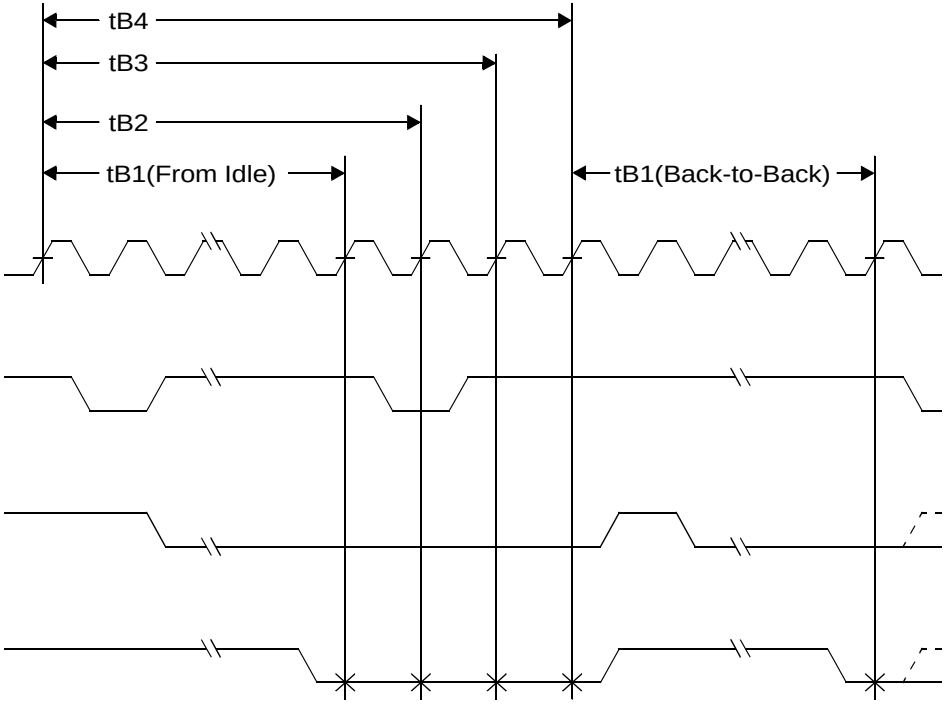


Figure 3-2. Timing Definitions for PPC Bus to SDRAM Access

Notes

1. When the initial bus state is idle, tB1 reflects the number of CLK periods from the rising edge of the CLK that drives TS_low, to the rising edge of the CLK that samples the first TA_low.

2. When the bus is busy and TS_ is being asserted as soon as possible after Hawk asserts AACK_ the back-to-back condition occurs. When back-to-back cycles occur, tB1 reflects the number of CLK periods from the rising edge of the CLK that samples the last TA_

low of a data tenure to the rising edge of the CLK that samples the first TA_ low of the next data tenure.

3. The tB2 function reflects the number of CLK periods from the rising edge of the CLK that samples the first TA_ low in a burst data tenure to the rising edge of the CLK that samples the second TA_ low in that data tenure.
4. The tB3 function reflects the number of CLK periods from the rising edge of the CLK that samples the second TA_ low in a burst data tenure to the rising edge of the CLK that samples the third TA_ low in that data tenure.
5. The tB4 function reflects the number of CLK periods from the rising edge of the CLK that samples the third TA_ low in a burst data tenure to the rising edge of the CLK that samples the last TA_ low in that data tenure.

Flash Memory

The MVME240x base board contains two banks of Flash memory. Bank B consists of two 32-pin devices which can be populated with 1MB of Flash memory. Only 8-bit writes are supported for this bank. Bank A has four 16-bit Smart Voltage Flash SMT devices. With the 16Mbit Flash devices, the Flash size is 8MB. A jumper header associated with the first set of eight Flash devices provides a total of 128KB of hardware-protected boot block. Only 32-bit writes are supported for this bank of Flash. There will be a jumper to tell the Hawk chip where to fetch the reset vector. When the jumper is installed, the Hawk chip maps 0xFFF00100 to these sockets (Bank B).

The onboard monitor/debugger, PPCBug, resides in the Flash chips. PPCBug provides functionality for:

- ❏ Booting the system
- ❏ Initializing after a reset
- ❏ Displaying and modifying configuration variables
- ❏ Running self-tests and diagnostics

- Updating firmware ROM

Under normal operation, the Flash devices are in “read-only” mode, their contents are pre-defined, and they are protected against inadvertent writes due to loss of power conditions. However, for programming purposes, programming voltage is always supplied to the devices and the Flash contents may be modified by executing the proper program command sequence. Refer to the **PFLASH** command in the *PPCbug Debugging Package User’s Manual* for further device-specific information on modifying Flash contents.

ROM/Flash Performance

The SMC provides the interface for two blocks of ROM/Flash. Access times to ROM/Flash are programmable for each block. Access times are also affected by block width. The following tables in this subsection show access times for ROM/Flash when configured for different device access times.

Table 3-9. PPC Bus to ROM/Flash Access Timing (120ns @ 100 MHz)

ACCESS TYPE	CLOCK PERIODS REQUIRED FOR:								Total Clocks	
	1st Beat		2nd Beat		3rd Beat		4th Beat			
	16 Bits	64 Bits	16 Bits	64 Bits	16 Bits	64 Bits	16 Bits	64 Bits	16 Bits	64 Bits
4-Beat Read	70	22	64	16	64	16	64	16	262	70
4-Beat Write	N/A								N/A	
1-Beat Read (1 byte)	22	22	-	-	-	-	-	-	22	22
1-Beat Read (2 to 8 bytes)	70	22	-	-	-	-	-	-	70	22
1-Beat Write	21	21	-	-	-	-	-	-	21	21

Note The information in Table 3-9 is appropriate when configured with devices with an access time equal to 12 CLK periods.

Table 3-10. PPC Bus to ROM/Flash Access Timing (80ns @ 100 MHz)

ACCESS TYPE	CLOCK PERIODS REQUIRED FOR:								Total Clocks	
	1st Beat		2nd Beat		3rd Beat		4th Beat			
	16 Bits	64 Bits	16 Bits	64 Bits	16 Bits	64 Bits	16 Bits	64 Bits	16 Bits	64 Bits
4-Beat Read	54	18	48	12	48	12	48	12	198	54
4-Beat Write	N/A								N/A	
1-Beat Read (1 byte)	18	18	-	-	-	-	-	-	18	18
1-Beat Read (2 to 8 bytes)	54	18	-	-	-	-	-	-	54	18
1-Beat Write	21	21	-	-	-	-	-	-	21	21

Note The information in Table 3-10 is appropriate when configured with devices with an access time equal to 8 CLK periods.

Table 3-11. PPC Bus to ROM/Flash Access Timing (50ns @ 100MHz)

ACCESS TYPE	CLOCK PERIODS REQUIRED FOR:								Total Clocks	
	1st Beat		2nd Beat		3rd Beat		4th Beat			
	16 Bits	64 Bits	16 Bits	64 Bits	16 Bits	64 Bits	16 Bits	64 Bits	16 Bits	64 Bits
4-Beat Read	42	15	36	9	36	9	36	9	150	42
4-Beat Write	N/A								N/A	
1-Beat Read (1 byte)	15	15	-	-	-	-	-	-	15	15
1-Beat Read (2 to 8 bytes)	42	15	-	-	-	-	-	-	42	15
1-Beat Write	21	21	-	-	-	-	-	-	21	21

Note The information in Table 3-11 is appropriate when configured with devices with an access time equal to 5 CLK periods.

Table 3-12. PPC Bus to ROM/Flash Access Timing (30ns @ 100 MHz)

ACCESS TYPE	CLOCK PERIODS REQUIRED FOR:								Total Clocks	
	1st Beat		2nd Beat		3rd Beat		4th Beat			
	16 Bits	64 Bits	16 Bits	64 Bits	16 Bits	64 Bits	16 Bits	64 Bits	16 Bits	64 Bits
4-Beat Read	34	13	28	7	28	7	28	7	118	34
4-Beat Write	N/A								N/A	
1-Beat Read (1 byte)	13	13	-	-	-	-	-	-	13	13
1-Beat Read (2 to 8 bytes)	34	13	-	-	-	-	-	-	34	13
1-Beat Write	21	21	-	-	-	-	-	-	21	21

Note The information in Table 3-12 is appropriate when configured with devices with an access time equal to 3 CLK periods.

Ethernet Interface

The MVME240x module uses Intel's DECchip 21143 PCI Fast Ethernet LAN controller to implement an Ethernet interface that supports 10BaseT/100BaseTX connections, via an RJ-45 connector on the front panel. The balanced differential transceiver lines are coupled via on-board transformers.

Every MVME240x is assigned an Ethernet station address. The address is $08003E2*xxxxx*, where *xxxxx* is the unique 5-nibble number assigned to the board (that is, every board has a different value for *xxxxx*).

Each MVME240x displays its Ethernet station address on a label attached to the base board in the PMC connector keepout area just behind the front panel. In addition, the six bytes including the Ethernet station address are stored in the NVRAM (BBRAM) configuration area specified by boot ROM. That is, the value 08003E2*xxxxx* is stored in NVRAM. The MVME240x debugger, PPCBug, has the capability to retrieve the Ethernet station address via the **CNFG** command.

Note The unique Ethernet address is set at the factory and should not be changed. Any attempt to change this address may create node or bus contention and thereby render the board inoperable.

If the data in NVRAM is lost, use the number on the label in the PMC connector keepout area to restore it.

For the pin assignments of the 10BaseT/100BaseTX connector, refer to Appendix B, *Connector Pin Assignments*.

At the physical layer, the Ethernet interface bandwidth is 10Mbit/second for 10BaseT. For the 100BaseTX, it is 100Mbit/second. Refer to the BBRAM/TOD Clock memory map description in the *MVME2400 Series VME Processor Module Programmer's Reference Guide* for detailed programming information.

PCI Mezzanine Card (PMC) Interface

A key feature of the MVME240x family is the PCI bus. In addition to the on-board local bus devices (Ethernet, etc.), the PCI bus supports an industry-standard mezzanine interface, IEEE P1386.1 PCI Mezzanine Card (PMC).

PMC modules offer a variety of possibilities for I/O expansion such as FDDI (Fiber Distributed Data Interface), ATM (Asynchronous Transfer Mode), graphics, Ethernet, or SCSI ports. For a complete listing of available PMCs, go to the GroupIPC's Web site at http://www.groupipc.com/. The MVME240x supports PMC front panel and rear P2 I/O. There is also provision for stacking one or two PMC carrier boards, or PMCspan PCI expansion modules, on the MVME240x for additional expansion.

The MVME240x supports two PMC slots. Two sets of four 64-pin connectors on the base board (J11 - J14, and J21 - J24) interface with 32-bit/64-bit IEEE P1386.1 PMC-compatible mezzanines to add any desirable function.

Refer to Appendix B, *Connector Pin Assignments* for the pin assignments of the PMC connectors. For detailed programming information, refer to the PCI bus descriptions in the *MVME2400 Series VME Processor Module Programmer's Reference Guide* and to the user documentation for the PMC modules you intend to use.

PMC Slot 1 (Single-Width PMC)

PMC slot 1 has the following characteristics:

Mezzanine Type	PCI Mezzanine Card (PMC)
Mezzanine Size	S1B: Single width, standard depth (75 mm x 150 mm) with front panel
PMC Connectors	J11 to J14 (32/64-bit PCI with front and rear I/O)
Signaling Voltage	V_{io} = 5.0Vdc

For P2 I/O configurations, all I/O pins of PMC slot 1 are routed to the 5-row power adapter card. Pins 1 through 64 of J14 are routed to row C and row A of P2.

PMC Slot 2 (Single-Width PMC)

PMC slot 2 has the following characteristics:

Mezzanine Type	PCI Mezzanine Card (PMC)
Mezzanine Size	S1B: Single width, standard depth (75 mm x 150 mm) with front panel
PMC Connectors	J21 to J24 (32/64-bit PCI with front and rear I/O)
Signaling Voltage	V_{io} = 5.0Vdc

For P2 I/O configurations, 46 I/O pins of PMC slot 2 are routed to the 5-row power adapter card. Pins 1 through 46 of J24 are routed to row D and row Z of P2.

PMC Slots 1 and 2 (Double-Width PMC)

PMC slots 1 and 2 with a double-width PMC have the following characteristics:

Mezzanine Type	PCI Mezzanine Card (PMC)
Mezzanine Size	Double width, standard depth (150 mm x 150 mm) with front panel
PMC Connectors	J11 to J14 and J21 to J24 (32/64-bit PCI) with front and rear I/O
Signaling Voltage	V_{io} = 5.0Vdc

PCI Expansion

The PMCspan expansion module connector, J6, is a 114-pin Mictor connector. It is located near P2 on the primary side of the MVME240x. Its interrupt lines are routed to the MPIC.

VMEbus Interface

The VMEbus interface is implemented with the CA91C142 Universe ASIC. The Universe chip interfaces the 32/64-bit PCI local bus to the VMEbus.

The Universe ASIC provides:

- The PCI-bus-to-VMEbus interface
- The VMEbus-to-PCI-bus interface
- The DMA controller functions of the local VMEbus

The Universe chip includes Universe Control and Status Registers (UCSRs) for interprocessor communications. It can provide the VMEbus system controller functions as well. For detailed programming information, refer to the *Universe User's Manual* and to the discussions in the *MVME2400 Series VME Processor Module Programmer's Reference Guide*.

Maximum performance is achieved with D64 Multiplexed Block Transfers (MBLT). The on-chip DMA channel should be used to move large blocks of data to/from the VMEbus. The Universe should be able to reach 50MB/second in 64-bit MBLT mode.

The MVME2400 interfaces to the VMEbus via the P1 and P2 connectors, which use the new 5-row 160-pin connectors as specified in the VME64 extension standard. It also draws +5V, +12V, and –12V power from the VMEbus backplane through these two connectors. 3.3V and 2.5V supplies are regulated onboard from the +5 power.

Asynchronous Debug Port

A Texas Instrument's Universal Asynchronous Receiver/Transmitter (UART) provides the asynchronous debug port. TTL-level signals for the port are routed through appropriate EIA-232-D drivers and receivers to an RJ-45 connector on the front panel. The external signals are ESD protected.

For detailed programming information, refer to the *MVME2400 Series VME Processor Module Programmer's Reference Guide* and to Texas Instrument's data sheet #SLLS057D, dated August 1989, revised March 1996 for Asynchronous Communications Element (ACE) TL16C550A.

PCI-ISA Bridge (PIB) Controller

The MVME240x uses a Winbond W83C553 PCI/ISA Bridge (PIB) Controller to supply the interface between the PCI local bus and the ISA system I/O bus (diagrammed in Figure 3-1 on page 3-4).

The PIB controller provides the following functions:

- PCI bus arbitration for:
  - ISA (Industry Standard Architecture) bus DMA (not functional on MVME240x)
  - The PHB (PCI Host Bridge) MPU/local bus interface function, implemented by the Hawk ASIC
  - All on-board PCI devices
  - The PMC slot

- ISA bus arbitration for DMA devices
- ISA interrupt mapping for four PCI interrupts
- Interrupt controller functionality to support 14 ISA interrupts
- Edge/level control for ISA interrupts
- Seven independently programmable DMA channels
- One 16-bit timer
- Three interval counters/timers

Accesses to the configuration space for the PIB controller are performed by way of the CONADD and CONDAT (Configuration Address and Data) registers in the PHB. The registers are located at offsets $CF8 and $CFC, respectively, from the PCI I/O base address.

Real-Time Clock/NVRAM/Timer Function

The MVME240x employs an SGS-Thomson surface-mount M48T559 RAM and clock chip to provide 8KB of non-volatile static RAM, a real-time clock, and a watchdog timer function. This chip supplies a clock, oscillator, crystal, power failure detection, memory write protection, 8KB of NVRAM, and a battery in a package consisting of two parts:

- A 28-pin 330mil SO device containing the real-time clock, the oscillator, power failure detection circuitry, timer logic, 8KB of static RAM, and gold-plated sockets for a battery
- A SNAPHAT® battery housing a crystal along with the battery

The SNAPHAT battery package is mounted on top of the M48T559 device. The battery housing is keyed to prevent reverse insertion.

The clock furnishes seconds, minutes, hours, day, date, month, and year in BCD 24-hour format. Corrections for 28-, 29- (leap year), and 30-day months are made automatically. The clock generates no interrupts. Although the M48T559 is an 8-bit device, 8-, 16-, and 32-bit accesses from the ISA bus to the M48T559 are supported. Refer to the *MVME2400 Series*

VME Processor Module Programmer's Reference Guide and to the M48T559 data sheet for detailed programming and battery life information.

PCI Host Bridge (PHB)

The PHB portion of the Hawk ASIC provides the bridge function between the MPC60x bus and the PCI Local Bus. It provides 32 bit addressing and 64 bit data. The 64 bit addressing (dual address cycle) is not supported. The Hawk supports various PowerPC processor external bus frequencies up to 100MHz and PCI frequencies up to 33 MHz.

There are four programmable map decoders for each direction to provide flexible address mappings between the MPC and the PCI Local Bus. Refer to the *MVME2400 Series VME Processor Module Programmer's Reference Guide* for additional information and programming details.

Interrupt Controller (MPIC)

The MPIC Interrupt Controller portion of the Hawk ASIC is designed to handle various interrupt sources. The interrupt sources are:

- Four MPIC timer interrupts
- Processor 0 self interrupt
- Memory Error interrupt from the SMC
- Interrupts from all PCI devices
- Two software interrupts
- ISA interrupts (actually handles as a single 8259 interrupt at INT0)

Programmable Timers

Among the resources available to the local processor are a number of programmable timers. Timers are incorporated into the PCI/ISA Bridge (PIB) controller and the Hawk device (diagrammed in Figure 3-1 on page 3-4). They can be programmed to generate periodic interrupts to the processor.

