MVME5100
Single Board Computer

Installation and Use

V5100A/IH1

October 2000 Edition

Printed in the United States of America.

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. Ersatz nur 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
>
> EN50082-1:1997 “Electromagnetic Compatibility—Generic Immunity Standard, Part 1. Residential, Commercial and Light Industry”

System products also fulfill EN60950 (product safety) which is essentially the requirement for the Low Voltage Directive (73/23/EEC).

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/safety performance.

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

Notice

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

Preface

The *MVME5100 Single Board Computer Installation and Use* manual provides the information you will need to install and use your MVME5100 Single Board Computer.

The MVME5100 is a high-performance VME single board computer featuring the Motorola Computer Group (MCG) PowerPlus II architecture with a choice of PowerPC® processors—either Motorola's MPC7400 with AltiVec™ technology for algorithmic intensive computations or the low-power MPC750.

As of the printing date of this manual, the MVME5100 is available in the configurations shown below. All models of the MVME5100 are available with either VME Scanbe front panel (-01x1) or IEEE 1101 compatible front panel (-01x3).

Model	Processor	Memory	Handles
MVME5100-0131	MCP750 @450 MHz	64MB SDRAM	SCANBE
MVME5100-0161		512MB SDRAM	
MVME5100-0133		64MB SDRAM	IEEE 1101
MVME5100-0163		512MB SDRAM	
MVME5101-0131	MCP7400 @400 MHz	64MB SDRAM	SCANBE
MVME5101-0161		512MB SDRAM	
MVME5101-0133		64MB SDRAM	IEEE 1101
MVME5101-0163		512MB SDRAM	

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 improve them. 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 provide the name of your company, followed by your name and position. Please be sure to include the title and part number of the manual along with a brief explanation on how you used it. Thereafter, summarize your feelings about its strengths and weaknesses and provide us with any recommendations for improvements.

Conventions Used in This Manual

The following typographical conventions are used in this document:

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>**

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**.

Terminology

A character precedes a data or address parameter to specify the numeric format, as follows (if not specified, the format is hexadecimal):

0x	Specifies a hexadecimal number
%	Specifies a binary number
&	Specifies a decimal number

An asterisk (#) following a signal name for signals that are *level significant* denotes that the signal is *true* or valid when the signal is low.

An asterisk (#) following a signal name for signals that are *edge significant* denotes that the#actions initiated by that signal occur on high to low transition.

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:

Byte	8 bits, numbered 0 through 7, with bit 0 being the least significant.
Half word	16 bits, numbered 0 through 15, with bit 0 being the least significant.
Word	32 bits, numbered 0 through 31, with bit 0 being the least significant.
Double word	64 bits, numbered 0 through 63, with bit 0 being the least significant.

Contents

CHAPTER 3 PPCBug Firmware

CHAPTER 4 Functional Description

CHAPTER 5 Pin Assignments

APPENDIX A Specifications

APPENDIX B Troubleshooting

APPENDIX C Related Documentation

List of Figures

List of Tables

Hardware Preparation and Installation

1

Introduction

This chapter provides information on hardware preparation and installation for the MVME5100 Single Board Computer.

Note Unless otherwise specified, the designation "MVME5100" refers to all models of the MVME5100-series Single Board Computers.

Getting Started

Overview and Equipment Requirements

The MVME5100 interfaces to a VMEbus system via its P1 and P2 connectors and contains two IEEE 1386.1 PCI Mezzanine Card (PMC) Slots. The PMC Slots are 64-bit and support both front and rear I/O.

Additionally, the MVME5100 is user configurable by setting on-board jumpers. Two I/O modes are possible: 761 mode and PMC mode. 761 mode is backwards compatible with the MVME761 transition card and the P2 adapter card (excluding PMC I/O routing) used on the MVME2600/2700 product. This mode is accomplished by configuring the on-board jumpers and by attaching an IPMC761 PMC Card in PMC slot 1.

PMC mode is backwards compatible with the MVME2300/MVME2400 and is accomplished by simply configuring the on-board jumpers.

The following equipment list is appropriate for use in an MVME5100 system:

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

Unpacking Instructions

Caution

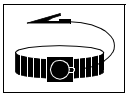
Avoid touching areas of integrated circuitry; static discharge can damage these circuits.

Note If the shipping carton(s) is/are 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(s). Refer to the packing list(s) and verify that all items are present. Save the packing material for storing and reshipping of equipment.

ESD Precautions

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 on a grounded, static-free, and adequately protected working surface. Do not slide the component over any surface. In the case of a Printed Circuit Board (PCB), place the board with the component side facing up.

If an ESD station is not available, you can avoid damage resulting from ESD by wearing an antistatic wrist strap (available locally) that is attached to an active electrical ground.

Note A system chassis may not be a suitable grounding source if it is unplugged.

Preparation

Hardware Configuration

To produce the desired board configuration and to ensure proper operation of the MVME5100, it may be necessary to perform certain modifications before and after installation. The following paragraphs discuss the preparation of the MVME5100 hardware components prior to installing them into a chassis and connecting them.

The MVME5100 provides software control over most of its options by setting bits in control registers. After installing it in a system, you can modify its configuration. For additional information on the board's control registers, refer to the *MVME5100 Single Board Computer Programmer's Reference Guide* in *Appendix C, Related Documentation*.

It is important to note that some options are not software-programmable. These specific options are controlled through manual installation or removal of jumpers or additional interface modules on the MVME5100. The following table lists the manually configured jumpers on board the MVME5100:

Table 1-1. MVME5100 On-Board Jumpers and Default Settings

Jumper	Description	Setting	Default
J4	Ethernet Port 2 Selection	For "P2" Ethernet Port 2: Pins 1,2; 3,4; 5,6; 7,8	No Jumper Installed
		For "Front Panel" Ethernet Port 2: No Jumpers Installed	
J6, J20	Operation Mode (Set Both Jumpers)	Pins 1,2 for PMC Mode	PMC Mode
		Pins 2,3 for 761 Mode	
J7	Flash Memory Selection	Pins 1,2 for Soldered Bank A	Socketed Bank B
		Pins 2,3 for Socketed Bank B	
J10, J17	Ethernet Port 2 Selection	For "Front Panel" Ethernet Port 2: Pins 1,3 and 2,4 on Both Jumpers	Front Panel Ethernet Port 2
		For "P2" Ethernet Port 2: Pins 3,5 and 4,6 on Both Jumpers	
J15	System Controller (VME)	Pins 1,2 for No SCON	Auto SCON
		Pins 2,3 for Auto SCON	
		No Jumper for ALWAYS SCON	
J16	Soldered Flash Protection	Pins 1,2 Enables Programming of Flash	Flash Prog. Enabled
		Pins 2,3 Disables Programming of Flash	

Installation Considerations

The MVME5100 draws power from the VMEbus backplane connectors P1 and P2. Connector P2 is also used for the upper 16 bits of data in 32-bit transfers, and for the upper 8 address lines in extended addressing mode. The MVME5100 may not function properly without its main board connected to VMEbus backplane connectors P1 and P2.