Interval Timers

The PIB controller has three built-in counters that are equivalent to those found in an 82C54 programmable interval timer. The counters are grouped into one timer unit, Timer 1, in the PIB controller. Each counter output has a specific function:

- Counter 0 is associated with interrupt request line IRQ0. It can be used for system timing functions, such as a timer interrupt for a time-of-day function.
- Counter 1 generates a refresh request signal for ISA memory. This timer is not used in the MVME240x.
- Counter 2 provides the tone for the speaker output function on the PIB controller (the SPEAKER_OUT signal which can be cabled to an external speaker via the remote reset connector). This function is not used on the MVME240x.

The interval timers use the OSC clock input as their clock source. The MVME240x drives the OSC pin with a 14.31818 MHz clock source.

16/32-Bit Timers

There is one 16-bit timer and four 32-bit timers on the MVME240x. The 16-bit timer is provided by the PIB. The Hawk device provides the four 32-bit timers that may be used for system timing or to generate periodic interrupts. For information on programming these timers, refer to the data sheet for the W83C553 PIB controller and to the *MVME2400 Series VME Processor Module Programmer's Reference Guide*.

Programming Details

4

Introduction

This chapter provides information useful in programming the MVME240x. This includes a description of memory maps, control and status registers, PCI arbitration, interrupt handling, sources of reset, and big/little-endian issues.

For additional programming information about the MVME240x, refer to the *MVME2400 Series VME Processor Module Programmer's Reference Guide*, listed in Appendix D, *Related Documentation*.

For programming information about the PMCs, refer to the applicable user's manual furnished with the PMCs.

Memory Maps

There are multiple buses on the MVME240x and each bus domain has its own view of the memory map. The following sections describe the MVME240x memory organization from the following three points of view:

- The mapping of all resources as viewed by the MPU (processor bus memory map)
- The mapping of onboard resources as viewed by PCI local bus masters (PCI bus memory map)
- The mapping of onboard resources as viewed by VMEbus masters (VMEbus memory map)

Additional, more detailed memory maps can be found in the *MVME2400 Series VME Processor Module Programmer's Reference Guide*.

Processor Bus Memory Map

The processor memory map configuration is under the control of the PHB and SMC portions of the Hawk ASIC. The Hawk adjusts system mapping to suit a given application via programmable map decoder registers. At system power-up or reset, a default processor memory map takes over.

Default Processor Memory Map

The default processor memory map that is valid at power-up or reset remains in effect until reprogrammed for specific applications. The table below defines the entire default map ($00000000 to $FFFFFFFF).

Table 4-1. Processor Default View of the Memory Map

Processor Address		Size	Definition
Start	End		
00000000	7FFFFFFF	2GB	Not Mapped
80000000	8001FFFF	128KB	PCI/ISA I/O Space
80020000	FEF7FFFF	2GB-16MB-640KB	Not Mapped
FEF80000	FEF8FFFF	64KB	SMC Registers
FEF90000	FEFEFFFF	384KB	Not Mapped
FEFF0000	FEFFFFFF	64KB	PHB Registers
FF000000	FFEFFFFF	15MB	Not Mapped
FFF00000	FFFFFFFF	1MB	Flash Bank A or Bank B **(See Not**e)

Note The first 1MB of Flash bank A (soldered Flash up to 8MB) appears in this range after a reset if the **rom_b_rv** control bit in the SMC's ROM B Base/Size register is cleared. If the **rom_b_rv** control bit is set, this address range maps to Flash bank B (socketed 1MB Flash).

For detailed processor memory maps, including suggested CHRP- and PREP-compatible memory maps, refer to the *MVME2400 Series VME Processor Module Programmer's Reference Guide.*

PCI Local Bus Memory Map

The PCI memory map is controlled by the MPU/PCI bus bridge controller portion of the Hawk ASIC and by the Universe PCI/VME bus bridge ASIC. The Hawk and Universe devices adjust system mapping to suit a given application via programmable map decoder registers.

No default PCI memory map exists. Resetting the system turns the PCI map decoders off, and they must be reprogrammed in software for the intended application.

For detailed PCI memory maps, including suggested CHRP- and PREP-compatible memory maps, refer to the *MVME2400 Series VME Processor Module Programmer's Reference Guide*.

VMEbus Memory Map

The VMEbus is programmable. Like other parts of the MVME240x memory map, the mapping of local resources as viewed by VMEbus masters varies among applications.

The Universe PCI/VME bus bridge ASIC includes a user-programmable map decoder for the VMEbus-to-local-bus interface. The address translation capabilities of the Universe enable the processor to access any range of addresses on the VMEbus.

Recommendations for VMEbus mapping, including suggested CHRP- and PREP-compatible memory maps, can be found in the *MVME2400 Series VME Processor Module Programmer's Reference Guide*. Figure 4-1 shows the overall mapping approach from the standpoint of a VMEbus master.

Programming Considerations

Good programming practice dictates that only one MPU at a time have control of the MVME240x control registers. Of particular note are:

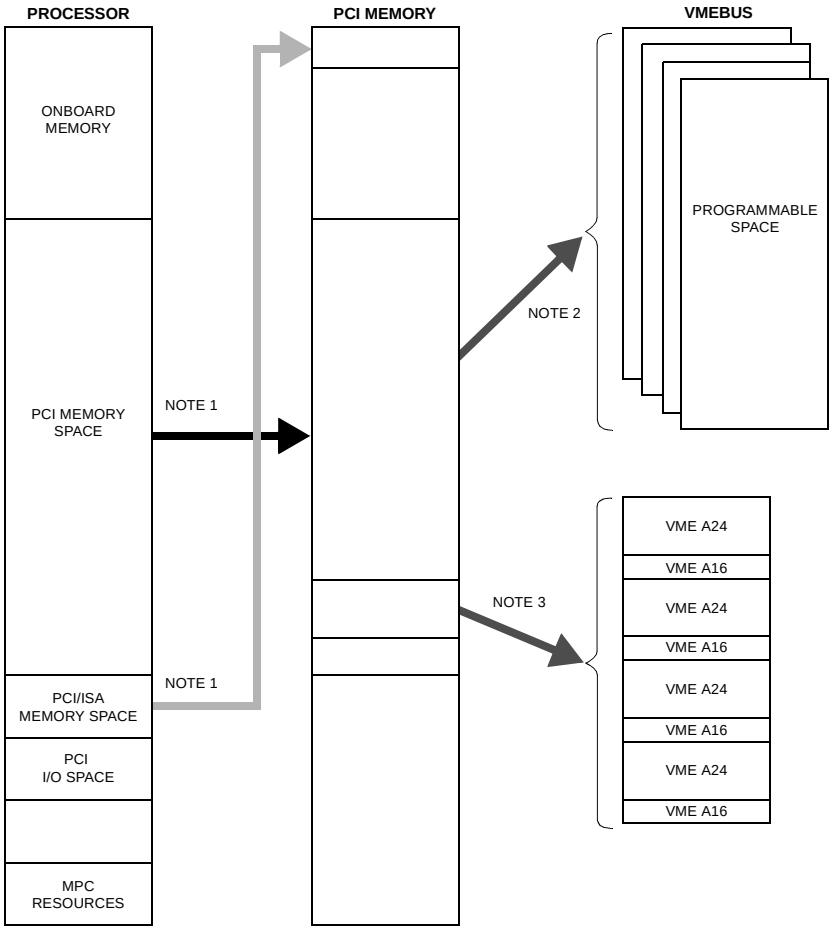
- Registers that modify the address map
- Registers that require two cycles to access
- VMEbus interrupt request registers

PCI Arbitration

There are seven potential PCI bus masters on the MVME240x:

- Hawk ASIC (MPU/PCI bus bridge controller)
- Winbond W83C553 PIB (PCI/ISA bus bridge controller)
- DECchip 21143 Ethernet controller
- Universe II ASIC (PCI/VME bus bridge controller)
- PMC Slot 1 (PCI mezzanine card)
- PMC Slot 2 (PCI mezzanine card)
- PCI Expansion Slot

The Winbond W83C553 PIB device supplies the PCI arbitration support for these seven types of devices. The PIB supports flexible arbitration modes of fixed priority, rotating priority, and mixed priority, as appropriate in a given application. Details on PCI arbitration can be found in the *MVME2400 Series VME Processor Module Programmer's Reference Guide*.



NOTES: 1. Programmable mapping done by Hawk ASIC.
2. Programmable mapping performed via PCI Slave images in Universe ASIC.
3. Programmable mapping performed via Special Slave image (SLSI) in Universe ASIC.

11553.00 9609

Figure 4-1. VMEbus Master Mapping

The arbitration assignments for the MVME240x are shown in the following table:

Table 4-2. PCI Arbitration Assignments

PCI Bus Request	PCI Master(s)
PIB (Internal)	PIB
CPU	Hawk ASIC
Request 0	PMC Slot 2
Request 1	PMC Slot 1
Request 2	PCI Expansion Slot
Request 3	Ethernet
Request 4	Universe ASIC (VMEbus)

Interrupt Handling

The Hawk ASIC, which controls the PHB (PCI Host Bridge) and the MPU/local bus interface functions on the MVME240x, performs interrupt handling as well. Sources of interrupts may be any of the following:

- The Hawk ASIC itself (timer interrupts, transfer error interrupts, or memory error interrupts)
- The processor (processor self-interrupts)
- The PCI bus (interrupts from PCI devices)
- The ISA bus (interrupts from ISA devices)

Figure 4-2 illustrates interrupt architecture on the MVME240x. For details on interrupt handling, refer to the *MVME2400 Series VME Processor Module Programmer's Reference Guide*.

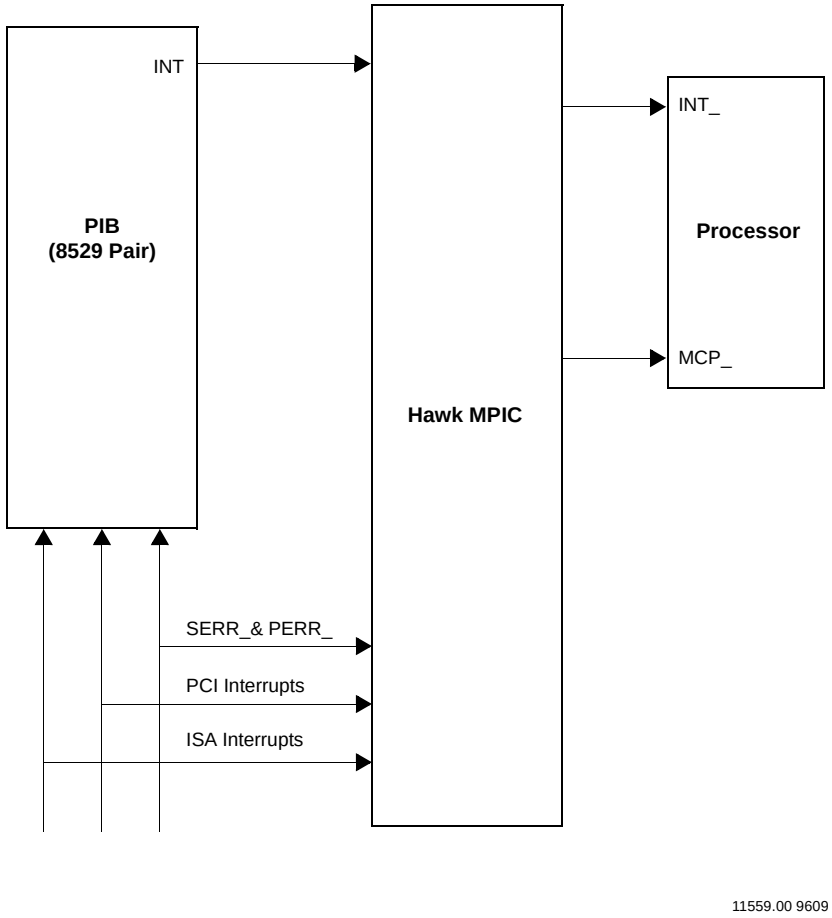
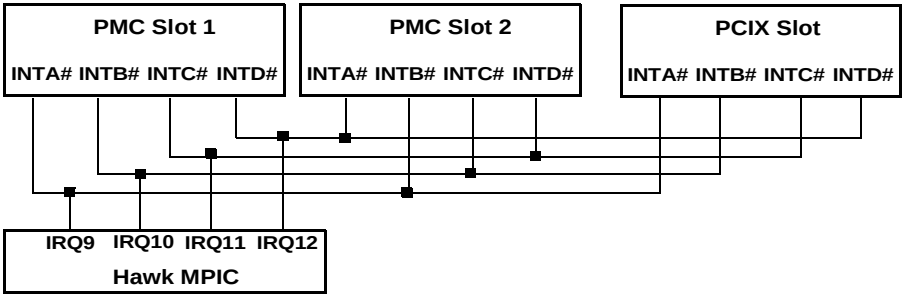


Figure 4-2. MVME240x Interrupt Architecture

The MVME240x routes the interrupts from the PMCs and PCI expansion slots as follows:



DMA Channels

The PIB supports seven DMA channels. They are not functional on the MVME240x.

Sources of Reset

The MVME240x has eight potential sources of reset:

1. Power-on reset
2. **RST** switch (resets the VMEbus when the MVME240x is system controller)
3. Watchdog timer Reset function controlled by the SGS-Thomson MK48T59 TIMEKEEPER device (resets the VMEbus when the MVME240x is system controller)
4. ALT_RST* function controlled by the Port 92 register in the PIB (resets the VMEbus when the MVME240x is system controller)
5. PCI/ISA I/O Reset function controlled by the Clock Divisor register in the PIB
6. The VMEbus SYSRESET* signal

7. VMEbus Reset sources from the Universe ASIC (PCI/VME bus bridge controller): the System Software reset and Local Software reset

Note On the MVME2400, Watchdog Timer 2 is a source of reset *only* if component R25 is installed on the board. Consult your local Motorola Computer Group (MCG) sales representative if this feature needs to be enabled.

Table 4-3 shows which devices are affected by the various types of resets. For details on using resets, refer to the *MVME2400 Series VME Processor Module Programmer's Reference Guide*.

Table 4-3. Classes of Reset and Effectiveness

Device Affected / Reset Source	Processor	Hawk ASIC	PCI Devices	ISA Devices	VMEbus (as system controller
Power-On reset	√	√	√	√	√
Reset switch	√	√	√	√	√
Watchdog reset	√	√	√	√	√
VME SYSRESET*signal	√	√	√	√	√
VME System SW reset	√	√	√	√	√
VME Local SW reset	√	√	√	√	
Hot reset (Port 92)	√	√	√	√	
PCI/ISA reset			√	√	

Endian Issues

The MVME240x supports both little-endian (for example, Windows NT) and big-endian (for example, AIX) software. The PowerPC processor and the VMEbus are inherently big-endian, while the PCI bus is inherently little-endian. The following sections summarize how the MVME240x handles software and hardware differences in big- and little-endian operations. For further details on endian considerations, refer to the *MVME2400 Series VME Processor Module Programmer's Reference Guide*.

Processor/Memory Domain

The MPC750 processor can operate in both big-endian and little-endian mode. However, it always treats the external processor/memory bus as big-endian by performing *address rearrangement and reordering* when running in little-endian mode. The MPC registers in the Hawk MPU/PCI bus bridge controller, SMC memory controller, as well as DRAM, Flash, and system registers, always appear as big-endian.

Role of the Hawk ASIC

Because the PCI bus is little-endian, the PHB portion of the Hawk performs byte swapping in both directions (from PCI to memory and from the processor to PCI) to maintain address invariance while programmed to operate in big-endian mode with the processor and the memory subsystem.

In little-endian mode, the PHB *reverse-rearranges* the address for PCI-bound accesses and *rearranges* the address for memory-bound accesses (from PCI). In this case, no byte swapping is done.

PCI Domain

The PCI bus is inherently little-endian. All devices connected directly to the PCI bus operate in little-endian mode, regardless of the mode of operation in the processor's domain.

PCI and Ethernet

Ethernet is byte-stream-oriented; the byte having the lowest address in memory is the first one to be transferred regardless of the endian mode. Since the PHB maintains address invariance in both little-endian and big-endian mode, no endian issues should arise for Ethernet data. Big-endian software must still take the byte-swapping effect into account when accessing the registers of the PCI/Ethernet device.

Role of the Universe ASIC

Because the PCI bus is little-endian while the VMEbus is big-endian, the Universe PCI/VME bus bridge ASIC performs byte swapping in both directions (from PCI to VMEbus and from VMEbus to PCI) to maintain address invariance, regardless of the mode of operation in the processor's domain.

VMEbus Domain

The VMEbus is inherently big-endian. All devices connected directly to the VMEbus must operate in big-endian mode, regardless of the mode of operation in the processor's domain.

In big-endian mode, byte-swapping is performed first by the Universe ASIC and then by the PHB. The result is transparent to big-endian software (a desirable effect).

In little-endian mode, however, software must take the byte-swapping effect of the Universe ASIC and the address *reverse-rearranging* effect of the PHB into account.

For further details on endian considerations, refer to the *MVME2400 Series VME Processor Module Programmer's Reference Guide*.

PPCBug Overview

The PPCBug firmware is the layer of software just above the hardware. The firmware provides the proper initialization for the devices on the MVME240x module upon power-up or reset.

This chapter describes the basics of the PPCBug and its architecture, along with the monitor (interactive command portion of the firmware), and gives information on using the PPCBug and the special commands. A complete list of PPCBug commands appears at the end of the chapter.

Chapter 6, *Environment Modification* contains information about the CNFG and ENV commands, system calls, and other advanced user topics.

For full user information about PPCbug, refer to the *PPCBug Firmware Package User's Manual* and the *PPCBug Diagnostics Manual*, listed in Appendix D, *Related Documentation*.

PPCBug Basics

The PowerPC debug firmware, PPCBug, is a powerful evaluation and debugging tool for systems built around the Motorola PowerPC microcomputers. Facilities are available for loading and executing user programs under complete operator control for system evaluation.

PPCBug provides a high degree of functionality, user friendliness, portability, and ease of maintenance.

It achieves good portability and comprehensibility because it was written entirely in the C programming language, except where necessary to use assembler functions.