Whether the MVME5100 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 appropriate address ranges. D8 and/or D16 devices in the system must be handled by the processor software.

If the MVME5100 tries to access off-board resources in a nonexistent location and if the system does not have a global bus time-out, the MVME5100 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 time-out; that is when the MVME5100 is not the system controller and there is no global bus time-out elsewhere in the system.

Note Software can also disable the bus timer by setting the appropriate bits in the Universe II VMEbus interface.

Multiple MVME5100 boards may be installed in a single VME chassis; however, each must have a unique Universe II address. Other MPUs on the VMEbus can interrupt, disable, communicate with, and determine the operational status of the processor(s).

Installation

This section discusses the installation of PMCs onto the MVME5100, installation of PMCspan modules onto the MVME5100, and the installation the MVME5100 into a VME chassis.

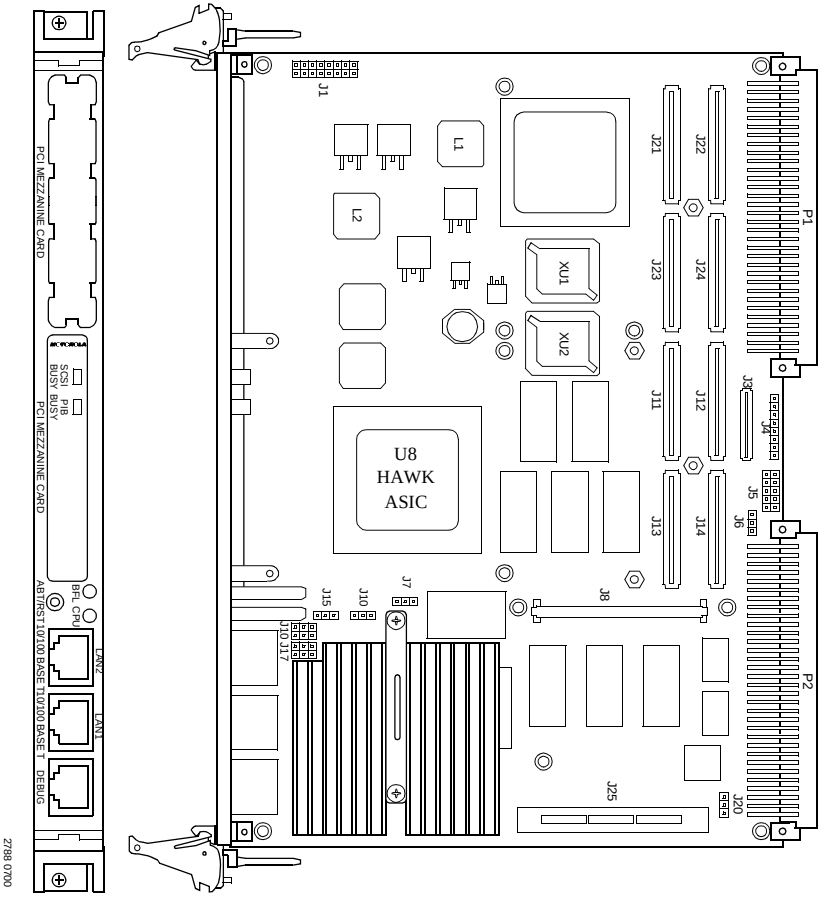


Figure 1-1. MVME5100 Layout

PMC Modules

PMC modules mount on top of the MVME5100. Perform the following steps to install a PMC module and on your MVME5100.

Warning

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

Caution

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

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

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 an electrical ground. Note that the system chassis may not be grounded if it is unplugged. 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.
3. If the MVME5100 has already been installed in a VMEbus card slot, carefully remove it as shown in Figure 1-2 and place it with connectors P1and P2 facing you.
4. Remove the filler plate(s) from the front panel of the MVME5100.
5. Align the PMC module's mating connectors to the MVME5100's mating connectors.
6. Insert the appropriate number of Phillips screws (typically 4) from the bottom of the MVME5100 into the standoffs on the PMC module and tighten the screws (refer to Figure 1-3).

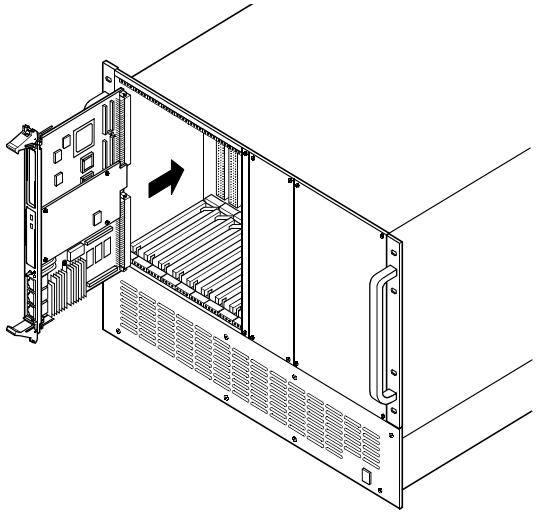


Figure 1-2. MVME5100 Installation and Removal From a VMEbus Chassis

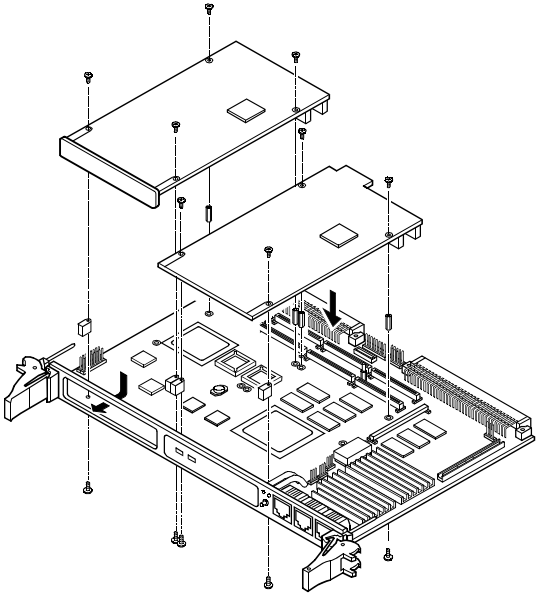


Figure 1-3. Typical PMC Module Placement on an MVME5100

Primary PMCspan

To install a PMCspan-002 PCI expansion module on your MVME5100, perform the following steps while referring to the figure on the next page:

Warning

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

Caution

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

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

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

1. Attach an ESD strap to your wrist. Attach the other end of the ESD strap to an electrical ground. Note that the system chassis may not be grounded if it is unplugged. 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.
3. If the MVME5100 has already been installed in a VMEbus card slot, carefully remove it as shown in Figure 1-2 and place it with connectors P1and P2 facing you.
4. Attach the four standoffs to the MVME5100. For each standoff:
   - Insert the threaded end into the standoff hole at each corner of the MVME5100.
   - Thread the locking nuts into the standoff tips and tighten.
5. Place the PMCspan on top of the MVME5100. Align the mounting holes in each corner to the standoffs and align PMCspan connector P4 with MVME5100 connector J25.