PPCBug includes commands for:

- Display and modification of memory
- Breakpoint and tracing capabilities
- A powerful assembler and disassembler useful for patching programs
- A self-test at power-up feature which verifies the integrity of the system

PPCBug consists of three parts:

1. A command-driven, user-interactive *software debugger*, described in the *PPCBug Firmware Package User's Manual*. It is hereafter referred to as "the debugger" or "PPCBug."
2. A command-driven *diagnostics package* for the MVME240x hardware, hereafter referred to as "the diagnostics." The diagnostics package is described in the *PPCBug Diagnostics Manual*.
3. A *user interface* or *debug*/*diagnostics monitor* that accepts commands from the system console terminal.

When using PPCBug, you operate out of either the *debugger directory* or the *diagnostic directory* if:

- You are in the debugger directory, the debugger prompt `PPC4-Bug>` is displayed and you have all of the debugger commands at your disposal.
- You are in the diagnostic directory, the diagnostic prompt PPC4-Diag> is displayed and you have all of the diagnostic commands at your disposal as well as all of the debugger commands.

Because PPCBug is command-driven, it performs its various operations in response to user commands entered at the keyboard. When you enter a command, PPCBug executes the command and the prompt reappears. However, if you enter a command that causes execution of user target code (for example, **GO**), then control may or may not return to PPCBug, depending on the outcome of the user program.

Memory Requirements

PPCBug requires a maximum of 768KB of read/write memory (that is, DRAM). The debugger allocates this space from the top of memory. For example, a system containing 64MB ($04000000) of read/write memory will place the PPCBug memory page at locations $03F40000 to $03FFFFFF.

PPCBug Implementation

PPCBug is written largely in the C programming language, providing benefits of portability and maintainability. Where necessary, assembly language has been used in the form of separately compiled program modules containing only assembler code. No mixed-language modules are used.

Physically, PPCBug is contained in two socketed 32-pin PLCC Flash devices that together provide 1MB of storage. The executable code is checksummed at every power-on or reset firmware entry, and the result (which includes a precalculated checksum contained in the Flash devices), is verified against the expected checksum.

MPU, Hardware, and Firmware Initialization

The debugger performs the MPU, hardware, and firmware initialization process. This process occurs each time the MVME240x is reset or powered up. The steps below are a high-level outline; not all of the detailed steps are listed.

1. Sets MPU.MSR to known value.
2. Invalidates the MPU's data/instruction caches.
3. Clears all segment registers of the MPU.
4. Clears all block address translation registers of the MPU.
5. Initializes the MPU-bus-to-PCI-bus bridge device.
6. Initializes the PCI-bus-to-ISA-bus bridge device.

7. Calculates the external bus clock speed of the MPU.
8. Delays for 750 milliseconds.
9. Determines the CPU base board type.
10. Sizes the local read/write memory (that is, DRAM).
11. Initializes the read/write memory controller. Sets base address of memory to $00000000.
12. Retrieves the speed of read/write memory.
13. Initializes the read/write memory controller with the speed of read/write memory.
14. Retrieves the speed of read only memory (that is, Flash).
15. Initializes the read only memory controller with the speed of read only memory.
16. Enables the MPU's instruction cache.
17. Copies the MPU's exception vector table from $FFF00000 to $00000000.
18. Verifies MPU type.
19. Enables the superscalar feature of the MPU (superscalar processor boards only).
20. Verifies the external bus clock speed of the MPU.
21. Determines the debugger's console/host ports, and initializes the PC16550A.
22. Displays the debugger's copyright message.
23. Displays any hardware initialization errors that may have occurred.
24. Checksums the debugger object, and displays a warning message if the checksum failed to verify.
25. Displays the amount of local read/write memory found.

26. Verifies the configuration data that is resident in NVRAM, and displays a warning message if the verification failed.
27. Calculates and displays the MPU clock speed, verifies that the MPU clock speed matches the configuration data, and displays a warning message if the verification fails.
28. Displays the BUS clock speed, verifies that the BUS clock speed matches the configuration data, and displays a warning message if the verification fails.
29. Probes PCI bus for supported network devices.
30. Probes PCI bus for supported mass storage devices.
31. Initializes the memory/IO addresses for the supported PCI bus devices.
32. Executes Self-Test, if so configured. (Default is no Self-Test.)
33. Extinguishes the board fail LED, if Self-Test passed, and outputs any warning messages.
34. Executes boot program, if so configured. (Default is no boot.)
35. Executes the debugger monitor (that is, issues the `PPC4-Bug>` prompt).

Using PPCBug

PPCBug is command-driven; it performs its various operations in response to commands that you enter at the keyboard. When the `PPC4-Bug` prompt appears on the screen, the debugger is ready to accept debugger commands. When the `PPC4-Diag` prompt appears on the screen, the debugger is ready to accept diagnostics commands. To switch from one mode to the other, enter **SD**.

What you key in is stored in an internal buffer. Execution begins only after you press the Return or Enter key. This allows you to correct entry errors, if necessary, with the control characters described in the *PPCBug Firmware Package User's Manual.*

After the debugger executes the command, the prompt reappears. However, if the command causes execution of user target code (for example **GO**) then control may or may not return to the debugger, depending on what the user program does. For example, if a breakpoint has been specified, then control returns to the debugger when the breakpoint is encountered during execution of the user program. Alternately, the user program could return to the debugger by means of the System Call Handler routine RETURN (described in the *PPCBug Firmware Package User's Manual*). For more about this, refer to the **GD**, **GO**, and **GT** command descriptions in the *PPCBug Firmware Package User's Manual*.

A debugger command is made up of the following parts:

- The command name, either uppercase or lowercase (for example, **MD** or **md**).
- Any required arguments, as specified by command.
- At least one space before the first argument. Precede all other arguments with either a space or comma.
- One or more options. Precede an option or a string of options with a semicolon (**;**). If no option is entered, the command's default option conditions are used.

Debugger Commands

The individual debugger commands are listed in the following table. The commands are described in detail in the *PPCBug Firmware Package User's Manual*.

Note You can list all the available debugger commands by entering the Help (**HE**) command alone. You can view the syntax for a particular command by entering **HE** and the command mnemonic, as listed below.

Table 5-1. Debugger Commands

Command	Description
AS	One Line Assembler
BC	Block of Memory Compare
BF	Block of Memory Fill
BI	Block of Memory Initialize
BM	Block of Memory Move
BR	Breakpoint Insert
NOBR	Breakpoint Delete
BS	Block of Memory Search
BV	Block of Memory Verify
CACHE	Modify Cache State
CM	Concurrent Mode
NOCM	No Concurrent Mode
CNFG	Configure Board Information Block
CS	Checksum
CSAR	PCI Configuration Space READ Access
CSAW	PCI Configuration Space WRITE Access
DC	Data Conversion
DS	One Line Disassembler
DU	Dump S-Records
ECHO	Echo String
ENV	Set Environment
FORK	Fork Idle MPU at Address
FORKWR	Fork Idle MPU with Registers

Table 5-1. Debugger Commands (Continued)

Command	Description
GD	Go Direct (Ignore Breakpoints)
GEVBOOT	Global Environment Variable Boot
GEVDEL	Global Environment Variable Delete
GEVDUMP	Global Environment Variable(s) Dump
GEVEDIT	Global Environment Variable Edit
GEVINIT	Global Environment Variable Initialization
GEVSHOW	Global Environment Variable(s) Display
GN	Go to Next Instruction
G, GO	Go Execute User Program
GT	Go to Temporary Breakpoint
HE	Help
IDLE	Idle Master MPU
IOC	I/O Control for Disk
IOI	I/O Inquiry
IOP	I/O Physical (Direct Disk Access)
IOT	I/O Teach for Configuring Disk Controller
IRD	Idle MPU Register Display
IRM	Idle MPU Register Modify
IRS	Idle MPU Register Set
LO	Load S-Records from Host
MA	Macro Define/Display
NOMA	Macro Delete
MAE	Macro Edit
MAL	Enable Macro Listing
NOMAL	Disable Macro Listing
MAR	Load Macros
MAW	Save Macros

Table 5-1. Debugger Commands (Continued)

Command	Description
MD, MDS	Memory Display
MENU	System Menu
M, MM	Memory Modify
MMD	Memory Map Diagnostic
MS	Memory Set
MW	Memory Write
NAB	Automatic Network Boot
NAP	Nap MPU
NBH	Network Boot Operating System, Halt
NBO	Network Boot Operating System
NIOC	Network I/O Control
NIOP	Network I/O Physical
NIOT	Network I/O Teach (Configuration)
NPING	Network Ping
OF	Offset Registers Display/Modify
PA	Printer Attach
NOPA	Printer Detach
PBOOT	Bootstrap Operating System
PF	Port Format
NOPF	Port Detach
PFLASH	Program FLASH Memory
PS	Put RTC into Power Save Mode
RB	ROMboot Enable
NORB	ROMboot Disable
RD	Register Display
REMOTE	Remote
RESET	Cold/Warm Reset

Table 5-1. Debugger Commands (Continued)

Command	Description
RL	Read Loop
RM	Register Modify
RS	Register Set
RUN	MPU Execution/Status
SD	Switch Directories
SET	Set Time and Date
SROM	SROM Examine/Modify
SYM	Symbol Table Attach
NOSYM	Symbol Table Detach
SYMS	Symbol Table Display/Search
T	Trace
TA	Terminal Attach
TIME	Display Time and Date
TM	Transparent Mode
TT	Trace to Temporary Breakpoint
VE	Verify S-Records Against Memory
VER	Revision/Version Display
WL	Write Loop

Caution Although a command to allow the erasing and reprogramming of Flash memory is available to you, keep in mind that reprogramming any portion of Flash memory will erase everything currently contained in Flash, including the PPCBug debugger.

Note Both banks A and B of Flash contain the PPCBug debugger.

Diagnostic Tests

The PPCBug hardware diagnostics are intended for testing and troubleshooting the MVME240x module.

In order to use the diagnostics, you must switch to the diagnostic directory. You may switch between directories by using the **SD** (Switch Directories) command. You may view a list of the commands in the directory that you are currently in by using the **HE** (Help) command.

If you are in the debugger directory, the debugger prompt `PPC4-Bug>` displays, and all of the debugger commands are available. Diagnostics commands cannot be entered at the `PPC4-Bug>` prompt.

If you are in the diagnostic directory, the diagnostic prompt `PPC4-Diag>` displays, and all of the debugger and diagnostic commands are available.

PPCBug's diagnostic test groups are listed in the Table 5-2. Note that not all tests are performed on the MVME240x. Using the **HE** command, you can list the diagnostic routines available in each test group. Refer to the *PPCBug Diagnostics Manual* for complete descriptions of the diagnostic routines and instructions on how to invoke them.

Table 5-2. Diagnostic Test Groups

Test Group	Description
CL1283	Parallel Interface (CL1283) Tests*
DEC	DEC21x43 Ethernet Controller Tests
HAWK	HAWK Tests
ISABRDGE	PCI/ISA Bridge Tests
KBD8730x	PC8730x Keyboard/Mouse Tests*
L2CACHE	Level 2 Cache Tests
NCR	NCR 53C8xx SCSI-2 I/O Processor Tests
PAR8730x	Parallel Interface (PC8730x) Test*
UART	Serial Input/Output Tests
PCIBUS	PCI/PMC Generic Tests
RAM	Local RAM Tests
RTC	MK48Txx Timekeeping Tests
SCC	Serial Communications Controller (Z85C230) Tests*
VGA54xx	VGA Controller (GD54xx) Tests
VME3	VME3 (Universe) Tests
Z8536	Z8536 Counter/Timer Tests*

Notes

1. You may enter command names in either uppercase or lowercase.
2. Some diagnostics depend on restart defaults that are set up only in a particular restart mode. Refer to the documentation on a particular diagnostic for the correct mode.
3. Test Sets marked with an asterisk (*) are not available on the MVME240x.

Environment Modification 6

Overview

You can use the factory-installed debug monitor, PPCBug, to modify certain parameters contained in the MVME240x's Non-Volatile RAM (NVRAM), also known as Battery Backed-up RAM (BBRAM).

- The Board Information Block in NVRAM contains various elements concerning operating parameters of the hardware. Use the PPCBug command **CNFG** to change those parameters.
- Use the PPCBug command **ENV** to change configurable PPCBug parameters in NVRAM.

The **CNFG** and **ENV** commands are both described in the *PPCBug Firmware Package User's Manual*, listed in Appendix D, *Related Documentation*. Refer to that manual for general information about their use and capabilities.

The following paragraphs present additional information about **CNFG** and **ENV** that is specific to the PPCBug debugger, along with the parameters that can be configured with the **ENV** command.

CNFG – Configure Board Information Block

Use this command to display and configure the Board Information Block, which is resident within the NVRAM. The board information block contains various elements detailing specific operational parameters of the MVME240x. The board structure for the MVME240x is as shown in the following example:

```
Board (PWA) Serial Number     = “MOT00xxxxxxx   ”
Board Identifier              = “MVME2400       ”
Artwork (PWA) Identifier      = “01-w3394FxxC   ”
MPU Clock Speed               = “350            ”
Bus Clock Speed               = “100            ”
Ethernet Address              = 08003E20C983
Primary SCSI Identifier       = “07”
System Serial Number          = “nnnnnnn        ”
System Identifier             = “Motorola MVME2400”
License Identifier            = “nnnnnnnn “
```

The parameters that are quoted are left-justified character (ASCII) strings padded with space characters, and the quotes (“) are displayed to indicate the size of the string. Parameters that are not quoted are considered data strings, and data strings are right-justified. The data strings are padded with zeroes if the length is not met.

The Board Information Block is factory-configured before shipment. There is no need to modify block parameters unless the NVRAM is corrupted.

Refer to the *MVME2400 Series VME Processor Module Programmer’s Reference Guide*, listed in Appendix D, *Related Documentation*, for the actual location and other information about the Board Information Block.

Refer to the *PPCBug Firmware Package User's Manual* for a description of CNFG and examples.

ENV – Set Environment

Use the **ENV** command to view and/or configure interactively all PPCBug operational parameters that are kept in Non-Volatile RAM (NVRAM).

Refer to the *PPCBug Firmware Package User's Manual* for a description of the use of **ENV**. Additional information on registers in the Universe ASIC that affect these parameters is contained in your *MVME2400 Series VME Processor Module Programmer's Reference Guide*.

Listed and described below are the parameters that you can configure using **ENV**. The default values shown were those in effect when this publication went to print.

Configuring the PPCBug Parameters

The parameters that can be configured using **ENV** are:

```
Bug or System environment [B/S] = B?
```

B Bug is the mode where no system type of support is displayed. However, system-related items are still available. (Default)

S System is the standard mode of operation, and is the default mode if NVRAM should fail. System mode is defined in the *PPCBug Firmware Package User's Manual*.

```
Field Service Menu Enable [Y/N] = N?
```

Y Display the field service menu.

N Do not display the field service menu. (Default)

```
Remote Start Method Switch [G/M/B/N] = B?
```

The Remote Start Method Switch is used when the MVME2400 is cross-loaded from another VME-based CPU, to start execution of the cross-loaded program.

G Use the Global Control and Status Register to pass and start execution of the cross-loaded program. *This selection is not applicable to the MVME2400 boards.*

M Use the Multiprocessor Control Register (MPCR) in shared RAM to pass and start execution of the cross-loaded program.

B Use both the GCSR and the MPCR methods to pass and start execution of the cross-loaded program. (Default)

N Do not use any Remote Start Method.

```
Probe System for Supported I/O Controllers [Y/N] = Y?
```

Y Accesses will be made to the appropriate system buses (for example, VMEbus, local MPU bus) to determine the presence of supported controllers. (Default)

N Accesses will not be made to the VMEbus to determine the presence of supported controllers.

```
Auto-Initialize of NVRAM Header Enable [Y/N] = Y?
```

Y NVRAM (PReP partition) header space will be initialized automatically during board initialization, but only if the PReP partition fails a sanity check. (Default)

N NVRAM header space will not be initialized automatically during board initialization.

```
Network PReP-Boot Mode Enable [Y/N] = N?
```

Y Enable PReP-style network booting (same boot image from a network interface as from a mass storage device).

N Do not enable PReP-style network booting. (Default)

```
Negate VMEbus SYSFAIL* Always [Y/N] = N?
```

Y Negate the VMEbus SYSFAIL* signal during board initialization.

N Negate the VMEbus SYSFAIL* signal after successful completion or entrance into the bug command monitor. (Default)

```
SCSI Bus Reset on Debugger Startup [Y/N] = N?
```

Y Local SCSI bus is reset on debugger setup.

N Local SCSI bus is not reset on debugger setup. (Default)

```
Primary SCSI Bus Negotiations Type [A/S/N] = A?
```

A Asynchronous SCSI bus negotiation. (Default)

S Synchronous SCSI bus negotiation.

N None.

```
Primary SCSI Data Bus Width [W/N] = N?
```

W Wide SCSI (16-bit bus).

N Narrow SCSI (8-bit bus). (Default)

```
Secondary SCSI identifier = 07?
```

Select the identifier. (Default = 07.)

```
NVRAM Bootlist (GEV.fw-boot-path) Boot Enable [Y/N] = N?
```

Y Give boot priority to devices defined in the *fw-boot-path* global environment variable (GEV).

N Do not give boot priority to devices listed in the *fw-boot-path* GEV. (Default)

Note When enabled, the GEV (Global Environment Variable) boot takes priority over all other boots, including Autoboot and Network Boot.


```
NVRAM Bootlist (GEV.fw-boot-path) Boot at power-up only [Y/N] = N?
```

Y Give boot priority to devices defined in the *fw-boot-path* GEV at power-up reset only.

N Give power-up boot priority to devices listed in the *fw-boot-path* GEV at any reset. (Default)

```
NVRAM Bootlist (GEV.fw-boot-path) Boot Abort Delay = 5?
```

The time in seconds that a boot from the NVRAM boot list will delay before starting the boot. The purpose for the delay is to allow you the option of stopping the boot by use of the **BREAK** key. The time value is from 0-255 seconds. (Default = 5 seconds)

```
Auto Boot Enable [Y/N]   = N?
```

Y The Autoboot function is enabled.

N The Autoboot function is disabled. (Default)

```
Auto Boot at power-up only [Y/N] = N?
```

Y Autoboot is attempted at power-up reset only.

N Autoboot is attempted at any reset. (Default)

```
Auto Boot Scan Enable [Y/N]   = Y?
```

Y If Autoboot is enabled, the Autoboot process attempts to boot from devices specified in the scan list (e.g., FDISK/CDROM/TAPE/HDISK). (Default)

N If Autoboot is enabled, the Autoboot process uses the Controller LUN and Device LUN to boot.

```
Auto Boot Scan Device Type List = FDISK/CDROM/TAPE/HDISK?
```

This is the listing of boot devices displayed if the Autoboot Scan option is enabled. If you modify the list, follow the format shown above (uppercase letters, using forward slash as separator).

```
Auto Boot Controller LUN   = 00?
```

Refer to the *PPCBug Firmware Package User's Manual* for a listing of disk/tape controller modules currently supported by PPCBug. (Default = $00)

```
Auto Boot Device LUN    = 00?
```

Refer to the *PPCBug Firmware Package User's Manual* for a listing of disk/tape devices currently supported by PPCBug.
(Default = $00)

```
Auto Boot Partition Number = 00?
```

Which disk "partition" is to be booted, as specified in the PowerPC Reference Platform (PRP) specification. If set to zero, the firmware will search the partitions in order (1, 2, 3, 4) until it finds the first "bootable" partition. That is then the partition that will be booted. Other acceptable values are 1, 2, 3, or 4. In these four cases, the partition specified will be booted without searching.

```
Auto Boot Abort Delay = 7?
```

The time in seconds that the Autoboot sequence will delay before starting the boot. The purpose for the delay is to allow you the option of stopping the boot by use of the <**Break**> key. The time value is from 0-255 seconds. (Default = 7 seconds)

```
Auto Boot Default String [NULL for an empty string] = ?
```

You may specify a string (filename) which is passed on to the code being booted. The maximum length of this string is 16 characters. (Default = null string)

```
ROM Boot Enable [Y/N] = N?
```

Y The ROMboot function is enabled.

N The ROMboot function is disabled. (Default)

```
ROM Boot at power-up only [Y/N] = Y?
```

Y ROMboot is attempted at power-up only. (Default)

N ROMboot is attempted at any reset.

```
ROM Boot Enable search of VMEbus [Y/N] = N?
```

Y VMEbus address space, in addition to the usual areas of memory, will be searched for a ROMboot module.

N VMEbus address space will not be accessed by ROMboot. (Default)

```
ROM Boot Abort Delay = 5?
```

The time in seconds that the ROMboot sequence will delay before starting the boot. The purpose for the delay is to allow you the option of stopping the boot by use of the <**Break**> key. The time value is from 0-255 seconds. (Default = 5 seconds)

```
ROM Boot Direct Starting Address = FFF00000?
```

The first location tested when PPCBug searches for a ROMboot module. (Default = $FFF00000)

```
ROM Boot Direct Ending Address = FFFFFFFC?
```

The last location tested when PPCBug searches for a ROMboot module. (Default = $FFFFFFFC)

```
Network Auto Boot Enable [Y/N] = N?
```

Y The Network Auto Boot (NETboot) function is enabled.

N The NETboot function is disabled. (Default)

```
Network Auto Boot at power-up only [Y/N] = N?
```

Y NETboot is attempted at power-up reset only.

N NETboot is attempted at any reset. (Default)

```
Network Auto Boot Controller LUN = 00?
```

Refer to the *PPCBug Firmware Package User's Manual* for a listing of network controller modules currently supported by PPCBug. (Default = $00)

```
Network Auto Boot Device LUN = 00?
```

Refer to the *PPCBug Firmware Package User's Manual* for a listing of network controller modules currently supported by PPCBug. (Default = $00)

```
Network Auto Boot Abort Delay = 5?
```

The time in seconds that the NETboot sequence will delay before starting the boot. The purpose for the delay is to allow you the option of stopping the boot by use of the <**Break**> key. The time value is from 0-255 seconds. (Default = 5 seconds)

```
Network Auto Boot Configuration Parameters Offset (NVRAM) =
00001000?
```

The address where the network interface configuration parameters are to be saved/retained in NVRAM; these parameters are the necessary parameters to perform an unattended network boot. A typical offset might be $1000, but this value is application-specific. (Default = $00001000)

Caution

If you use the **NIOT** debugger command, these parameters need to be saved somewhere in the offset range $00001000 through $000016F7. The **NIOT** parameters do not exceed 128 bytes in size. The setting of this ENV pointer determines their location. If you have used the same space for your own program information or commands, they will be overwritten and lost.

You can relocate the network interface configuration parameters in this space by using the **ENV** command to change the Network Auto Boot Configuration Parameters Offset from its default of $00001000 to the value you need to be clear of your data within NVRAM.


```
Memory Size Enable [Y/N] = Y?
```

Y	Memory will be sized for Self Test diagnostics. (Default)
N	Memory will not be sized for Self Test diagnostics.

```
Memory Size Starting Address = 00000000?
```

The default Starting Address is $00000000.

```
Memory Size Ending Address = 02000000?
```

The default Ending Address is the calculated size of local memory. If the memory start is changed from $00000000, this value will also need to be adjusted.

```
DRAM Speed in NANO Seconds = 60?
```

The default setting for this parameter will vary depending on the speed of the DRAM memory parts installed on the board. The default is set to the slowest speed found on the available banks of DRAM memory.

```
ROM First Access Length (0 - 31) = 10?
```

This is the value programmed into the “ROMFAL” field (Memory Control Configuration Register 8: bits 23-27) to indicate the number of clock cycles used in accessing the ROM. The lowest allowable ROMFAL setting is $00; the highest allowable is $1F. The value to enter depends on processor speed; refer to Chapter 1, *Hardware*

Preparation and Installation or Appendix A, *Specifications* for appropriate values. The default value varies according to the system's bus clock speed (see note below).

Note ROM First Access Length is not applicable to the MVME2400. The configured value is ignored by PPCBug.

```
ROM Next Access Length (0 - 15) = 0?
```

The value programmed into the "ROMNAL" field (Memory Control Configuration Register 8: bits 28-31) to represent wait states in access time for nibble (or burst) mode ROM accesses. The lowest allowable ROMNAL setting is $0; the highest allowable is $F. The value to enter depends on processor speed; refer to Chapter 1, *Hardware Preparation and Installation* or Appendix A, *Specifications* for appropriate values. The default value varies according to the system's bus clock speed.

Note ROM Next Access Length is not applicable to the MVME2400. The configured value is ignored by PPCBug.

```
DRAM Parity Enable [On-Detection/Always/Never - O/A/N] = O?
```

O	DRAM parity is enabled upon detection. (Default)
A	DRAM parity is always enabled.
N	DRAM parity is never enabled.

Note This parameter (above) also applies to enabling ECC for DRAM.

```
L2 Cache Parity Enable [On-Detection/Always/Never - O/A/N] = O?
```

O L2 Cache parity is enabled upon detection. (Default)

A L2 Cache parity is always enabled.

N L2 Cache parity is never enabled.

```
PCI Interrupts Route Control Registers (PIRQ0/1/2/3) = 0A0B0E0F?
```

Initializes the PIRQx (PCI Interrupts) route control registers in the IBC (PCI/ISA bus bridge controller). The **ENV** parameter is a 32-bit value that is divided by four to yield the values for route control registers PIRQ0/1/2/3. The default is determined by system type. For details on PCI/ISA interrupt assignments and for suggested values to enter for this parameter, refer to the *8259 Interrupts* section of the *MVME2400 Series VME Processor Module Programmer's Reference Guide*.

Note LED/Serial Startup Diagnostic Codes: these codes can be displayed at key points in the initialization of the hardware devices. Should the debugger fail to come up to a prompt, the last code displayed will indicate how far the initialization sequence had progressed before stalling. The codes are enabled by an **ENV** parameter:

```
Serial Startup Code Master Enable [Y/N]=N?
```

A line feed can be inserted after each code is displayed to prevent it from being overwritten by the next code. This is also enabled by an **ENV** parameter:

```
Serial Startup Code LF Enable [Y/N]=N?
```

The list of LED/serial codes is included in the section on *MPU, Hardware, and Firmware Initialization* in the *PPCBug Firmware Package User's Manual*.

Configuring the VMEbus Interface

ENV asks the following series of questions to set up the VMEbus interface for the MVME240x modules. To perform this configuration, you should have a working knowledge of the Universe ASIC as described in your *MVME2400 Series VME Processor Module Programmer's Reference Guide*.

```
VME3PCI Master Master Enable [Y/N] = Y?
```

Y Set up and enable the VMEbus Interface. (Default)

N Do not set up or enable the VMEbus Interface.


```
PCI Slave Image 0 Control = 00000000?
```

The configured value is written into the LSI0_CTL register of the Universe chip.

```
PCI Slave Image 0 Base Address Register = 00000000?
```

The configured value is written into the LSI0_BS register of the Universe chip.

```
PCI Slave Image 0 Bound Address Register = 00000000?
```

The configured value is written into the LSI0_BD register of the Universe chip.

```
PCI Slave Image 0 Translation Offset = 00000000?
```

The configured value is written into the LSI0_TO register of the Universe chip.

```
PCI Slave Image 1 Control = C0820000?
```

The configured value is written into the LSI1_CTL register of the Universe chip.

```
PCI Slave Image 1 Base Address Register = 01000000?
```

The configured value is written into the LSI1_BS register of the Universe chip.

```
PCI Slave Image 1 Bound Address Register = 20000000?
```

The configured value is written into the LSI1_BD register of the Universe chip.

```
PCI Slave Image 1 Translation Offset = 00000000?
```

The configured value is written into the LSI1_TO register of the Universe chip.

```
PCI Slave Image 2 Control = C0410000?
```

The configured value is written into the LSI2_CTL register of the Universe chip.


```
PCI Slave Image 2 Base Address Register = 20000000?
```

The configured value is written into the LSI2_BS register of the Universe chip.

```
PCI Slave Image 2 Bound Address Register = 22000000?
```

The configured value is written into the LSI2_BD register of the Universe chip.

```
PCI Slave Image 2 Translation Offset = D0000000?
```

The configured value is written into the LSI2_TO register of the Universe chip.

```
PCI Slave Image 3 Control = C0400000?
```

The configured value is written into the LSI3_CTL register of the Universe chip.

```
PCI Slave Image 3 Base Address Register = 2FFF0000?
```

The configured value is written into the LSI3_BS register of the Universe chip.

```
PCI Slave Image 3 Bound Address Register = 30000000?
```

The configured value is written into the LSI3_BD register of the Universe chip.

```
PCI Slave Image 3 Translation Offset = D0000000?
```

The configured value is written into the LSI3_TO register of the Universe chip.

```
VMEbus Slave Image 0 Control = E0F20000?
```

The configured value is written into the VSI0_CTL register of the Universe chip.

```
VMEbus Slave Image 0 Base Address Register = 00000000?
```

The configured value is written into the VSI0_BS register of the Universe chip.

```
VMEbus Slave Image 0 Bound Address Register = (Local DRAM Size)?
```

The configured value is written into the VSI0_BD register of the Universe chip. The value is the same as the Local Memory Found number already displayed.


```
VMEbus Slave Image 0 Translation Offset = 80000000?
```

The configured value is written into the VSI0_TO register of the Universe chip.

```
VMEbus Slave Image 1 Control = 00000000?
```

The configured value is written into the VSI1_CTL register of the Universe chip.

```
VMEbus Slave Image 1 Base Address Register = 00000000?
```

The configured value is written into the VSI1_BS register of the Universe chip.

```
VMEbus Slave Image 1 Bound Address Register = 00000000?
```

The configured value is written into the VSI1_BD register of the Universe chip.

```
VMEbus Slave Image 1 Translation Offset = 00000000?
```

The configured value is written into the VSI1_TO register of the Universe chip.

```
VMEbus Slave Image 2 Control = 00000000?
```

The configured value is written into the VSI2_CTL register of the Universe chip.

```
VMEbus Slave Image 2 Base Address Register = 00000000?
```

The configured value is written into the VSI2_BS register of the Universe chip.

```
VMEbus Slave Image 2 Bound Address Register = 00000000?
```

The configured value is written into the VSI2_BD register of the Universe chip.

```
VMEbus Slave Image 2 Translation Offset = 00000000?
```

The configured value is written into the VSI2_TO register of the Universe chip.


```
VMEbus Slave Image 3 Control = 00000000?
```

The configured value is written into the VSI3_CTL register of the Universe chip.

```
VMEbus Slave Image 3 Base Address Register = 00000000?
```

The configured value is written into the VSI3_BS register of the Universe chip.

```
VMEbus Slave Image 3 Bound Address Register = 00000000?
```

The configured value is written into the VSI3_BD register of the Universe chip.

```
VMEbus Slave Image 3 Translation Offset = 00000000?
```

The configured value is written into the VSI3_TO register of the Universe chip.

```
PCI Miscellaneous Register = 10000000?
```

The configured value is written into the LMISC register of the Universe chip.

```
Special PCI Slave Image Register = 00000000?
```

The configured value is written into the SLSI register of the Universe chip.

```
Master Control Register = 80C00000?
```

The configured value is written into the MAST_CTL register of the Universe chip.

```
Miscellaneous Control Register = 52060000?
```

The configured value is written into the MISC_CTL register of the Universe chip.

```
User AM Codes = 00000000?
```

The configured value is written into the USER_AM register of the Universe chip.

Specifications

The following table lists the general specifications for the MVME240x VME processor module. The later sections provide information on cooling requirements and EMC regulatory compliance.

Specifications for the optional PMCs can be found in the documentation for those modules.

Table A-1. Specifications

Characteristics		Specifications
MPU	MPC750 @ 233 MHz	16KB/16KB I/D on-chip cache
	MPC750 @ 350 MHz	32KB/32KB I/D on-chip cache
	MPC750 @ 450 MHz	32KB/32KB I/D on-chip cache
Memory	SDRAM	32MB, 64MB, or 128MB ECC-protected
	Flash	1MB via two 32-pin PLCC sockets
		8MB via surface mount
TOD clock device	M48T559	8KB NVRAM
Timers	One watchdog timer; time-out generates reset	
	Four real-time 16-bit programmable timers	
Power requirements, with no PMCs installed (See Note)	+12Vdc, 0mA –12Vdc, 0mA (typical)	+5Vdc (±5%), 4A typical, 4.75A maximum with MP603
		+5Vdc (±5%), 4.5A typical, 5.5A maximum with MP604
Operating temperature	0° C to 55° C entry air with forced-air cooling (refer to *Cooling Requirements*)	
Storage temperature	–40° C to +85° C	

Table A-1. Specifications (Continued)

Characteristics		Specifications
Relative humidity	10% to 90% NC (Non-Condensing)	
Vibration (operating)	1 G RMS, 20 Hz-2000 Hz random	
Altitude (operating)	5000 meters (16,405 feet)	
Physical dimensions (base board only)	Height	Double-high VME board, 9.2 in. (233 mm)
	Front panel width	0.8 in. (19.8 mm)
	Front panel height	10.3 in. (261.7 mm)
	Depth	6.3 in. (160 mm)
PCI Mezzanine Card (PMC) slots	Address/Data	A32/D32/D64, PMC PN1-4 connectors
	Bus Clock	33 MHz
	Signaling	5V
	Power	7.5 watts maximum per slot (see Note)
	Module types	Basic, single-wide (74.0 mm x 149.0 mm)
		Basic, double-wide, (149.0 mm x 149.0 mm)
	PMC I/O	Front panel and/or VMEbus P2 I/O
PCI expansion connector	Address/Data	A32/D32/D46, 114-pin connector
	PCI bus clock	33 MHz
	Signalling	5V
Peripheral Computer Interface (PCI)	PCI bridge	
	PCIbus, 32-/64-bit, 33 MHz	
VMEbus ANSI/VITA 1-1994 VME64 (previously IEEE STD 1014)	DTB master	A16-A32; D08-D64, BLT
	DTB slave	A24-A32; D08-D64, BLT, UAT
	Arbiter	Round Robin or Priority
	Interrupt handler	IRQ 1-7
	Interrupt controller	Any one of seven
	System controller	Via jumper or auto detect
	Location monitor	Two LMA32
Ethernet interface	DEC 21143 controller with PCI local bus DMA	
	Front panel I/O through RJ-45 connector	

Table A-1. Specifications (Continued)

Characteristics	Specifications
Asynchronous serial debug port	PC16550
	Front panel I/O through RJ-45 connector
Front panel: switches and status indicators	Reset and Abort switches
	Four LEDs: BFL, CPU, PMC (one for PMC slot 2, one for slot 1)

Note The power requirement listed for the MVME240x does not include the power requirements for the PMC slots. The PMC specification allows for 7.5 watts per PMC slot. The 15 watts total can be drawn from any combination of the four voltage sources provided by the MVME240x: +3.3V, +5V, +12V, and –12V.

Cooling Requirements

The MVME240x VME processor Module is designed and tested to operate reliably with an incoming air temperature range from 0° to 55° C (32° to 131° F) with forced air cooling of the entire assembly (base board and modules) at a velocity typically achievable by using a 100 CFM axial fan.

Temperature qualification is performed in a standard Motorola VMEsystem chassis. Twenty-five-watt load boards are inserted in two card slots (one on each side adjacent to the board under test) to simulate a high power density system configuration. An assembly of three axial fans, rated at 100 CFM per fan, is placed directly under the VME card cage. The incoming air temperature is measured between the fan assembly and the card cage, where the incoming airstream first encounters the module under test.

Test software is executed as the module is subjected to ambient temperature variations. Case temperatures of critical, high power density integrated circuits are monitored to ensure component vendors' specifications are not exceeded.