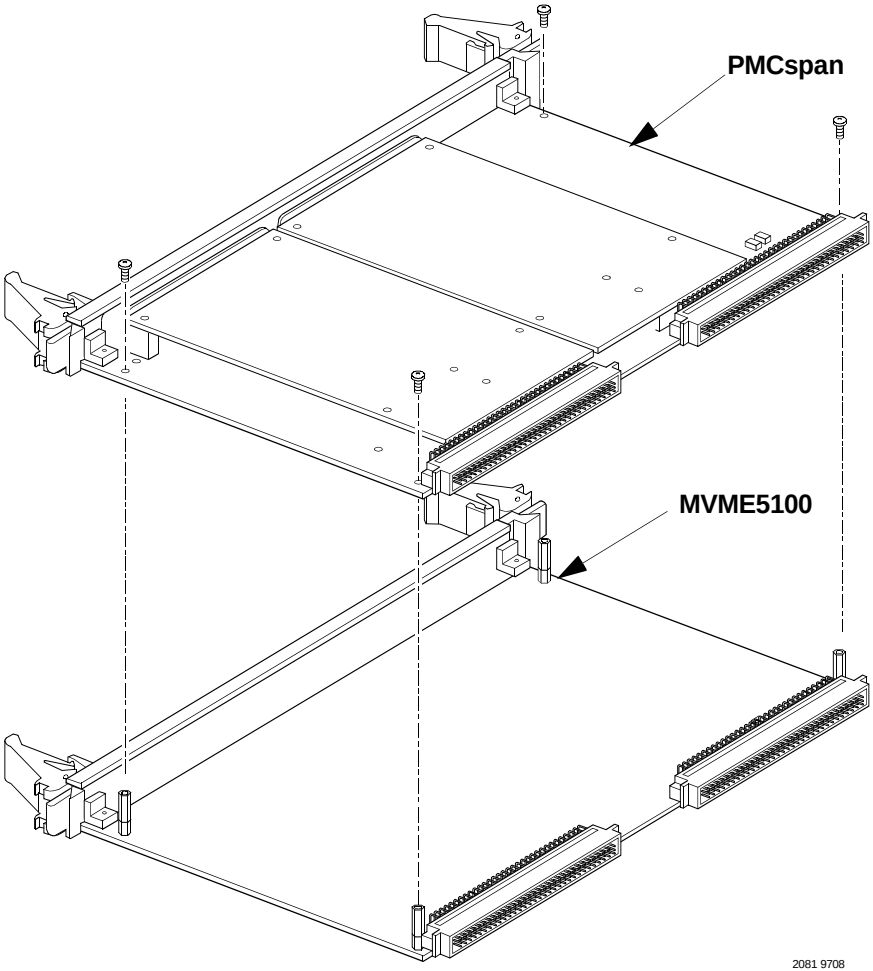


Figure 1-4. PMCspan Installation on an MVME5100

6. Gently press the PMCspan and MVME5100 together and verify that P4 is fully seated in J25.
7. Insert four short screws (Phillips type) through the holes at the corners of the PMCspan and into the standoffs on the MVME5100. Tighten screws securely.

Secondary PMCspan

The PMCspan-010 PCI expansion module mounts on top of a PMCspan-002 PCI expansion module. To install a PMCspan-010 on your MVME5100, perform the following steps while referring to the figure on the next page:

Warning

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

Caution

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

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

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 PMC modules on your PMCspan according to the instructions provided in the PMCspan and PMC manuals.

1. Attach an ESD strap to your wrist. Attach the other end of the ESD strap to an electrical ground. Note that the system chassis may not be grounded if it is unplugged. 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 module
3. If the Primary PMC Carrier Module and MVME5100 assembly is already installed in the VME chassis, carefully remove it as shown in Figure 1-2 and place it with connectors P1and P2 facing you.
4. Remove four screws (Phillips type) from the standoffs in each corner of the primary PCI expansion module.
5. Attach the four standoffs from the PMCspan-010 mounting kit to the PMCspan-002 by screwing the threaded male portion of the standoffs in the locations where the screws were removed in the previous step.

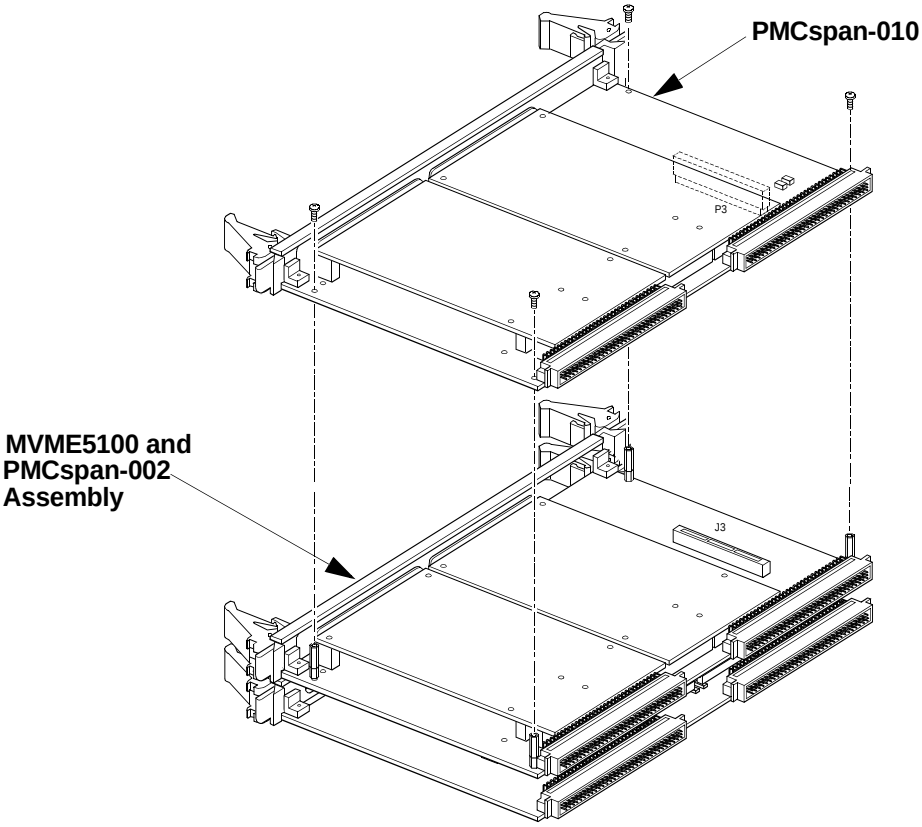


Figure 1-5. PMCspan-010 Installation on a PMCspan-002/MVME5100

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 and verify that P3 is fully seated in J3.
8. Insert the four screws (Phillips type) through the holes at the corners of PMCspan-010 and into the standoffs on the primary PMCspan-002. Tighten screws securely.

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.