While the exact amount of airflow required for cooling depends on the ambient air temperature and the type, number, and location of boards and other heat sources, adequate cooling can usually be achieved with 10 CFM and 490 LFM flowing over the module. Less airflow is required to cool the module in environments having lower maximum ambients. Under more favorable thermal conditions, it may be possible to operate the module reliably at higher than 55° C with increased airflow.

It is important to note that there are several factors, in addition to the rated CFM of the air mover, which determine the actual volume and speed of air flowing over a module.

EMC Regulatory Compliance

The MVME240x was tested in an EMC-compliant chassis and meets the requirements for EN55022 CE Class B equipment. Compliance was achieved under the following conditions:

- Shielded cables on all external I/O ports.
- Cable shields connected to chassis ground via metal shell connectors bonded to a conductive module front panel.
- Conductive chassis rails connected to chassis ground. This provides the path for connecting shields to chassis ground.
- Front panel screws properly tightened.
- All peripherals were EMC-compliant.

For minimum RF emissions, it is essential that the conditions above be implemented. Failure to do so could compromise the EMC compliance of the equipment containing the module.

The MVME240x is a board level product and meant to be used in standard VME applications. As such, it is the responsibility of the OEM to meet the regulatory guidelines as determined by its application.

Connector Pin Assignments

B

Introduction

This appendix provides the pin assignments for the following groups of interconnect signals for the MVME240x:

Connector		Location	Table
VMEbus connector		P1	Table B-1
VMEbus connector, P2 I/O		P2	Table B-2
Debug serial port, RJ-45		DEBUG (J2)	Table B-3
Ethernet port, RJ-45		10/100BASET (J3)	Table B-4
CPU debug connector		J1	Table B-5
PCI expansion connector		J6	Table B-6
PMC connectors, Slot 1	32-bit PCI	J11, J12	Table B-7
	64-bit PCI extension and P2 I/O	J13, J14	Table B-8
PMC connectors, Slot 2	32-bit PCI	J21, J22	Table B-9
	64-bit PCI extension and P2 I/O	J23, J24	Table B-10

Pin Assignments

The following tables furnish pin assignments only. For detailed descriptions of the various interconnect signals, consult the support information documentation for the MVME240x.

B

VMEbus Connector – P1

Two 160-pin DIN type connectors, P1 and P2, supply the interface between the base board and the VMEbus. P1 provides power and VME signals for 24-bit addressing and 16-bit data. Its pin assignments are set by the IEEE P1014-1987 VMEbus Specification and the VME64 Extension Standard. They are listed in the following table.

Table B-1. P1 VMEbus Connector Pin Assignments

	Row Z	Row A	Row B	Row C	Row D	
1	Not Used	VD0	VBBSY*	VD8	Not Used	1
2	GND	VD1	VBCLR*	VD9	GND	2
3	Not Used	VD2	VACFAIL*	VD10	Not Used	3
4	GND	VD3	VBGIN0*	VD11	Not Used	4
5	Not Used	VD4	VBGOUT0*	VD12	Not Used	5
6	GND	VD5	VBGIN1*	VD13	Not Used	6
7	Not Used	VD6	VBGOUT1*	VD14	Not Used	7
8	GND	VD7	VBGIN2*	VD15	Not Used	8
9	Not Used	GND	VBGOUT2*	GND	VMEGAP*	9
10	GND	VSYSCLK	VBGIN3*	VSYSFAIL*	VMEGA0*	10
11	Not Used	GND	VBGOUT3*	VBERR*	VMEGA1*	11
12	GND	VDS1*	VBR0*	VSYSRESET*	Not Used	12
13	Not Used	VDS0*	VBR1*	VLWORD	VMEGA2*	13
14	GND	VWRITE*	VBR2*	VAM5	Not Used	14
15	Not Used	GND	VBR3*	VA23	VMEGA3*	15
16	GND	VDTACK*	VAM0	VA22	Not Used	16
17	Not Used	GND	VAM1	VA21	VMEGA4*	17
18	GND	VAS*	VAM2	VA20	Not Used	18
19	Not Used	GND	VAM3	VA19	Not Used	19
20	GND	VIACK*	GND	VA18	Not Used	20
21	Not Used	VIACKIN*	VSERCLK	VA17	Not Used	21
22	GND	VIACKOUT*	VSERDAT	VA16	Not Used	22
23	Not Used	VAM4	GND	VA15	Not Used	23
24	GND	VA7	VIRQ7*	VA14	Not Used	24

Table B-1. P1 VMEbus Connector Pin Assignments (Continued)

25	Not Used	VA6	VIRQ6*	VA13	Not Used	25
26	GND	VA5	VIRQ5*	VA12	Not Used	26
27	Not Used	VA4	VIRQ4*	VA11	Not Used	27
28	GND	VA3	VIRQ3*	VA10	Not Used	28
29	Not Used	VA2	VIRQ2*	VA9	Not Used	29
30	GND	VA1	VIRQ1*	VA8	Not Used	30
31	Not Used	–12V	+5VSTDBY	+12V	GND	31
32	GND	+5V	+5V	+5V	Not Used	32

VMEbus Connector – P2

Row B of the P2 connector provides power to the MVME240x, the upper eight VMEbus lines, and additional 16 VMEbus data lines as specified by the VMEbus specification. Rows A, C, Z, and D of the P2 connector provide power and interface signals to a transition module, when one is used. The pin assignments are as follows:

Table B-2. P2 Connector Pin Assignment

	ROW Z	ROW A	ROW B	ROW C	ROW D	
1	PMC2_2 (J24-2)	PMC1_2 (J14-2)	+5V	PMC1_1 (J14-1)	PMC2_1 (J24-1)	1
2	GND	PMC1_4 (J14-4)	GND	PMC1_3 (J14-3)	PMC2_3 (J24-3)	2
3	PMC2_5 (J24-5)	PMC1_6 (J14-6)	RETRY#	PMC1_5 (J14-5)	PMC2_4 (J24-4)	3
4	GND	PMC1_8 (J14-8)	VA24	PMC1_7 (J14-7)	PMC2_6 (J24-6)	4
5	PMC2_8 (J24-8)	PMC1_10 (J14-10)	VA25	PMC1_9 (J14-9)	PMC2_7 (J24-7)	5
6	GND	PMC1_12 (J14-12)	VA26	PMC1_11 (J14-11)	PMC2_9 (J24-9)	6
7	PMC2_11 (J24-11)	PMC1_14 (J14-14)	VA27	PMC1_13 (J14-13)	PMC2_10 (J24-10)	7
8	GND	PMC1_16 (J14-16)	VA28	PMC1_15 (J14-15)	PMC2_12 (J24-12)	8
9	PMC2_14 (J24-14)	PMC1_18 (J14-18)	VA29	PMC1_17 (J14-17)	PMC2_13 (J24-13)	9
10	GND	PMC1_20 (J14-20)	VA30	PMC1_19 (J14-19)	PMC2_15 (J24-15)	10
11	PMC2_17 (J24-17)	PMC1_22 (J14-22)	VA31	PMC1_21 (J14-21)	PMC2_16 (J24-16)	11
12	GND	PMC1_24 (J14-24)	GND	PMC1_23 (J14-23)	PMC2_18 (J24-18)	12

Table B-2. P2 Connector Pin Assignment (Continued)

13	PMC2_20 (J24-20)	PMC1_26 (J14-26)	+5V	PMC1_25 (J14-25)	PMC2_19 (J24-19)	13
14	GND	PMC1_28 (J14-28)	VD16	PMC1_27 (J14-27)	PMC2_21 (J24-21)	14
15	PMC2_23 (J24-23)	PMC1_30 (J14-30)	VD17	PMC1_29 (J14-29)	PMC2_22 (J24-22)	15
16	GND	PMC1_32 (J14-32)	VD18	PMC1_31 (J14-31)	PMC2_24 (J24-24)	16
17	PMC2_26 (J24-26)	PMC1_34 (J14-34)	VD19	PMC1_33 (J14-33)	PMC2_25 (J24-25)	17
18	GND	PMC1_36 (J14-36)	VD20	PMC1_35 (J14-35)	PMC2_27 (J24-27)	18
19	PMC2_29 (J24-29)	PMC1_38 (J14-38)	VD21	PMC1_37 (J14-37)	PMC2_28 (J24-28)	19
20	GND	PMC1_40 (J14-40)	VD22	PMC1_39 (J14-39)	PMC2_30 (J24-30)	20
21	PMC2_32 (J24-32)	PMC1_42 (J14-42)	VD23	PMC1_41 (J14-41)	PMC2_31 (J24-31)	21
22	GND	PMC1_44 (J14-44)	GND	PMC1_43 (J14-43)	PMC2_33 (J24-33)	22
23	PMC2_35 (J24-35)	PMC1_46 (J14-46)	VD24	PMC1_45 (J14-45)	PMC2_34 (J24-34)	23
24	GND	PMC1_48 (J14-48)	VD25	PMC1_47 (J14-47)	PMC2_36 (J24-36)	24
25	PMC2_38 (J24-38)	PMC1_50 (J14-50)	VD26	PMC1_49 (J14-49)	PMC2_37 (J24-37)	25
26	GND	PMC1_52 (J14-52)	VD27	PMC1_51 (J14-51)	PMC2_39 (J24-39)	26
27	PMC2_41 (J24-41)	PMC1_54 (J14-54)	VD28	PMC1_53 (J14-53)	PMC2_40 (J24-40)	27
28	GND	PMC1_56 (J14-56)	VD29	PMC1_55 (J14-55)	PMC2_42 (J24-42)	28
29	PMC2_44 (J24-44)	PMC1_58 (J14-58)	VD30	PMC1_57 (J14-57)	PMC2_43 (J24-43)	29
30	GND	PMC1_60 (J14-60)	VD31	PMC1_59 (J14-59)	PMC2_45 (J24-45)	30
31	PMC2_46 (J24-46)	PMC1_62 (J14-62)	GND	PMC1_61 (J14-61)	GND	31
32	GND	PMC1_64 (J14-64)	+5V	PMC1_63 (J14-63)	VPC	32

Serial Port Connector – DEBUG (J2)

A standard RJ-45 connector located on the front plate of the MVME2040x provides the interface to the asynchronous serial debug port. The pin assignments for this connector are as follows:

Table B-3. DEBUG (J2)Connector Pin Assignments

1	DCD
2	RTS
3	GND
4	TXD
5	RXD
6	GND
7	CTS
8	DTR

Ethernet Connector – 10BaseT (J3)

The 10BaseT/100BaseTx connector is an RJ-45 connector located on the front plate of the MVME240x. The pin assignments for this connector are as follows:

Table B-4. 10/100 BASET (J3) Connector Pin Assignments

1	TD+
2	TD–
3	RD+
4	No Connect
5	No Connect
6	RD–
7	No Connect
8	No Connect

B

CPU Debug Connector – J1

One 190-pin Mictor connector with center row of power and ground pins is used to provide access to the Processor Bus and some miscellaneous signals. The pin assignments for this connector are as follows:

Table B-5. Debug Connector Pin Assignments

1	PA0	GND	PA1	2
3	PA2		PA3	4
5	PA4		PA5	6
7	PA6		PA7	8
9	PA8		PA9	10
11	PA10		PA11	12
13	PA12		PA13	14
15	PA14		PA15	16
17	PA16		PA17	18
19	PA18		PA19	20
21	PA20		PA21	22
23	PA22		PA23	24
25	PA24		PA25	26
27	PA26		PA27	28
29	PA28		PA29	30
31	PA30		PA31	32
33	PAPAR0		PAPAR1	34
35	PAPAR2		PAPAR3	36
37	APE#		RSRV#	38

Table B-5. Debug Connector Pin Assignments (Continued)

39	PD0	+5V	PD1	40
41	PD2		PD3	42
43	PD4		PD5	44
45	PD6		PD7	46
47	PD8		PD9	48
49	PD10		PD11	50
51	PD12		PD13	52
53	PD14		PD15	54
55	PD16		PD17	56
57	PD18		PD19	58
59	PA20		PD21	60
61	PD22		PD23	62
63	PD24		PD25	64
65	PD26		PD27	66
67	PD28		PD29	68
69	PD30		PD31	70
71	PD32		PD33	72
73	PD34		PD35	74
75	PD36		PD37	76

Table B-5. Debug Connector Pin Assignments (Continued)

77	PD38	GND	PD39	78
79	PD40		PD41	80
81	PD42		PD43	82
83	PD44		PD45	84
85	PD46		PD47	86
87	PD48		PD49	88
89	PA50		PD51	90
91	PD52		PD53	92
93	PD54		PD55	94
95	PD56		PD57	96
97	PD58		PD59	98
99	PD60		PD61	100
101	PD62		PD63	102
103	PDPAR0		PDPAR1	104
105	PDPAR2		PDPAR3	106
107	PDPAR4		PDPAR5	108
109	PDPAR6		PDPAR7	110
111				112
113	DPE#		DBDIS#	114

Table B-5. Debug Connector Pin Assignments (Continued)

115	TT0	+3.3V	TSIZ0	116
117	TT1		TSIZ1	118
119	TT2		TSIZ2	120
121	TT3		TC0	122
123	TT4		TC1	124
125	CI#		TC2	126
127	WT#		CSE0	128
129	GLOBAL#		CSE1	130
131	SHARED#		DBWO#	132
133	AACK#		TS#	134
135	ARTY#		XATS#	136
137	DRTY#		TBST#	138
139	TA#			140
141	TEA#			142
143			DBG#	144
145			DBB#	146
147			ABB#	148
149	TCLK_OUT		CPUGNT0#	150
151			CPUREQ0#	152

Table B-5. Debug Connector Pin Assignments (Continued)

153	CPUREQ1#	GND	INT0#	154
155	CPUGNT1#		MCPI#	156
157	INT1#		SMI#	158
159	MCPI1#		CKSTPI#	160
161	L2BR#		CKSTPO#	162
163	L2BG#		HALTED	164
165	L2CLAIM#		TLBISYNC#	166
167			TBEN	168
169			SUSPEND#	170
171			DRVMOD0	172
173			DRVMOD1	174
175			NAPRUN	176
177	SRESET1#		QREQ#	178
179	SRESET0#		QACK#	180
181	HRESET#		TDO	182
183	GND		TDI	184
185	CPUCLK		TCK	186
187	CPUCLK		TMS	188
189	CPUCLK		TRST#	190

PCI Expansion Connector – J6

One 114-pin Mictor connector with center row of power and ground pins is used to provide PCI/PMC expansion capability. The pin assignments for this connector are as follows:

Table B-6. J6 - PCI Expansion Connector Pin Assignments

1	+3.3V	GND	+3.3V	2
3	PCICLK		PMCINTA#	4
5	GND		PMCINTB#	6
7	PURST#		PMCINTC#	8
9	HRESET#		PMCINTD#	10
11	TDO		TDI	12
13	TMS		TCK	14
15	TRST#		PCIXP#	16
17	PCIXGNT#		PCIXREQ#	18
19	+12V		–12V	20
21	PERR#		SERR#	22
23	LOCK#		SDONE	24
25	DEVSEL#		SBO#	26
27	GND		GND	28
29	TRDY#		IRDY#	30
31	STOP#		FRAME#	32
33	GND		GND	34
35	ACK64#		Reserved	36
37	REQ64#		Reserved	38

Table B-6. J6 - PCI Expansion Connector Pin Assignments (Continued)

39	PAR	+5V	PCIRST#	40
41	C/BE1#		C/BE0#	42
43	C/BE3#		C/BE2#	44
45	AD1		AD0	46
47	AD3		AD2	48
49	AD5		AD4	50
51	AD7		AD6	52
53	AD9		AD8	54
55	AD11		AD10	56
57	AD13		AD12	58
59	AD15		AD14	60
61	AD17		AD16	62
63	AD19		AD18	64
65	AD21		AD20	66
67	AD23		AD22	68
69	AD25		AD24	70
71	AD27		AD26	72
73	AD29		AD28	74
75	AD31		AD30	76

Table B-6. J6 - PCI Expansion Connector Pin Assignments (Continued)

77	PAR64	GND	Reserved	78
79	C/BE5#		C/BE4#	80
81	C/BE7#		C/BE6#	82
83	AD33		AD32	84
85	AD35		AD34	86
87	AD37		AD36	88
89	AD39		AD38	90
91	AD41		AD40	92
93	AD43		AD42	94
95	AD45		AD44	96
97	AD47		AD46	98
99	AD49		AD48	100
101	AD51		AD50	102
103	AD53		AD52	104
105	AD55		AD54	106
107	AD57		AD56	108
109	AD59		AD58	110
111	AD61		AD60	112
113	AD63		AD62	114

B

PCI Mezzanine Card Connectors – J11 through J14

Four 64-pin SMT connectors, J11 through J14, supply 32/64-bit PCI interfaces and P2 I/O between the MVME240x board and an optional add-on PCI Mezzanine Card (PMC) in PMC Slot 1. The pin assignments for PMC Slot 1 are listed in the following two tables.

Table B-7. J11 - J12 PMC1 Connector Pin Assignments

	J11				J12		
1	TCK	–12V	2	1	+12V	TRST#	2
3	GND	INTA#	4	3	TMS	TDO	4
5	INTB#	INTC#	6	5	TDI	GND	6
7	PMCPRSNT1#	+5V	8	7	GND	Not Used	8
9	INTD#	Not Used	10	9	Not Used	Not Used	10
11	GND	Not Used	12	11	Pull-up	+3.3V	12
13	CLK	GND	14	13	RST#	Pull-down	14
15	GND	PMCGNT1#	16	15	+3.3V	Pull-down	16
17	PMCREQ1#	+5V	18	17	Not Used	GND	18
19	+5V (Vio)	AD31	20	19	AD30	AD29	20
21	AD28	AD27	22	21	GND	AD26	22
23	AD25	GND	24	23	AD24	+3.3V	24
25	GND	C/BE3#	26	25	IDSEL1	AD23	26
27	AD22	AD21	28	27	+3.3V	AD20	28
29	AD19	+5V	30	29	AD18	GND	30
31	+5V (Vio)	AD17	32	31	AD16	C/BE2#	32
33	FRAME#	GND	34	33	GND	Not Used	34
35	GND	IRDY#	36	35	TRDY#	+3.3V	36
37	DEVSEL#	+5V	38	37	GND	STOP#	38
39	GND	LOCK#	40	39	PERR#	GND	40
41	SDONE#	SBO#	42	41	+3.3V	SERR#	42
43	PAR	GND	44	43	C/BE1#	GND	44
45	+5V (Vio)	AD15	46	45	AD14	AD13	46
47	AD12	AD11	48	47	GND	AD10	48

Table B-7. J11 - J12 PMC1 Connector Pin Assignments (Continued)

49	AD09	+5V	50	49	AD08	+3.3V	50
51	GND	C/BE0#	52	51	AD07	Not Used	52
53	AD06	AD05	54	53	+3.3V	Not Used	54
55	AD04	GND	56	55	Not Used	GND	56
57	+5V (Vio)	AD03	58	57	Not Used	Not Used	58
59	AD02	AD01	60	59	GND	Not Used	60
61	AD00	+5V	62	61	ACK64#	+3.3V	62
63	GND	REQ64#	64	63	GND	Not Used	64

Table B-8. J13 - J14 PMC1 Connector Pin Assignments

	J13				J14		
1	Reserved	GND	2	1	PMC1_1 (P2-C1)	PMC1_2 (P2-A1)	2
3	GND	C/BE7#	4	3	PMC1_3 (P2-C2)	PMC1_4 (P2-A2)	4
5	C/BE6#	C/BE5#	6	5	PMC1_5 (P2-C3)	PMC1_6 (P2-A3)	6
7	C/BE4#	GND	8	7	PMC1_7 (P2-C4)	PMC1_8 (P2-A4)	8
9	+5V (Vio)	PAR64	10	9	PMC1 _9 (P2-C5)	PMC1_10 (P2-A5)	10
11	AD63	AD62	12	11	PMC1_11 (P2-C6)	PMC1_12 (P2-A6)	12
13	AD61	GND	14	13	PMC1_13 (P2-C7)	PMC1_14 (P2-A7)	14
15	GND	AD60	16	15	PMC1_15 (P2-C8)	PMC1_16 (P2-A8)	16
17	AD59	AD58	18	17	PMC1_17 (P2-C9)	PMC1_18 (P2-A9)	18
19	AD57	GND	20	19	PMC1_19 (P2-C10)	PMC1_20 (P2-A10)	20
21	+5V (Vio)	AD56	22	21	PMC1_21 (P2-C11)	PMC1_22 (P2-A11)	22
23	AD55	AD54	24	23	PMC1_23 (P2-C12)	PMC1_24 (P2-A12)	24
25	AD53	GND	26	25	PMC1_25 (P2-C13)	PMC1_26 (P2-A13)	26
27	GND	AD52	28	27	PMC1_27 (P2-C14)	PMC1_28 (P2-A14)	28
29	AD51	AD50	30	29	PMC1_29 (P2-C15)	PMC1_30 (P2-A15)	30
31	AD49	GND	32	31	PMC1_31 (P2-C16)	PMC1_32 (P2-A16)	32
33	GND	AD48	34	33	PMC1_33 (P2-C17)	PMC1_34 (P2-A17)	34
35	AD47	AD46	36	35	PMC1_35 (P2-C18)	PMC1_36 (P2-A18)	36

B

Table B-8. J13 - J14 PMC1 Connector Pin Assignments (Continued)

37	AD45	GND	38	37	PMC1_37 (P2-C19)	PMC1_38 (P2-A19)	38
39	+5V (Vio)	AD44	40	39	PMC1_39 (P2-C20)	PMC1_40 (P2-A20)	40
41	AD43	AD42	42	41	PMC1_41 (P2-C21)	PMC1_42 (P2-A21)	42
43	AD41	GND	44	43	PMC1_43 (P2-C22)	PMC1_44 (P2-A22)	44
45	GND	AD40	46	45	PMC1_45 (P2-C23)	PMC1_46 (P2-A23)	46
47	AD39	AD38	48	47	PMC1_47 (P2-C24)	PMC1_48 (P2-A24)	48
49	AD37	GND	50	49	PMC1_49 (P2-C25)	PMC1_50 (P2-A25)	50
51	GND	AD36	52	51	PMC1_51 (P2-C26)	PMC1_52 (P2-A26)	52
53	AD35	AD34	54	53	PMC1_53 (P2-C27)	PMC1_54 (P2-A27)	54
55	AD33	GND	56	55	PMC1_55 (P2-C28)	PMC1_56 (P2-A28)	56
57	+5V (Vio)	AD32	58	57	PMC1_57 (P2-C29)	PMC1_58 (P2-A29)	58
59	Reserved	Reserved	60	59	PMC1_59 (P2-C30)	PMC1_60 (P2-A30)	60
61	Reserved	GND	62	61	PMC1_61 (P2-C31)	PMC1_62 (P2-A31)	62
63	GND	Reserved	64	63	PMC1_63 (P2-C32)	PMC1_64 (P2-A32)	64

PCI Mezzanine Card Connectors – J21 through J24

Four 64-pin SMT connectors, J21 through J24, supply 32/64-bit PCI interfaces and P2 I/O between the MVME240x board and an optional add-on PCI Mezzanine Card (PMC) in PMC Slot 2. The pin assignments for PMC Slot 2 are listed in the following two tables.

Table B-9. J21 and J22 PMC2 Connector Pin Assignments

	J21				J22		
1	TCK	–12V	2	1	+12V	TRST#	2
3	GND	INTA#	4	3	TMS	TDO	4
5	INTB#	INTC#	6	5	TDI	GND	6
7	PMCPRSNT2#	+5V	8	7	GND	Not Used	8
9	INTD#	Not Used	10	9	Not Used	Not Used	10
11	GND	Not Used	12	11	Pull-up	+3.3V	12
13	CLK	GND	14	13	RST#	Pull-down	14
15	GND	PMCGNT2#	16	15	+3.3V	Pull-down	16
17	PMCREQ2#	+5V	18	17	Not Used	GND	18
19	+5V (Vio)	AD31	20	19	AD30	AD29	20
21	AD28	AD27	22	21	GND	AD26	22
23	AD25	GND	24	23	AD24	+3.3V	24
25	GND	C/BE3#	26	25	IDSEL2	AD23	26
27	AD22	AD21	28	27	+3.3V	AD20	28
29	AD19	+5V	30	29	AD18	GND	30
31	+5V (Vio)	AD17	32	31	AD16	C/BE2#	32
33	FRAME#	GND	34	33	GND	Not Used	34
35	GND	IRDY#	36	35	TRDY#	+3.3V	36
37	DEVSEL#	+5V	38	37	GND	STOP#	38
39	GND	LOCK#	40	39	PERR#	GND	40
41	SDONE#	SBO#	42	41	+3.3V	SERR#	42
43	PAR	GND	44	43	C/BE1#	GND	44
45	+5V	AD15	46	45	AD14	AD13	46
47	AD12	AD11	48	47	GND	AD10	48

Table B-9. J21 and J22 PMC2 Connector Pin Assignments (Continued)

49	AD09	+5V (Vio)	50	49	AD08	+3.3V	50
51	GND	C/BE0#	52	51	AD07	Not Used	52
53	AD06	AD05	54	53	+3.3V	Not Used	54
55	AD04	GND	56	55	Not Used	GND	56
57	+5V	AD03	58	57	Not Used	Not Used	58
59	AD02	AD01	60	59	GND	Not Used	60
61	AD00	+5V (Vio)	62	61	ACK64#	+3.3V	62
63	GND	REQ64#	64	63	GND	Not Used	64

Table B-10. J23 and J24 PMC2 Connector Pin Assignments

J23				J24			
1	Reserved	GND	2	1	PMC2_1 (P2-D1)	PMC2_2 (P2-Z1)	2
3	GND	C/BE7#	4	3	PMC2_3 (P2-D2)	PMC2_4 (P2-D3)	4
5	C/BE6#	C/BE5#	6	5	PMC2_5 (P2-Z3)	PMC2_6 (P2-D4)	6
7	C/BE4#	GND	8	7	PMC2_7 (P2-D5)	PMC2_8 (P2-Z5)	8
9	+5V (Vio)	PAR64	10	9	PMC2_9 (P2-D6)	PMC2_10 (P2-D7)	10
11	AD63	AD62	12	11	PMC2_11 (P2-Z7)	PMC2_12 (P2-D8)	12
13	AD61	GND	14	13	PMC2_13 (P2-D9)	PMC2_14 (P2-Z9)	14
15	GND	AD60	16	15	PMC2_15 (P2-D10	PMC2_16 (P2-D11)	16
17	AD59	AD58	18	17	PMC2_17 (P2-Z11)	PMC2_18 (P2-D12)	18
19	AD57	GND	20	19	PMC2_19 (P2-D13)	PMC2_20 (P2-Z13)	20
21	+5V (Vio)	AD56	22	21	PMC2_21 (P2-D14)	PMC2_22 (P2-D15)	22
23	AD55	AD54	24	23	PMC2_23 (P2-Z15)	PMC2_24 (P2-D16)	24
25	AD53	GND	26	25	PMC2_25 (P2-D17)	PMC2_26 (P2-Z17)	26
27	GND	AD52	28	27	PMC2_27 (P2-D18)	PMC2_28 (P2-D19)	28
29	AD51	AD50	30	29	PMC2_29 (P2-Z19)	PMC2_30 (P2-D20)	30
31	AD49	GND	32	31	PMC2_31 (P2-D21)	PMC2_32 (P2-Z21)	32
33	GND	AD48	34	33	PMC2_33 (P2-D22	PMC2_34 (P2-D23)	34
35	AD47	AD46	36	35	PMC2_35 (P2-Z23)	PMC2_36 (P2-D24)	36
37	AD45	GND	38	37	PMC2_37 (P2-D25)	PMC2_38 (P2-Z25	38

Table B-10. J23 and J24 PMC2 Connector Pin Assignments (Continued)

39	+5V (Vio)	AD44	40	39	PMC2_39 (P2-D26)	PMC2_40 (P2-D27)	40
41	AD43	AD42	42	41	PMC2_41 (P2-Z27)	PMC2_42 (P2-D28)	42
43	AD41	GND	44	43	PMC2_43 (P2-D29)	PMC2_44 (P2-Z29)	44
45	GND	AD40	46	45	PMC2_45 (P2-D30)	PMC2_46 (P2-Z31)	46
47	AD39	AD38	48	47	Not Used	Not Used	48
49	AD37	GND	50	49	Not Used	Not Used	50
51	GND	AD36	52	51	Not Used	Not Used	52
53	AD35	AD34	54	53	Not Used	Not Used	54
55	AD33	GND	56	55	Not Used	Not Used	56
57	+5V (Vio)	AD32	58	57	Not Used	Not Used	58
59	Reserved	Reserved	60	59	Not Used	Not Used	60
61	Reserved	GND	62	61	Not Used	Not Used	62
63	GND	Reserved	64	63	Not Used	Not Used	64

Solving Startup Problems

In the event problems arise with the operation of your module, perform the troubleshooting steps in this appendix prior to calling for help or sending the board back for repair. Some of the procedures will return the board to the factory debugger environment. The board was tested under these conditions before it left the factory. The self-tests may not run in all user-customized environments.

Table C-1. Troubleshooting MVME240x Modules

Condition	Possible Problem	Try This:
I. Nothing works, no display on the terminal.	A. If the CPU LED is not lit, the board may not be getting correct power.	1. Make sure the system is plugged in. 2. Check that the board is securely installed in its backplane or chassis. 3. Check that all necessary cables are connected to the board, per this manual. 4. Check for compliance with Installation Considerations, per this manual. 5. Review the Installation and Startup procedures, per this manual. They include a step-by-step power-up routine. Try it.
	B. If the LEDs are lit, the board may be in the wrong slot.	1. The VME processor module should be in the first (left-most) slot. 2. Also check that the “system controller” function on the board is enabled, per this manual.
	C. The “system console” terminal may be configured incorrectly.	Configure the system console terminal per this manual.

Table C-1. Troubleshooting MVME240x Modules (Continued)

Condition	Possible Problem	Try This:
II. There is a display on the terminal, but input from the keyboard and/or mouse has no effect.	A. The keyboard or mouse may be connected incorrectly.	Recheck the keyboard and/or mouse connections and power.
	B. Board jumpers may be configured incorrectly.	Check the board jumpers per this manual.
	C. You may have invoked flow control by pressing a HOLD or PAUSE key, or by typing: **<CTRL>-S**	Press the HOLD or PAUSE key again. If this does not free up the keyboard, type in: **<CTRL>-Q**
III. Debug prompt `PPC1-Bug>` does not appear at power-up, and the board does not autoboot.	A. Debugger Flash may be missing	1. Disconnect *all* power from your system. 2. Check that the proper debugger devices are installed. 3. Reconnect power. 4. Restart the system by "double-button reset": press the RST and ABT switches at the same time; release RST first, wait seven seconds, then release ABT. 5. If the debug prompt appears, go to step IV or step V, as indicated. If the debug prompt does not appear, go to step VI.
	B. The board may need to be reset.	

Table C-1. Troubleshooting MVME240x Modules (Continued)

Condition	Possible Problem	Try This:
IV. Debug prompt `PPC1-Bug>` appears at power-up, but the board does not autoboot.	A. The initial debugger environment parameters may be set incorrectly.	1. Start the onboard calendar clock and timer. Type: **set *mmddyyhhmm* <CR>** where the characters indicate the month, day, year, hour, and minute. The date and time will be displayed.
	B. There may be some fault in the board hardware.	**Caution** Performing the next step (**env;d**) will change some parameters that may affect your system's operation. (continues>)

Table C-1. Troubleshooting MVME240x Modules (Continued)

Condition	Possible Problem	Try This:
IV. *Continued*		2. At the command line prompt, type in: **env;d <CR>** This sets up the default parameters for the debugger environment. 3. When prompted to Update Non-Volatile RAM, type in: **y <CR>** 4. When prompted to Reset Local System, type in: **y <CR>** 5. After clock speed is displayed, immediately (within five seconds) press the Return key: **<CR>** or **BREAK** to exit to the System Menu. Then enter a 3 for "Go to System Debugger" and Return: **3 <CR>** Now the prompt should be: `PPC1-Diag>` 6. You may need to use the **cnfg** command (see your board Debugger Manual) to change clock speed and/or Ethernet Address, and then later return to: **env <CR>** and step 3. 7. Run the self-tests by typing in: **st <CR>** The tests take as much as 10 minutes, depending on RAM size. They are complete when the prompt returns. (The onboard self-test is a valuable tool in isolating defects.) 8. The system may indicate that it has passed all the self-tests. Or, it may indicate a test that failed. If neither happens, enter: **de <CR>** Any errors should now be displayed. If there are any errors, go to step VI. If there are no errors, go to step V.

Table C-1. Troubleshooting MVME240x Modules (Continued)

Condition	Possible Problem	Try This:
V. The debugger is in system mode and the board auto-boots, or the board has passed self-tests.	A. No apparent problems — troubleshooting is done.	No further troubleshooting steps are required.
VI. The board has failed one or more of the tests listed above, and cannot be corrected using the steps given.	A. There may be some fault in the board hardware or the on-board debugging and diagnostic firmware.	1. Document the problem and return the board for service. 2. Phone 1-800-222-5640.
TROUBLESHOOTING PROCEDURE COMPLETE.		

Related Documentation D

Motorola Computer Group Documents

The Motorola publications listed below are referenced in this manual. You can obtain paper or electronic copies of Motorola Computer Group publications by:

- Contacting your local Motorola sales office
- Visiting MCG's World Wide Web literature site http://www.motorola.com/computer/literature

Table D-1. Motorola Computer Group Documents

Document Title	Motorola Publication Number
MVME2400 Series VME Processor Module Installation and Use	V2400A/IH
MVME2400 Series VME Processor Module Programmer's Reference Guide	V2400A/PG
PPCBug Firmware Package User's Manual (Parts 1 and 2)	PPCBUGA1/UM
	PPCBUGA2/UM
PPCBug Diagnostics Manual	PPCDIAA/UM
PMCspan PMC Adapter Carrier Module Installation and Use	PMCSPANA/IH

Manufacturers' Documents

For additional information, refer to the following table for manufacturers' data sheets and user's manuals. For your convenience, a source for the listed document is also provided.

It is important to note that in many cases, the information shown is preliminary and the revision levels of the documents are subject to change without notice.

Table D-2. Manufacturers' Documents

Document Title and Source	Publication Number
PowerPC 750 RISC Microprocessor Technical Summary Literature Distribution Center for Motorola Telephone: 1-800- 441-2447 FAX: (602) 994-6430 or (303) 675-2150 Web Site: http://e-www.motorola.com/webapp/DesignCenter/ E-mail: ldcformotorola@hibbertco.com	MPC750/D
PowerPC 750 RISC Microprocessor User's Manual Literature Distribution Center for Motorola Telephone: 1-800- 441-2447 FAX: (602) 994-6430 or (303) 675-2150 Web Site: http://e-www.motorola.com/webapp/DesignCenter/ E-mail: ldcformotorola@hibbertco.com OR IBM Microelectronics PowerPC603/EM603e User Manual PowerPC604e User Manual Web Site: http://www.chips.ibm.com/techlib/products/powerpc/manuals	MPC750UM/AD MPR604UMU-01
PC16550 UART National Semiconductor Corporation Web Site: http://www.national.com/	PC16550DV

Table D-2. Manufacturers' Documents (Continued)

Document Title and Source	Publication Number
21143 PCI/CardBus 10/100Mb/s Ethernet LAN Controller Hardware Reference Manual Web Site: http://developer.intel.com/design/network/manuals/278074.htm	27807401.pdf
W83C553 Enhanced System I/O Controller with PCI Arbiter (PIB) Winbond Electronics Corporation; Web Site: http://www.winbond.com.tw/product/	W83C553F
M48T59 CMOS 8K x 8 TIMEKEEPER™ SRAM Data Sheet STMicroelectronics; Web Site: http://eu.st.com/stonline/index.shtml	M48T59
Universe User Manual Tundra Semiconductor Corporation Web Site: http://www.tundra.com/page.cfm?tree_id=100008#Universe II (CA91C042)	8091042_MD300_05.pdf

Related Specifications

For additional information, refer to the following table for related specifications. For your convenience, a source for the listed document is also provided.