MVME5100

Before installing the MVME5100 into your VME chassis, ensure that the jumpers are configured properly. This procedure assumes that you have already installed the PMCspan(s) and any PMCs that you have selected.

Perform the following steps to install the MVME5100 in your VME chassis:

Warning

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

Caution

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

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

1. Attach an ESD strap to your wrist. Attach the other end of the ESD strap to an electrical ground. Note that the system chassis may not be grounded if it is unplugged. 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 module
3. Remove the filler panel from the VMEbus chassis card slot where you are going to install the MVME5100. If you have installed one or more PMCspan PCI expansion modules onto your MVME5100, you will need to remove filler panels from one additional card slot for each PMCspan, above the card slot for the MVME5100.
   - If you intend to use the MVME5100 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 MVME5100 as system controller, it can occupy any unused card slot.

4. Slide the MVME5100 (and PMCspans if used) into the selected card slot(s). Verify that the module or module(s) seated properly in the P1 and P2 connectors on the chassis backplane. Do not damage or bend connector pins.

5. Secure the MVME5100 (and PMCspans if used) in the chassis with the screws in the top and bottom of its front panel and verify proper contact with the transverse mounting rails to minimize RF emissions.

Note Some VME backplanes (such as those used in Motorola "Modular Chassis" systems) have an auto-jumpering feature for automatic propagation of the IACK and BG signals.
The step immediately below 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 slots occupied by the MVME5100 and any PMCspan modules.

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 MVME5100.

Note In normal operation, the host CPU controls MVME5100 operation via the VMEbus Universe registers.

8. Replace the chassis or system cover(s) and cable peripherals to the panel connectors as required.

9. Reconnect the system to the AC or DC power source and turn the system power on.

10. The MVME5100's green **CPU** LED indicates activity as a set of confidence tests is run, and the debugger prompt `PPC6-Bug>` appears.

Operation 2

Introduction

This chapter provides operating instructions for the MVME5100 Single Board Computer. It includes necessary information about powering up the system along with the functionality of the switches, status indicators, and I/O ports on the front panels of the board.

Switches and Indicators

The front panel of the MVME5100 as shown in Figure 1-1, incorporates one dual function toggle switch (**ABT**/**RST**) and two Light-Emitting Diode (LED) status indicators (**BFL**, **CPU**) located on the front panel.

ABT/RST Switch

Abort Function

When toggled to **ABT**, the switch generates an interrupt signal to the processor. The interrupt is normally used to abort program execution and return control to the debugger firmware located in the processor and flash memory.

The interrupt signal reaches the processor via ISA bus interrupt line IRQ8. The interrupter connected to the ABORT switch is an edge-sensitive circuit, filtered to remove switch bounce.

Reset Function

When toggled to **RST**, the switch resets all onboard devices. To generate a reset, the switch must be depressed for more than 5 seconds.

The on-board Universe ASIC includes both a global and a local reset driver. When the ASIC operates as the System Controller, the reset driver provides a global system reset by asserting the SYSRESET# signal.

Additionaly, when the MVME5100 is configured as a System Controller (SCON), a SYSRESET# signal may be generated by toggling the **ABT/RST** switch to **RST**, or by a power-up reset, or by a watchdog timeout, or by a control bit in the Miscellaneous Control Register (MISC_CTL) in the Universe ASIC.

Note SYSRESET# remains asserted for at least 200 ms, as required by the VMEbus specification.

Status Indicators

There are two Light-Emitting Diode (LED) status indicators located on the MVME5100 front panel. They are labeled **BFL** and **CPU**.

RST Indicator (DS1)

The *yellow* **BFL** LED indicates board failure; this indicator is also illuminated during reset as an LED test.

CPU Indicator (DS2)

The *green* **CPU** LED indicates CPU activity.

Connectors

There are three connectors on the front panel of the MVME5100. Two are bottom-labeled **10/100 BASE T** and one is labeled **DEBUG**.

10/100 BASE T Ports

The two RJ45 ports labeled **10/100 BASE T** provides the 10 Base-T/100 Base-Tx Ethernet LAN interface. These connectors are top-labeled with the designation **LAN1** and **LAN2**.

DEBUG Port

The RJ45 port labeled **DEBUG** provides an RS232 serial communications interface, based on TL16C550 Universal Asynchronous Receiver/Transmitter (UART) controller chip. It is asynchronous only. For additional information on pin assignments, refer to Chapter 5, *Pin Assignments*.

The **DEBUG** port may be used for connecting a terminal to the MVME5100 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.

System Powerup

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.

Initialization Process

The MPU, hardware, and firmware initialization process is performed by the PPCBug firmware upon system powerup or system reset. The firmware initializes the devices on the MVME5100 in preparation for booting an operating system.

The firmware is shipped from the factory with an appropriate set of defaults. Depending on your system and specific application, there may or may not be a need to modify the firmware configuration before you boot the operating system. If it is necessary, refer to Chapter 3, *PPCBug Firmware* for additional information on modifying firmware default parameters.

The following flowchart in Figure 2-1 shows the basic initialization process that takes place during MVME5100 system start-ups.

For further information on PPCBug, refer to the following:

- Chapter 3, *PPCBug Firmware*
- Appendix B, *Troubleshooting*
- Appendix C, *Related Documentation*

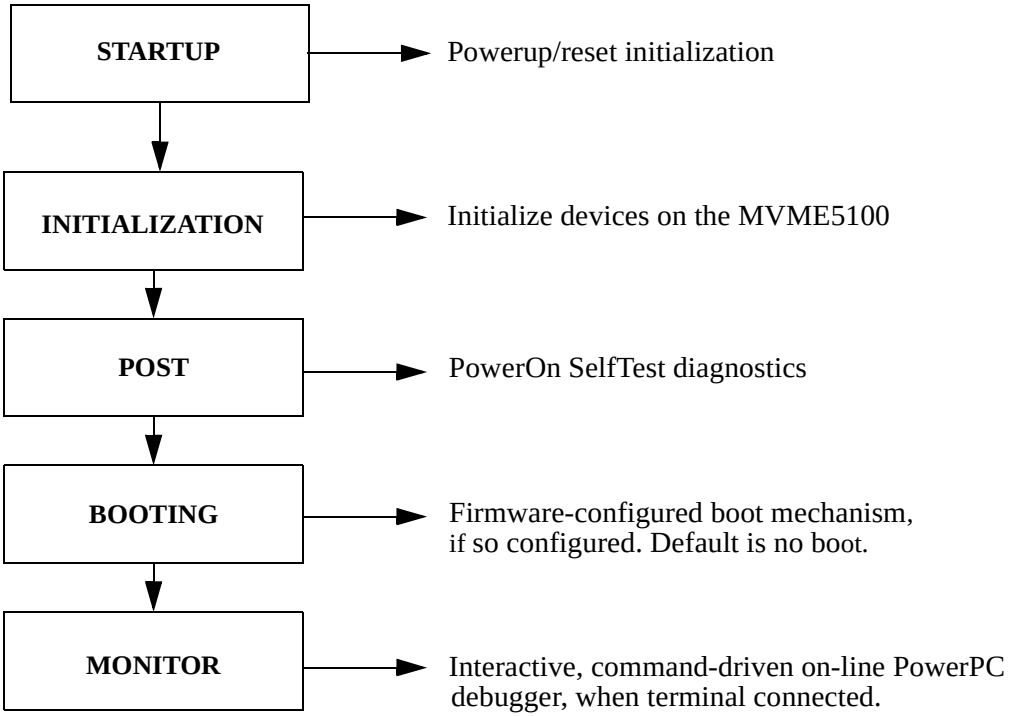


Figure 2-1. Boot-Up Sequence

PPCBug Firmware 3

Introduction

The PPCBug firmware is the layer of software just above the hardware. The firmware provides the proper initialization for the devices on the MVME5100 upon powerup or reset.

This chapter describes the basics of the PPCBug and its architecture. It also describes the monitor (interactive command portion of the firmware), and provides information on using the PPCBug debugger and the special commands. A complete list of PPCBug commands is also provided.

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

PPCBug Overview

The PPCBug (also known as PowerPC debug firmware) 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. The PPCBug provides a high degree of functionality, user friendliness, portability, and ease of maintenance.

The PPCBug also achieves its portability 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 powerup feature which verifies the integrity of the system

PPCBug consists of three parts:

- A command-driven, user-interactive software debugger, described in the *PPCBug Firmware Package User's Manual* listed in Appendix C, *Related Documentation* (hereafter referred to as "debugger" or "PPCBug").
- A command-driven diagnostics package for the MVME5100 hardware (hereafter referred to as "diagnostics"). The diagnostics package is described in the *PPCBug Diagnostics Manual* listed in Appendix C, *Related Documentation*.
- 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 PPC6-Bug> is displayed and you have all of the debugger commands at your disposal.
- If you are in the diagnostic directory, the diagnostic prompt PPC6-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 (e.g., **GO**), then control may or may not return to PPCBug, depending on the outcome of the user program.

Implementation and Memory Requirements

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.

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. The result (which includes a precalculated checksum contained in the flash devices), is verified against the expected checksum.

PPCBug requires a maximum of 768KB of read/write memory. The debugger allocates this space from the top of memory. For example, a system containing 64MB (0x04000000) of read/write memory will place the PPCBug memory locations 0x03F40000 to 0x3FFFFFF.

Using PPCBug

PPCBug is command-driven; it performs its various operations in response to commands that you enter at the keyboard. When the PPC6-Bug prompt appears on the screen, the debugger is ready to accept debugger commands. When the `PPC6-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 enter 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* listed in Appendix C, *Related Documentation*.

After the debugger executes the command, the prompt reappears. However, depending on what the user program does, if the command causes execution of a user target code (i.e. **GO**), then control may or may not return to the debugger.

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 (e.g., **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.

Hardware and Firmware Initialization

The debugger performs the hardware and firmware initialization process. This process occurs each time the MVME5100 is reset or powered up. The steps listed below are a high-level outline; be aware that 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 (i.e., DRAM).
11. Initializes the read/write memory controller. Sets base address of memory to 0x00000000.
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 (i.e., 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 0xFFF00000 to 0x00000000.
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 (i.e., issues the PPC6-Bug> prompt).

Default Settings

The following sections provide information pertaining to the firmware settings of the MVME5100. Default (factory set) Environment (ENV) commands are provided to inform you on how the MVME5100 was configured at the time it left the factory.

CNFG - Configure Board Information Block

Use this command to display and configure the Board Information Block, which is resident within the NVRAM. This data block contains various elements detailing specific operational parameters of the MVME5100. The structure for the board is shown in the following example:

```
Board (PWA) Serial Number     = MOT00xxxxxxx
Board Identifier              = MVME5100
Artwork (PWA) Identifier      = 01-W3403FxxC
MPU Clock Speed               = 450
Bus Clock Speed               = 100
Ethernet Address              = 0800AF2A0A57
Primary SCSI Identifier       = 07
System Serial Number          = nnnnnnn
System Identifier             = Motorola MVME5100
License Identifier            = nnnnnnnn
```

The Board Information Block parameters shown above are left-justified character (ASCII) strings padded with space characters.

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 *PPCBug Firmware Package User's Manual* listed in Appendix C, *Related Documentation* 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 *MVME5100 Programmer's Reference Guide* listed in Appendix C, *Related Documentation*.

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* listed in Appendix C, *Related Documentation*.

```
Maximum Memory Usage (MB,0=AUTO) = 1?
```

This parameter specifies the maximum number of megabytes the bug is allowed to use. Allocation begins at the top of physical memory and expands downward as more memory is required until the maximum value is reached.

If a value of zero is specified, memory will continue to be increased as needed until half of the available memory is consumed (i.e. 32MB in a 64MB system). This mode is useful for determining the full memory required for a specific configuration. Once this is determined, a hard value may be given to the parameter and it is guaranteed that no memory will be used over this amount.

The default value for this parameter is one.

Note: The bug does not automatically acquire all of the memory it is allowed. It accumulates memory as necessary in one megabyte blocks.

```
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 MVME5100 is cross-loaded from another VME-based CPU in order 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.

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 (e.g., 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 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 powerup reset only.

N Give powerup 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 powerup only [Y/N] = N?
```

Y Autoboot is attempted at powerup 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 = 0x00)

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

Refer to the *PPCBug Firmware Package User's Manual* listed in Appendix C, *Related Documentation* for a listing of disk/tape devices currently supported by PPCBug.
(Default = 0x00)

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

Which disk "partition" is to be booted, as specified in the PowerPC Reference Platform (PReP) 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 = 0xFFF00000)

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

The last location tested when PPCBug searches for a ROMboot module. (Default = 0xFFFFFFFC)

```
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 powerup 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* listed in Appendix C, *Related Documentation* for a listing of network controller modules currently supported by PPCBug. (Default = 0x00)

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

Refer to the *PPCBug Firmware Package User's Manual* listed in Appendix C, *Related Documentation* for a listing of network controller modules currently supported by PPCBug. (Default = 0x00)

```
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 0x1000, but this value is application-specific.
(Default = 0x00001000)

Caution

If you use the **NIOT** debugger command, these parameters need to be saved somewhere in the offset range 0x00001000 through 0x000016F7. 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 0x00001000 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 SelfTest diagnostics. (Default)
N	Memory will not be sized for SelfTest diagnostics.

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

The default Starting Address is 0x00000000.

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

The default Ending Address is the calculated size of local memory. If the memory start is changed from 0x0x00000000, this value will also need to be adjusted.

```
DRAM Speed in NANO Seconds = 15?
```

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 Bank A Access Speed (ns) = 80?
```

This defines the minimum access speed for the Bank A Flash Device(s) in nanoseconds.

```
ROM Bank B Access Speed (ns) = 70?
```

This defines the minimum access speed for the Bank B Flash Device(s) in nanoseconds.

```
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 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 4 to yield the values for route control registers PIRQ0/1/2/3. The default is determined by system type.