It is important to note that in many cases, the information is preliminary and the revision levels of the documents are subject to change without notice.

Table D-3. Related Specifications

Document Title and Source	Publication Number
VME64 Specification VITA (VMEbus International Trade Association) Web Site: http://www.vita.com/	ANSI/VITA 1-1994
IEEE - Common Mezzanine Card Specification (CMC) Institute of Electrical and Electronics Engineers, Inc. Web Site: http://standards.ieee.org/catalog/	P1386 Draft 2.0
IEEE - PCI Mezzanine Card Specification (PMC) Institute of Electrical and Electronics Engineers, Inc. Web Site: http://standards.ieee.org/catalog/	P1386.1 Draft 2.0
Peripheral Component Interconnect (PCI) Local Bus Specification, Revision 2.0, 2.1, 2.2 PCI Special Interest Group; Web Site: http://www.pcisig.com/	PCI Local Bus Specification
PowerPC Reference Platform (PRP) Specification, Third Edition, Version 1.0, Volumes I and II; International Business Machines Corporation Web Site: http://www.ibm.com	MPR-PPC-RPU-02

Table D-3. Related Specifications (Continued)

Document Title and Source	Publication Number
PowerPC Microprocessor Common Hardware Reference Platform: A System Architecture (CHRP), Version 1.0 Literature Distribution Center for Motorola Telephone: 1-800- 441-2447 FAX: (602) 994-6430 or (303) 675-2150 Web Site: http://e-www.motorola.com/webapp/DesignCenter/ E-mail: ldcformotorola@hibbertco.com OR Morgan Kaufmann Publishers, Inc. Telephone: (415) 392-2665 Telephone: 1-800-745-7323 Web Site: http://www.mkp.com/books_catalog/	
Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange; Electronic Industries Alliance; Web Site: http://global.ihs.com/index.cfm (for publications) Web Site: http://www.eia.org/	TIA/EIA-232 Standard

Glossary

10Base-5	An Ethernet implementation in which the physical medium is a doubly shielded, 50-ohm coaxial cable capable of carrying data at 10 Mbps for a length of 500 meters (also referred to as thicknet). Also known as thick Ethernet.
10Base-2	An Ethernet implementation in which the physical medium is a single-shielded, 50-ohm RG58A/U coaxial cable capable of carrying data at 10 Mbps for a length of 185 meters (also referred to as AUI or thinnet). Also known as thin Ethernet.
10BaseT	An Ethernet implementation in which the physical medium is an unshielded twisted pair (UTP) of wires capable of carrying data at 10 Mbps for a maximum distance of 185 meters. Also known as twisted-pair Ethernet.
100BaseTX	An Ethernet implementation in which the physical medium is an unshielded twisted pair (UTP) of wires capable of carrying data at 100 Mbps for a maximum distance of 100 meters. Also known as fast Ethernet.
ACIA	**A**synchronous **C**ommunications **I**nterface **A**dapter
AIX	**A**dvanced **I**nteractive e**X**ecutive (IBM version of UNIX)
architecture	The main overall design in which each individual hardware component of the computer system is interrelated. The most common uses of this term are 8-bit, 16-bit, or 32-bit architectural design systems.
ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange. This is a 7-bit code used to encode alphanumeric information. In the IBM-compatible world, this is expanded to 8-bits to encode a total of 256 alphanumeric and control characters.
ASIC	**A**pplication-**S**pecific **I**ntegrated **C**ircuit
AUI	**A**ttachment **U**nit **I**nterface
BBRAM	Battery Backed-up Random Access Memory
bi-endian	Having big-endian and little-endian byte ordering capability.

big-endian A byte-ordering method in memory where the address *n* of a word corresponds to the most significant byte. In an addressed memory word, the bytes are ordered (left to right) 0, 1, 2, 3, with 0 being the most significant byte.

BIOS **B**asic **I**nput/**O**utput **S**ystem. This is the built-in program that controls the basic functions of communications between the processor and the I/O (peripherals) devices. Also referred to as ROM BIOS.

BitBLT **Bit** Boundary **BL**ock **T**ransfer. A type of graphics drawing routine that moves a rectangle of data from one area of display memory to another. The data specifically need not have any particular alignment.

BLT **BL**ock **T**ransfer

board The term more commonly used to refer to a PCB (printed circuit board). Basically, a flat board made of nonconducting material, such as plastic or fiberglass, on which chips and other electronic components are mounted. Also referred to as a circuit board or card.

bpi **b**its **p**er **i**nch

bps **b**its **p**er **s**econd

bus The pathway used to communicate between the CPU, memory, and various input/output devices, including floppy and hard disk drives. Available in various widths (8-, 16-, and 32-bit), with accompanying increases in speed.

cache A high-speed memory that resides logically between a central processing unit (CPU) and the main memory. This temporary memory holds the data and/or instructions that the CPU is most likely to use over and over again and avoids accessing the slower hard or floppy disk drive.

CAS **C**olumn **A**ddress **S**trobe. The clock signal used in dynamic RAMs to control the input of column addresses.

CD **C**ompact **D**isc. A hard, round, flat portable storage unit that stores information digitally.

CD-ROM **C**ompact **D**isk **R**ead-**O**nly **M**emory

CFM **C**ubic **F**eet per **M**inute

CHRP See Common Hardware Reference Platform (CHRP).

CHRP-compliant See Common Hardware Reference Platform (CHRP).

CHRP Spec See Common Hardware Reference Platform (CHRP).

CISC **C**omplex-**I**nstruction-**S**et **C**omputer. A computer whose processor is designed to sequentially run variable-length instructions, many of which require several clock cycles, that perform complex tasks and thereby simplify programming.

CODEC **CO**der/**DEC**oder

Color Difference (CD) The signals of (R-Y) and (B-Y) without the luminance (-Y) signal. The Green signals (G-Y) can be extracted by these two signals.

Common Hardware Reference Platform (CHRP)
A specification published by Apple, IBM, and Motorola which defines the devices, interfaces, and data formats that make up a CHRP-compliant system using a PowerPC processor.

Composite Video Signal (CVS/CVBS)
Signal that carries video picture information for color, brightness and synchronizing signals for both horizontal and vertical scans. Sometimes referred to as “Baseband Video”.

cpi **c**haracters **p**er **i**nch

CPU **C**entral **P**rocessing **U**nit. The master computer unit in a system.

DCE **D**ata **C**ircuit-terminating **E**quipment.

DIMM **D**ual **I**nline **M**emory **M**odule. A 64 bit memory module designed to fit into a standard DIMM slot.

DLL **D**ynamic **L**ink **L**ibrary. A set of functions that are linked to the referencing program at the time it is loaded into memory.

DMA **D**irect **M**emory **A**ccess. A method by which a device may read or write to memory directly without processor intervention. DMA is typically used by block I/O devices.

DOS **D**isk **O**perating **S**ystem

dpi **d**ots **p**er **i**nch

DRAM **D**ynamic **R**andom **A**ccess **M**emory. A memory technology that is characterized by extreme high density, low power, and low cost. It must be more or less continuously refreshed to avoid loss of data.

DTE **D**ata **T**erminal **E**quipment.

ECC — **E**rror **C**orrection **C**ode

ECP — **E**xtended **C**apability **P**ort

EEPROM — **E**lectrically **E**rasable **P**rogrammable **R**ead-**O**nly **M**emory. A memory storage device that can be written repeatedly with no special erasure fixture. EEPROMs do not lose their contents when they are powered down.

EIDE — **E**nhanced **I**ntegrated **D**rive **E**lectronics. An improved version of **IDE**, with faster data rates, 32-bit transactions, and DMA. Also known as **Fast ATA-2**.

EISA (bus) — **E**xtended **I**ndustry **S**tandard **A**rchitecture (bus) (IBM). An architectural system using a 32-bit bus that allows data to be transferred between peripherals in 32-bit chunks instead of 16-bit or 8-bit that most systems use. With the transfer of larger bits of information, the machine is able to perform much faster than the standard ISA bus system.

EPP — **E**nhanced **P**arallel **P**ort

EPROM — **E**rasable **P**rogrammable **R**ead-**O**nly **M**emory. A memory storage device that can be written once (per erasure cycle) and read many times.

ESCC — **E**nhanced **S**erial **C**ommunication **C**ontroller

ESD — **E**lectro-**S**tatic **D**ischarge/Damage

Ethernet — A local area network standard that uses radio frequency signals carried by coaxial cables.

Falcon — The code name for the DRAM controller chip developed by Motorola for the MVME2600 and MVME3600 series of boards. It is intended to be used in sets of two to provide the necessary interface between the Power PC60*x* bus and the 144-bit ECC DRAM (system memory array) and/or ROM/Flash.

fast Ethernet — See 100BaseTX.

FDC — **F**loppy **D**isk **C**ontroller

FDDI — **F**iber **D**istributed **D**ata **I**nterface. A network based on the use of optical-fiber cable to transmit data in non-return-to-zero, invert-on-1s (NRZI) format at speeds up to 100 Mbps.

FIFO	**F**irst-**I**n, **F**irst-**O**ut. A memory that can temporarily hold data so that the sending device can send data faster than the receiving device can accept it. The sending and receiving devices typically operate asynchronously.
firmware	The program or specific software instructions that have been more or less permanently burned into an electronic component, such as a ROM (read-only memory) or an EPROM (erasable programmable read-only memory).
frame	One complete television picture frame consists of 525 horizontal lines with the NTSC system. One frame consists of two Fields.
graphics controller	On EGA and VGA, a section of circuitry that can provide hardware assist for graphics drawing algorithms by performing logical functions on data written to display memory.
HAL	**H**ardware **A**bstraction **L**ayer. The lower level hardware interface module of the Windows NT operating system. It contains platform specific functionality.
hardware	A computing system is normally spoken of as having two major components: hardware and software. Hardware is the term used to describe any of the physical embodiments of a computer system, with emphasis on the electronic circuits (the computer) and electromechanical devices (peripherals) that make up the system.
HCT	**H**ardware **C**onformance **T**est. A test used to ensure that both hardware and software conform to the Windows NT interface.
HAWK	The next generation ASIC, combining the functionality of the Falcon and Raven chips onto one chip.
I/O	**I**nput/**O**utput
IBC	PCI/**I**SA **B**ridge **C**ontroller
IDC	**I**nsulation **D**isplacement **C**onnector
IDE	**I**ntegrated **D**rive **E**lectronics. A disk drive interface standard. Also known as **ATA** (**A**dvanced **T**echnology **A**ttachment).
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
interlaced	A graphics system in which the even scanlines are refreshed in one vertical cycle (field), and the odd scanlines are refreshed in another vertical cycle. The advantage is that the video bandwidth is roughly half that required for a non-interlaced system of the same resolution.

	This results in less costly hardware. It also may make it possible to display a resolution that would otherwise be impossible on given hardware. The disadvantage of an interlaced system is flicker, especially when displaying objects that are only a few scanlines high.
IQ Signals	Similar to the color difference signals (R-Y), (B-Y) but using different vector axis for encoding or decoding. Used by some USA TV and IC manufacturers for color decoding.
ISA (bus)	**I**ndustry **S**tandard **A**rchitecture (bus). The de facto standard system bus for IBM-compatible computers until the introduction of VESA and PCI. Used in the reference platform specification. (IBM)
ISASIO	**ISA S**uper **I**nput/**O**utput device
ISDN	**I**ntegrated **S**ervices **D**igital **N**etwork. A standard for digitally transmitting video, audio, and electronic data over public phone networks.
LAN	**L**ocal **A**rea **N**etwork
LED	**L**ight-**E**mitting **D**iode
LFM	**L**inear **F**eet per **M**inute
little-endian	A byte-ordering method in memory where the address *n* of a word corresponds to the least significant byte. In an addressed memory word, the bytes are ordered (left to right) 3, 2, 1, 0, with 3 being the most significant byte.
MBLT	**M**ultiplexed **BL**ock **T**ransfer
MCA (bus)	**M**icro **C**hannel **A**rchitecture
MCG	**M**otorola **C**omputer **G**roup
MFM	**M**odified **F**requency **M**odulation
MIDI	**M**usical **I**nstrument **D**igital **I**nterface. The standard format for recording, storing, and playing digital music.
MPC	**M**ultimedia **P**ersonal **C**omputer
MPC105	The PowerPC-to-PCI bus bridge chip developed by Motorola for the Ultra 603/Ultra 604 system board. It provides the necessary interface between the MPC603/MPC604 processor and the Boot ROM (secondary cache), the DRAM (system memory array), and the PCI bus.

MPC601 Motorola's component designation for the PowerPC 601 microprocessor.

MPC603 Motorola's component designation for the PowerPC 603 microprocessor.

MPC604 Motorola's component designation for the PowerPC 604 microprocessor.

MPIC **M**ulti-**P**rocessor **I**nterrupt **C**ontroller

MPU **M**icro**P**rocessing **U**nit

MTBF **M**ean **T**ime **B**etween **F**ailures. A statistical term relating to reliability as expressed in power on hours (poh). It was originally developed for the military and can be calculated several different ways, yielding substantially different results. The specification is based on a large number of samplings in one place, running continuously, and the rate at which failure occurs. MTBF is not representative of how long a device, or any individual device is likely to last, nor is it a warranty, but rather, a gauge of the relative reliability of a family of products.

multisession The ability to record additional information, such as digitized photographs, on a CD-ROM after a prior recording session has ended.

non-interlaced A video system in which every pixel is refreshed during every vertical scan. A non-interlaced system is normally more expensive than an interlaced system of the same resolution, and is usually said to have a more pleasing appearance.

nonvolatile memory A memory in which the data content is maintained whether the power supply is connected or not.

NTSC **N**ational **T**elevision **S**tandards **C**ommittee (USA)

NVRAM **N**on-**V**olatile **R**andom **A**ccess **M**emory

OEM **O**riginal **E**quipment **M**anufacturer

OMPAC **O**ver - **M**olded **P**ad **A**rray **C**arrier

OS **O**perating **S**ystem. The software that manages the computer resources, accesses files, and dispatches programs.

OTP **O**ne-**T**ime **P**rogrammable

palette The range of colors available on the screen, not necessarily simultaneously. For VGA, this is either 16 or 256 simultaneous colors out of 262,144.

parallel port A connector that can exchange data with an I/O device eight bits at a time. This port is more commonly used for the connection of a printer to a system.

PCI (local bus) **P**eripheral **C**omponent **I**nterconnect (local bus) (Intel). A high-performance, 32-bit internal interconnect bus used for data transfer to peripheral controller components, such as those for audio, video, and graphics.

PCMCIA (bus) **P**ersonal **C**omputer **M**emory **C**ard **I**nternational **A**ssociation (bus). A standard external interconnect bus which allows peripherals adhering to the standard to be plugged in and used without further system modification.

PCR **P**CI **C**onfiguration **R**egister

PDS **P**rocessor **D**irect **S**lot

PHB **P**CI **H**ost **B**ridge

physical address A binary address that refers to the actual location of information stored in secondary storage.

PIB **P**CI-to-**I**SA **B**ridge

pixel An acronym for picture element, and is also called a pel. A pixel is the smallest addressable graphic on a display screen. In RGB systems, the color of a pixel is defined by some Red intensity, some Green intensity, and some Blue intensity.

PLL **P**hase-**L**ocked **L**oop

PMC **P**CI **M**ezzanine **C**ard

POWER **P**erformance **O**ptimized **W**ith **E**nhanced **R**ISC architecture (IBM)

PowerPC™ The trademark used to describe the **P**erformance **O**ptimized **W**ith **E**nhanced **R**ISC microprocessor architecture for **P**ersonal **C**omputers developed by the IBM Corporation. PowerPC is superscalar, which means it can handle more than one instruction per clock cycle. Instructions can be sent simultaneously to three types of independent execution units (branch units, fixed-point units, and

floating-point units), where they can execute concurrently, but finish out of order. PowerPC is used by Motorola, Inc. under license from IBM.

PowerPC 601™ The first implementation of the PowerPC family of microprocessors. This CPU incorporates a memory management unit with a 256-entry buffer and a 32KB unified (instruction and data) cache. It provides a 64-bit data bus and a separate 32-bit address bus. PowerPC 601 is used by Motorola, Inc. under license from IBM.

PowerPC 603™ The second implementation of the PowerPC family of microprocessors. This CPU incorporates a memory management unit with a 64-entry buffer and an 8KB (instruction and data) cache. It provides a selectable 32-bit or 64-bit data bus and a separate 32-bit address bus. PowerPC 603 is used by Motorola, Inc. under license from IBM.

PowerPC 604™ The third implementation of the PowerPC family of microprocessors currently under development. PowerPC 604 is used by Motorola, Inc. under license from IBM.

PowerPC Reference Platform (PRP)
A specification published by the IBM Power Personal Systems Division which defines the devices, interfaces, and data formats that make up a PRP-compliant system using a PowerPC processor.

PowerStack™ RISC PC (System Board)
A PowerPC-based computer board platform developed by the Motorola Computer Group. It supports Microsoft's Windows NT and IBM's AIX operating systems.

PRP See PowerPC Reference Platform (PRP).

PRP-compliant See PowerPC Reference Platform (PRP).

PRP Spec See PowerPC Reference Platform (PRP).

PROM **P**rogrammable **R**ead-**O**nly **M**emory

PS/2 **P**ersonal **S**ystem/**2** (IBM)

QFP **Q**uad **F**lat **P**ackage

RAM **R**andom-**A**ccess **M**emory. The temporary memory that a computer uses to hold the instructions and data currently being worked with. All data in RAM is lost when the computer is turned off.

RAS — **R**ow **A**ddress **S**trobe. A clock signal used in dynamic RAMs to control the input of the row addresses.

Raven — The PowerPC-to-PCI local bus bridge chip developed by Motorola for the MVME2600 and MVME3600 series of boards. It provides the necessary interface between the PowerPC 60x bus and the PCI bus, and acts as interrupt controller.