LED/Serial Startup Diagnostic Codes

These codes can be displayed at key points in the initialization of the hardware devices. The codes are enabled by an **ENV** parameter.

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

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.

```
Serial Startup Code LF 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:

The list of LED/serial codes is included in the section on *MPU, Hardware, and Firmware Initialization* found in Chapter 1 of the *PPCBug Firmware Package User's Manual* listed in Appendix C, *Related Documentation*.

Configuring the VMEbus Interface

ENV asks the following series of questions to set up the VMEbus interface for the MVME5100. To perform this configuration, you should have a working knowledge of the Universe ASIC as described in your *MVME5100 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 = 81000000?
```

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

```
PCI Slave Image 1 Bound Address Register = A0000000?
```

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

```
PCI Slave Image 1 Translation Offset = 80000000?
```

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 = A0000000?
```

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

```
PCI Slave Image 2 Bound Address Register = A2000000?
```

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

```
PCI Slave Image 2 Translation Offset = 500000000?
```

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 = AFFF0000?
```

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

```
PCI Slave Image 3 Bound Address Register = B0000000?
```

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

```
PCI Slave Image 3 Translation Offset = 50000000?
```

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 = 00000000?
```

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.

Firmware Command Buffer

```
Firmware Command Buffer Enable = N?
```

Y	Enables Firmware Command Buffer execution.
N	Disables Firmware Command Buffer execution (Default).

```
Firmware Command Buffer Delay = 5?
```

Defines the number of seconds to wait before firmware begins executing the startup commands in the startup command buffer. During this delay, you may press any key to prevent the execution of the startup command buffer.

The default value of this parameter causes a startup delay of 5 seconds.

```
Firmware Command Buffer:
['NULL' terminates entry]?
```

The Firmware Command Buffer contents contain the BUG commands which are executed upon firmware startup.

BUG commands you place into the command buffer should be typed just as you enter the commands from the command line.

The string '**NULL**' on a new line terminates the command line entries.

All BUG commands except for the following may be used within the command buffer: **DU, ECHO, LO, TA, VE**.

Note Interactive editing of the startup command buffer is not supported. If changes are needed to an existing set of startup commands, a new set of commands with changes must be reentered.

Standard 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* listed in Appendix C, *Related Documentation*.

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 3-1. Debugger Commands

Command	Description
AS	Assembler
BC	Block of Memory Compare
BF	Block of Memory Fill
BI	Block of Memory Initialize
BM	Block of Memory Move
BS	Block of Memory Search
BR	Breakpoint Insert
BV	Block of Memory Verify
CACHE	Modify Cache State
CM	Concurrent Mode
CNFG	Configure Board Information Block
CS	Checksum a Block of data
CSAR	PCI Configuration Space READ Access
CSAW	PCI Configuration Space WRITE Access
DC	Data Conversion and Expression Evaluation
DE	Detect Errors
DS	Disassembler
DU	Dump S-Records
ECHO	Echo String

Table 3-1. Debugger Commands (Continued)

Command	Description
ENV	Set Environment to Bug/Operating System
FORK	Fork Idle MPU at Address
FORKWR	Fork Idle MPU with Registers
G	“Alias” for “GO” Command
GD	Go Direct (Ignore Breakpoints)
GEVBOOT	Global Environment Variable Boot - Bootstrap Operating System
GEVDEL	Global Environment Variable Delete
GEVDUMP	Global Environment Variable(s) Dump (NVRAM Header + Data)
GEVEDIT	Global Environment Variable Edit
GEVINIT	Global Environment Variable Initialize (NVRAM Header)
GEVSHOW	Global Environment Variable Show
GN	Go to Next Instruction
GO	Go Execute User Program
GT	Go to Temporary Breakpoint
HE	Help on Command(s)
IBM	Indirect Block Move
IDLE	Idle Master MPU
IOC	I/O Control for Disk
IOI	I/O Inquiry
IOP	I/O Physical to Disk
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

Table 3-1. Debugger Commands (Continued)

Command	Description
M	"Alias" for "MM" Command
MA	Macro Define/Display
MAE	Macro Edit
MAL	Enable Macro Expansion Listing
MAR	Macro Load
MAW	Macro Save
MD	Memory Display
MDS	Memory Display (Sector)
MENU	System Menu
MM	Memory Modify
MMD	Memory Map Diagnostic
MMGR	Access Memory Manager
MS	Memory Set
MW	Memory Write
NAB	Automatic Network Bootstrap Operating System
NAP	Nap MPU
NBH	Network Bootstrap Operating System and Halt
NBO	Network Bootstrap Operating System
NIOC	Network I/O Control
NIOP	Network I/O Physical
NIOT	I/O "Teach" for Configuring Network Controller
NOBR	Breakpoint Delete
NOCM	No Concurrent Mode
NOMA	Macro Delete
NOMAL	Disable Macro Expansion Listing
NOPA	Printer Detach

Table 3-1. Debugger Commands (Continued)

Command	Description
NOPF	Port Detach
NORB	No ROM Boot
NOSYM	Detach Symbol Table
NPING	Network Ping
OF	Offset Registers Display/Modify
PA	Printer Attach
PBOOT	Bootstrap Operating System
PF	Port Format
PFLASH	Program FLASH Memory
PS	Put RTC into Power Save Mode
RB	ROMboot Enable
RD	Register Display
REMOTE	Remote
RESET	Cold/Warm Reset
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
ST	Self Test
SYM	Symbol Table Attach
SYMS	Symbol Table Display/Search
T	Trace
TA	Terminal Attach
TIME	Display Time and Date

Table 3-1. Debugger Commands (Continued)

Command	Description
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 (PFLASH) 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.

Diagnostics

The PPCBug hardware diagnostics are intended for testing and troubleshooting the MVME5100.

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 `PPC6-Bug>` is displayed, and all of the debugger commands are available. Diagnostics commands cannot be entered at the PPC6-Bug> prompt.

If you are in the diagnostic directory, the diagnostic prompt `PPC6-Diag>` is displayed, and all of the debugger and diagnostic commands are available.

PPCBug's diagnostic test groups are listed in Table 3-2. Note that not all tests are performed on the MVME5100. Using the **HE** command, you can list the diagnostic routines available in each test group. Refer to the *PPCBug Diagnostics Manual* listed in Appendix C, *Related Documentation* for complete descriptions of the diagnostic routines and instructions on how to invoke them.

Table 3-2. Diagnostic Test Groups

Test Group	Description
EPIC	EPIC Timers Test
PHB	PCI Bridge Revision Test
RAM	RAM Tests (various)
HOSTDMA	DMA Transfer Test
RTC	MK48Txx Real Time Clock Tests
UART	Serial Input/Output Tests (Register, IRQ, Baud, & Loopback)
Z8536	Z8536 Counter/Timer Tests*
SCC	Serial Communications Controller (Z85C230) Tests*
PAR8730x	Parallel Interface (PC8730x) Test*
KBD8730x	PC8730x Keyboard/Mouse Tests*
ISABRDGE	PCI/ISA Bridge Tests (Register Access & IRQ)
VME3	VME3 Tests (Register Read & Register Walking Bit)
DEC	DEC21x43 Ethernet Controller Tests
CL1283	Parallel Interface (CL1283) 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 MVME5100.

Functional Description 4

Introduction

This chapter provides a functional description for the MVME5100 Single Board Computer. The MVME5100 is a high-performance product featuring Motorola's PowerPlus II architecture with a choice of PowerPC processors—either Motorola's MPC7400 with AltiVec™ technology for algorithmic intensive computations or the low-power MPC750.

The MVME5100 incorporates a highly optimized PCI interface and memory controller enabling up to 582MB memory read bandwidth and 640MB burst write bandwidth.

The optimization of the memory bus is as important as optimization of the system bus in order to achieve maximum system performance. The MVME5100's advanced PowerPlus II Architecture supports full PCI throughput of 264MB without starving the CPU of its memory.

Additional features of the MVME5100 include dual Ethernet ports, dual serial ports, and up to 17MB of Flash.

Features Summary

The table below lists the general features for the MVME5100. Refer to the MVME5100 Specification Data Sheet for additional product specifications and information.

Table 4-1. MVME5100 General Features

Feature	Specification
Microprocessors and Bus Clock Frequency	• MPC7400 @400 MHz Internal Clock Frequency • MPC750 @450 MHz Internal Clock Frequency • Bus Clock Frequency up to 100 MHz
L2 Cache (Optional)	• 1MB (MPC750) or 2MB (MPC7400) using burst-mode SRAM modules.

Table 4-1. MVME5100 General Features (Continued)

Feature	Specification
Memory	• EEPROM, on-board programmable • 1MB via two 32-pin PLCC/CLCC sockets; 16MB Surface Mount
Main Memory (SDRAM)	• PC100 ECC SDRAM with 100 MHz bus • 32MB to 512MB on board, expandable to 1GB via RAM500 memory mezzanine
NVRAM	• 32KB (4KB available for users)
Memory Controller	• Hawk System Memory Controller (SMC)
PCI Host Bridge	• Hawk PCI Host Bridge (PHB)
Interrupt Controller	• Hawk Multi-Processor Interrupt Controller (MPIC)
Peripheral Support	• Dual 16550-Compatible Asynchronous Serial Port's Routed to the Front Panel RJ45 Connnector (COM1) and On-Board Header (COM2) • Dual Ethernet Interfaces, one routed to the Front Panel RJ45, One Routed to the Front Panel RJ45 or Optionally Routed to P2, RJ45 on MVME761
VMEbus	• Tundra Universe Controller, 64-bit PCI • Programmable Interrupter & Interrupt Handler • Programmable DMA Controller With Link List Support • Full System Controller Functions
PCI/PMC/Expansion	• Two 32/64-bit PMC Slots With Front-Panel I/O Plus, P2 Rear I/O (MVME2300 Routing) • One PCI Expansion Connector (for the PMCSpan)
Miscellaneous	• Combined RESET and ABORT Switch • Status LEDs
Form Factor	• 6U VME

Features Descriptions

General

As stated earlier, the MVME5100 is a high-performance VME based Single Board Computer featuring Motorola's PowerPlus II architecture with a choice of processors. The board can be equipped with either the Motorola MPC7400 processor with AltiVec™ technology for algorithmic intensive computations or with the low-power MPC750 for low-power or field applications.