Reduced-Instruction-Set Computer (RISC) — A computer in which the processor's instruction set is limited to constant-length instructions that can usually be executed in a single clock cycle.

RFI — **R**adio **F**requency **I**nterference

RGB — The three separate color signals: **R**ed, **G**reen, and **B**lue. Used with color displays, an interface that uses these three color signals as opposed to an interface used with a monochrome display that requires only a single signal. Both digital and analog RGB interfaces exist.

RISC — See Reduced Instruction Set Computer (RISC).

ROM — **R**ead-**O**nly **M**emory

RTC — **R**eal-**T**ime **C**lock

SBC — **S**ingle **B**oard **C**omputer

SCSI — **S**mall **C**omputer **S**ystems **I**nterface. An industry-standard high-speed interface primarily used for secondary storage. Data transfer rates range from 5 Mbps (SCSI-1) to 160 Mbps (Ultra-160).

serial port — A connector that can exchange data with an I/O device one bit at a time. It may operate synchronously or asynchronously, and may include start bits, stop bits, and/or parity.

SIM — **S**erial **I**nterface **M**odule

SIMM — **S**ingle **I**nline **M**emory **M**odule. A 32 bit memory module designed to fit into a standard SIMM (72 pin) slot.

SIO — **S**uper **I/O** controller

SMP — **S**ymmetric **M**ulti**P**rocessing. A computer architecture in which tasks are distributed among two or more local processors.

SMT	**S**urface **M**ount **T**echnology. A method of mounting devices (such as integrated circuits, resistors, capacitors, and others) on a printed circuit board, characterized by not requiring mounting holes. Rather, the devices are soldered to pads on the printed circuit board. Surface-mount devices are typically smaller than the equivalent through-hole devices.
software	A computing system is normally spoken of as having two major components: hardware and software. Software is the term used to describe any single program or group of programs, languages, operating procedures, and documentation of a computer system. Software is the real interface between the user and the computer.
SRAM	**S**tatic **R**andom **A**ccess **M**emory
SSBLT	**S**ource **S**ynchronous **BL**ock **T**ransfer
standard(s)	A set of detailed technical guidelines used as a means of establishing uniformity in an area of hardware or software development.
SVGA	**S**uper **V**ideo **G**raphics **A**rray (IBM). An improved VGA monitor standard that provides at least 256 simultaneous colors and a screen resolution of 800 x 600 pixels.
Teletext	One way broadcast of digital information. The digital information is injected in the broadcast TV signal, VBI, or full field, The transmission medium could be satellite, microwave, cable, etc. The display medium is a regular TV receiver.
thick Ethernet	See 10base-5.
thin Ethernet	See 10base-2.
twisted-pair Ethernet	See 10BaseT.
UART	**U**niversal **A**synchronous **R**eceiver/**T**ransmitter
Universe	ASIC developed by Tundra in consultation with Motorola, that provides the complete interface between the PCI bus and the 64-bit VMEbus.
UV	**U**ltra**V**iolet
UVGA	**U**ltra **V**ideo **G**raphics **A**rray. An improved VGA monitor standard that provides at least 256 simultaneous colors and a screen resolution of 1024 x 768 pixels.

Vertical Blanking Interval (VBI)
The time it takes the beam to fly back to the top of the screen in order to retrace the opposite field (odd or even). VBI is in the order of 20 TV lines. Teletext information is transmitted over 4 of these lines (lines 14-17).

VESA (bus) — **V**ideo **E**lectronics **S**tandards **A**ssociation (or VL bus). An internal interconnect standard for transferring video information to a computer display system.

VGA — **V**ideo **G**raphics **A**rray (IBM). The third and most common monitor standard used today. It provides up to 256 simultaneous colors and a screen resolution of 640 x 480 pixels.

virtual address — A binary address issued by a CPU that indirectly refers to the location of information in primary memory, such as main memory. When data is copied from disk to main memory, the physical address is changed to the virtual address.

VL bus — See **V**ESA **L**ocal bus (VL bus).

VMEchip2 — MCG second generation VMEbus interface ASIC (Motorola)

VME2PCI — MCG ASIC that interfaces between the PCI bus and the VMEchip2 device.

volatile memory — A memory in which the data content is lost when the power supply is disconnected.

VRAM — **V**ideo (Dynamic) **R**andom **A**ccess **M**emory. Memory chips with two ports, one used for random accesses and the other capable of serial accesses. Once the serial port has been initialized (with a transfer cycle), it can operate independently of the random port. This frees the random port for CPU accesses. The result of adding the serial port is a significantly reduced amount of interference from screen refresh. VRAMs cost more per bit than DRAMs.

Windows NT™ — The trademark representing **Windows N**ew **T**echnology, a computer operating system developed by the Microsoft Corporation (currently known as Windows 2000™).

XGA — E**X**tended **G**raphics **A**rray. An improved IBM VGA monitor standard that provides at least 256 simultaneous colors and a screen resolution of 1024 x 768 pixels.

Y Signal — Luminance. This determines the brightness of each spot (pixel) on a CRT screen either color or B/W systems, but not the color.

Index

Numerics

10/100 BASET port 2-4
16/32-bit timers 3-26

A

abort (interrupt) signal 2-3
ABT switch (S1) 2-3
altitude (operating) A-2
ambient air temperature A-4
architecture
    MVME240x 1-2
assembly language 5-3
asynchronous debug port 3-23
Auto Boot Abort Delay 6-7
Auto Boot Controller 6-7
Auto Boot Default String 6-8
Auto Boot Device 6-7
Auto Boot Partition Number 6-7
Autoboot enable 6-6, 6-7

B

backplane
    connectors, P1 and P2 1-22
    jumpers 1-21
    VMEbus 1-2
battery 3-26
baud rate 1-11, 2-5
BFL
    LED 2-4
BFL LED (DS1) 2-4
BG and IACK signals 1-21
bits per character 1-11, 2-5
block diagram
    MVME240x 3-3
board
    configuration 1-6
board information block 6-2
board placement 1-20
board structure 6-2
bridge
    as Hawk function 3-6
bug
    basics 5-1
buses, standard 3-3, 4-1

C

cables, I/O ports A-4
chassis, VMEsystem 1-3
CNFG 6-2
commands
    PPCBug 5-5
commands, debugger 5-6
conductive chassis rails A-4
configuration, debug port 2-6
configure
    PPC1Bug parameters 6-3
    VMEbus interface 6-13
Configure Board Information Block (CNFG) 6-2
connector pin assignments xvii, B-1
connectors
    MVME2400 1-8
console terminal 1-3
    preparing 1-11
cooling requirements A-3
counters 3-26

CPU
    LED 2-4
CPU LED (DS2) 2-3, 2-4

D

debug console terminal 1-3
debug firmware, PPCBug 5-1
DEBUG port 1-21
debug port 3-23
debugger
    directory 5-11
    prompt 5-2
debugger commands 5-6
debugger firmware 3-16
debugger firmware (PPCBug) 6-1
DECchip 21143 LAN controller 3-19, 4-4
description of MVME240x 1-1
diagnostics
    directory 5-11
    hardware 5-11
    prompt 5-2
    test groups 5-12
dimensions of base board A-2
directories, debugger and diagnostic 5-11
DMA channels 4-8
documentation, related D-1
DRAM
    base address 1-22
DRAM latency 3-13
DRAM speed 6-10

E

EMC regulatory compliance A-4
endian issues
    function of Hawk ASIC 4-10
    function of Universe ASIC 4-11
    PCI domain 4-10
    processor/memory domain 4-10
    VMEbus domain 4-11
ENV
    Auto Boot Abort Delay 6-7
    Auto Boot Controller 6-7
    Auto Boot Default String 6-8
    Auto Boot Device 6-7
    Auto Boot Partition Number 6-7
    L2 Cache Parity Enable 6-12
    Memory Size 6-10
    Negate VMEbus SYSFAIL* Always 6-5
    Network Auto Boot Controller 6-9
    NVRAM Bootlist 6-6
    Primary SCSI Bus Negotiations 6-5
    Primary SCSI Data Bus Width 6-5
    ROM Boot Enable 6-8
    SCSI bus reset on debugger startup 6-5
    Secondary SCSI identifier 6-5
ENV command
    parameters 6-3
environmental parameters 6-1
equipment, required 1-1
Ethernet
    station address 3-19
Ethernet controller 4-4
Ethernet interface 3-19

F

features
    Universe ASIC 3-22
features, hardware 3-1
firmware
    location 3-16
firmware initialization 5-3
firmware, PPCBug 5-1
Flash bank selection (J8) 1-9
Flash latency 3-19
Flash memory 1-9, 3-16
forced air cooling A-3
front panel
    controls 2-3
front panels, using xvi, 2-1

G

general description
    MVME240x 3-3
global bus timeout 1-22

H

hardware
- diagnostics 5-11
- initialization 5-3

hardware features 3-1
Hawk
- as MPIC Interrupt Controller 3-6
- as MPU/PCI bus bridge controller ASIC 3-23, 4-4, 4-6, 4-10, 4-11
- as SDRAM block controller 3-12
- function 3-6

Hawk SMC/PHB ASIC 3-6
HE (Help) command 5-11
headers
- J8 1-9
- J9 1-9

help command 5-11
humidity, relative A-2

I

IACK and BG signals 1-21
initialization process
- as performed by firmware 5-3

installation considerations 1-22
installing
- multiple MVME240x boards 1-22
- MVME230x into chassis 1-20
- MVME240x 1-20
- MVME240x hardware 1-12
- PCI mezzanine cards 1-12
- PMCs 1-12
- PMCspan 1-14, 1-17
- primary PMCspan 1-14
- secondary PMCspan 1-17

interconnect signals xvii, B-1
interface
- Ethernet 3-19
- PCI bus 3-20
- VMEbus 3-22

interrupt architecture, MVME230x 4-7
Interrupt Controller (MPIC) 3-25
interrupt support 4-6
interval timers 3-26
ISA Bus 3-3
ISA bus 2-3, 3-24, 4-4, 4-6

J

J11 and J12 pin assignments B-14
J21 and J22 pin assignments B-17
J9, VMEbus system controller selection header 1-9
jumper headers 1-7
jumpers, backplane 1-21
jumpers, software readable 1-10

L

L2 cache 3-1
- location/use 3-5

L2 Cache Parity Enable 6-12
latency
- DRAM 3-13

LED/serial startup diagnostic codes 6-12
LEDs
- MVME240x 2-4
- MVME240x front panel 2-4

local reset (LRST) 2-3
lowercase 5-12

M

M48T59/T559 3-24
manufacturers' documents D-2
memory available A-1
memory map
- PCI local bus 4-2, 4-3

memory maps
- MVME240x 4-1
- VMEbus 4-3

memory size 6-10
Memory Size Enable 6-10
memory sizes
- SDRAM 3-12

Motorola Computer Group documents D-1
MPC750 processor 3-5

MPIC (MultiProcessor Interrupt Controller) 3-25
MPU initialization 5-3
MPU specifications A-1
MVME230x
  EMC regulatory compliance A-4
  specifications A-1
MVME240x
  cooling requirements A-3
  installing 1-20
  LEDs 2-4
  programming xvi, 4-1
  regulatory compliance A-4
  status indicators 2-4
MVME240x features 3-1
MVME240x VME Processor Module 1-2

N

Negate VMEbus SYSFAIL* Always 6-5
NETboot enable 6-9
Network Auto Boot Controller 6-9
Network Auto Boot enable 6-9
NIOT debugger command
  using 6-10
Non-Volatile RAM (NVRAM) 6-1, 6-3
NVRAM (BBRAM) configuration area 3-19
NVRAM Bootlist 6-6

O

operating parameters 6-1
operation
  parameter (Auto Boot Abort Delay) 6-7
  parameter (Auto Boot Controller) 6-7
  parameter (Auto Boot Default String) 6-8
  parameter (Auto Boot Device) 6-7
  parameter (Auto Boot Partition Number) 6-7
  parameter (L2 Cache Parity Enable) 6-12
  parameter (Memory Size) 6-10
  parameter (Negate VMEbus SYSFAIL* Always) 6-5
  parameter (Network Auto Boot Controller) 6-9
  parameter (NVRAM Bootlist) 6-6
  parameter (Primary SCSI Bus Negotiations) 6-5
  parameter (Primary SCSI Data Bus Width) 6-5
  parameter (ROM Boot Enable) 6-8
  parameter (SCSI bus reset on debugger startup) 6-5
  parameter (Secondary SCSI identifier) 6-5

P

P1 and P2 1-22
P1 and P2 connectors 1-2, B-2
parallel port 4-8
parity 1-11, 2-5
PC16550 2-5
PCI bus 3-3, 3-20, 3-23, 4-3, 4-6
PCI bus latency 3-7
PCI expansion 3-20, 3-22
  connector description/location 3-22
PCI expansion slot
  arbiter 4-4
PCI Host Bridge (PHB) 3-25
PCI Mezzanine Card (PMC) 2-7
PCI mezzanine cards
  slots A-2
PCI Mezzanine Cards (PMCs) 1-3
PCI-ISA Bridge (PIB) controller 3-23
PCI-ISA bridge controller (PIB)
  functions 3-23
Peripheral Computer Interface (PCI) A-2
PHB
  as part of PIB controller function 3-23
  function/use 3-25
PHB (PCI Host Bridge) 3-25
PIB controller 3-23, 4-4
pin assignments, connector xvii, B-1

PMC
slot 1 arbiter 4-4
slot 1 characteristics 3-21
slot 2 arbiter 4-4
slot 2 characteristics 3-21
slots 1 & 2 double-wide characteristics 3-22
PMC Carrier Board Placement on MVME240x 1-18
PMC Module Placement on MVME240x 1-14
PMC power requirements A-3
PMC slots 1-2, 2-7
PMC1
LED 2-4
use 2-7
PMC1 (PMC slot 1) 2-7
PMC1 connector pin assignments, J11 and J12 B-14
PMC1 connector pin assignments, J13 and J14 B-15
PMC1 LED (DS4) 2-4
PMC2
LED 2-4
PMC2 (PMC slot 2)
use 2-7
PMC2 connector pin assignments, J21 and J22 B-17
PMC2 connector pin assignments, J23 and J24 B-18
PMC2 LED (DS3) 2-4
PMCs
installing 1-12
preparing 1-11
PMCspan 1-11, 2-8
preparing 1-11
PMCspan Expansion Mezzanine 1-2
PMCspan-002 Installation on an MVME240x 1-16
port
asynchronous 3-23
debug 3-23
port, Ethernet 2-4
power needs 1-2, 1-22
power requirements 3-5, A-1, A-3
PPC1-Bug> 5-2, 5-11
PPC1-Diag> 5-2, 5-11
PPCBug
basics 5-1
commands 5-5
overview 5-1
prompt 5-2
PPCBug commands
uses of 5-2
PPCBug debugger firmware 3-16, 6-1
PPCBug firmware 3-16
preparing
PMCs 1-11
PMCspan 1-11
system console terminal 1-11
preparing and installing MVME2400 xvi, 1-1
primary PMCspan
installing 1-14
Primary SCSI Bus Negotiations 6-5
Primary SCSI Data Bus Width 6-5
processor bus 3-3
programming the MVME240x xvi, 4-1
prompt, debugger 5-11
prompts
PPCBug 5-2

R

Raven MPU/PCI bus bridge controller ASIC 3-5, 4-2
readable jumpers 1-10
real-time clock 3-24
Real-Time Clock/NVRAM/timer function 3-24
regulatory guidelines A-4
related documentation, ordering D-1
related specifications D-4
remote control/status connector 3-26
required equipment 1-1
reset 4-8

resetting the system 2-3, 4-8
restart mode 5-12
RF emissions A-4
ROM Boot Enable 6-8
ROM First Access Length 6-10
ROMboot enable 6-8, 6-11, 6-12
ROMFAL 6-10
ROMNAL
    ROM Next Access Length 6-11

S

SCSI bus 6-5
SCSI bus reset on debugger startup 6-5
SD command 5-11
SDRAM
    control of blocks 3-12
    latency 3-13
    memory sizes 3-12
SDRAM memory 3-12
secondary PMCspan
    installing 1-17
Secondary SCSI identifier 6-5
serial port, MVME240x 2-5
set environment to bug/operating system (ENV) 6-3
setup terminal 1-21
SGS-Thomson MK48T559 timekeeper device 4-8
shielded cables A-4
size of base board A-2
SNAPHAT battery
    as part of Real-Time Clock 3-24
software readable jumpers 1-10
software-readable header (S3) 1-10
sources of reset 4-8
speaker output 3-26
specifications
    related D-4
start-up procedures 1-4
status indicators 2-4
stop bit per character 1-11, 2-5
switch
    S3 1-10
switch S3 1-11
switches 2-3
switches, MVME240x front panel 2-3
SYSFAIL* 6-5
system console terminal 1-3
system console, connecting 1-21
system controller 1-20
system controller function 2-3
system controller selection header 1-9

T

temperature
    operating A-1
    storage A-1
terminal setup 1-21
testing the hardware 5-11
timeout, global 1-22
timers
    16/32-bit 3-26
    interval 3-26
timers, programmable 3-26
timers, via Universe chip A-1
troubleshooting procedures C-1
troubleshooting the MVME240x 5-11
Typical Single-width PMC Module Placement on MVME230x 1-14

U

Universe ASIC 3-22
Universe VMEbus interface ASIC 2-3, 4-3, 4-4, 4-9, 4-11
unpacking the hardware 1-6
uppercase 5-12
using the front panels 2-1

V

vibration (operating) A-2
VME Processor Module MVME240x 1-2
VMEbus 3-3, A-2
    address/data configurations 1-22
    backplane 1-2

connectors B-2
memory map 4-3
memory maps 4-3
system controller selection header (J9) 1-9
Universe ASIC 3-22
VMEbus interface 6-13
VMEsystem enclosure 1-3

W

Winbond PCI/ISA bus bridge controller 3-23, 4-4
Winbond W83C553
as PCI arbiter support 4-4

INDEX