Designed to meet the needs of OEMs servicing the military and aerospace, industrial automation, and semiconductor process equipment market segments, the MVME5100 is available in both commercial grade (0° to 55° C) and industrial grade (–20° to 71° C) temperatures.

The MVME5100 has two Input/Output (I/O) modes of operation: PMC mode and MVME761 mode. In PMC mode, it is fully backwards compatible with previous generation dual PMC products such as the MVME2300 and MVME2400.

In MVME761 mode, the MVME5100 is backwards compatible with Motorola's MVME761 transition board originated for use with previous generation single-board computer products such as the MVME2600 and MVME2700.

It is important to note that MVME761 compatibility is accomplished with the addition of the IPMC761 (an optional add-on PMC card). The IPMC761 provides rear I/O support for one single-ended ultra-wide SCSI device, one parallel port, four serial ports (two synchronous and two asynchronous/synchronous), and I^2C. This multi-function PMC card is offered with the MVME5100 as a factory bundled configuration.

The following diagram illustrates the architecture of the MVME5100 Single Board Computer.

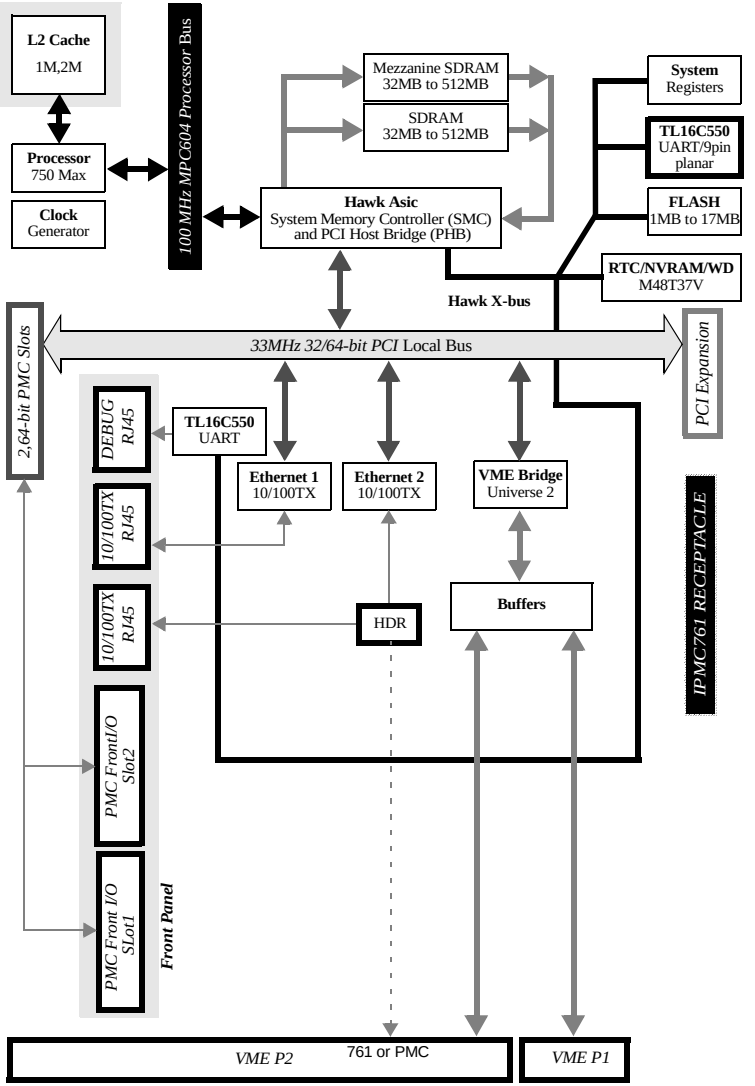


Figure 4-1. MVME5100 Block Diagram

Processor

The MVME5100 incorporates a “BGA” foot print that supports both the MCP7400 and the MCP750 processor. The maximum external processor bus speed is 100 MHz.

Note The MCP7400 can be configured to operate only in the PowerPC 60x interface.

System Memory Controller and PCI Host Bridge

The on-board “Hawk” ASIC provides the bridge function between the processor’s bus and the PCI bus. It provides 32bit addressing and 64bit data; however, 64bit addressing (dual address cycle) is not supported. The ASIC also supports various processor external bus frequencies up to 10MHz.

There are four programmable map decoders for each direction to provide flexible address mappings between the processor and the PCI bus. The ASIC also provides an Multi-Processor Interrupt Controller (MPIC) to handle various interrupt sources. They are: four MPIC timer interrupts, interrupts from all PCI devices, and two software interrupts.

Memory

Flash Memory

The MVME5100 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 4 16-bit Smart Voltage FLASH SMT devices. With 32Mbit flash devices, the flash memory size is 16MB. Note that only 32-bit writes are supported for this bank of flash memory.

ECC SDRAM Memory

The MVME5100's on-board memory and optional memory mezzanines allow for a variety of memory size options. Memory size can be 64 or 512MB for a total of 1GB on-board and mezzanine ECC memory. The memory is controlled by the hardware which provides single-bit error correction and double-bit error detection (ECC is calculated over 72-bits).

Either 1 or 2 mezzanines can be installed. Each mezzanine will add 1 bank of SDRAM memory of 256MB. A total of 512MB of mezzanine memory can be added.

P2 Input/Output (I/O) Modes

The MVME5100 has two P2 I/O modes (761 and PMC) that are user-configurable with jumpers on the board. The jumpers route the on-board Ethernet port 2 to row C of the P2 connector.

761 mode is backwards compatible with the MVME761 transition card and P2 adapter card (excluding PMC I/O routing) used on the MVME2600/2700. 761 mode is accomplished by configuring the on-board jumpers and attaching an IPMC761 PMC Card in PMC slot 1 of the MVME5100.

PMC mode is backwards compatible with the MVME2300/MVME2400. PMC mode is accomplished by simply configuring the on-board jumpers.

Note Refer to Chapter 5, *Pin Assignments* for P2 Input/Output Mode jumper settings.

Input/Output Interfaces

Ethernet Interface

The MVME5100 incorporates dual Ethernet interfaces (Port 1 and Port 2) via two Fast Ethernet PCI controller chips.

The Port 1 10 Base-T/100 Base-Tx interface is routed to the front panel. The Port 2 Ethernet interface is routed to either the front panel or the P2 connector as configured by jumpers. The front panel connectors are of the RJ45 type.

Every board will be assigned two Ethernet Station Addresses. The address is $0001AFXXXXX where XXXXX is the unique number assigned to each interface. Each Ethernet Station Address is displayed on a label attached to the PMC front-panel keep-out area.

In addition, LAN 1 Ethernet address is stored in the configuration area of the NVRAM specified by the Boot ROM and in SROM.

VMEbus Interface

The VMEbus interface is provided by the Universe ASIC. Refer to the *Universe User's Manual* for additional information.

Asynchronous Communications

The MVME5100 provides dual asynchronous debug ports. The serial signals COM1 and COM2 are routed through appropriate EIA-232 drivers and receivers to an RJ45 connector on the front panel (COM1) and an on-board connector (COM2). The external signals are ESD protected.

Real-Time Clock & NVRAM & Watchdog Timer

The MVME5100's design incorporates 32KB of non-volatile static RAM, along with a real-time clock and a watchdog function an integrated device. Refer to the MVME5100 product data sheet for the latest information on the specific device used and to that device's data sheets for programming and engineering information.

Timers

Timers and counters on the MVME5100 are provided by the board's hardware (Hawk ASIC). There are four 32-bit timers on the board that may be used for system timing or to generate periodic interrupts.

Interrupt Routing

Legacy interrupt assignment for the PCI/ISA Bridge is maintained to ensure software compatibility between the MVME5100 and the MVME2700 while in 761 mode.

This is accomplished by using the on-board IPMC761 connector to route the PCI/ISA Bridge interrupt signal to the external interrupt 0 of the Hawk ASIC (MPIC).

Note The SCSI device on the IPMC761 uses the standard INTA# pin J11-04 of PMC Slot 1.

IDSEL Routing

Legacy IDSEL assignment for the PCI/ISA Bridge is also maintained to ensure software compatibility between MVME5100 and the MVME2700 while in 761 mode

This is accomplished by using the on-board IPMC761 connector to route IDSEL (AD11) to the PCI/ISA Bridge on the IPMC761.

Note The SCSI device on the IPMC761 uses the standard IDSEL pin J12-25 connected to AD16.

When a standard PMC card (not the IPMC761) is plugged into slot 1, its IDSEL assignment shall correspond to the standard IDSEL pin J12-25 and shall be connected to AD16.

Pin Assignments 5

Introduction

This chapter provides information on pin assignments for various jumpers and connectors on the MVME5100 Single Board Computer.

Summary

The following tables summarize all of the jumpers and connectors:

Jumper	Description
J1	Riscwatch Header
J2	Pal Programming Header
J4	Ethernet Port 2 Configuration
J5	Planner Serial Port 2
J6, J20	Operation Mode Jumpers
J7	Flash Memory Selection
J10, J17	Ethernet Port Selection
J15	System Controller (VME)
J16	Soldered Flash Protection

Connector	Description
J3	IPMC761 Interface
J8	Memory Expansion
J25	PCI Expansion Interface
J11 - J14	PMC Interface (Slot 1)
J21 - J24	PMC Interface (Slot 2)
P1, P2	VMEbus Interface
J9 J18	Ethernet Interface (LAN1) Ethernet Interface (LAN2)
J19	COM1 Interface
J5	COM2 Interface

Jumper Settings

The following table provides information about the jumper settings associated with th MVME5100 Single Board Computer. The table below provides a brief description of each jumper and the appropriate setting(s) for proper board operation.

Table 5-1. Jumper Switches and Settings

Jumper	Description	Setting	Default
J1	Riscwatch Header	None (Factory Use Only)	N/A
J2	Pal Programming Header	None (Lab Use Only)	N/A
J4	Ethernet Port 2 Selection (set in conjunction with jumpers J10 and J17)	For “P2” Ethernet Port 2: Pins 1,2; 3,4; 5,6; 7,8	No Jumper Installed
		For “Front Panel” Ethernet Port 2: No Jumpers Installed	
J5	On-Board Serial Port (COM2)	Not Available as of Publication Date	N/A
J6, J20	Operation Mode (Set Both Jumpers)	Pins 1,2 for PMC Mode	PMC Mode
		Pins 2,3 for 761 Mode	
J7	Flash Memory Selection at Boot	Pins 1,2 for Soldered Bank A	Socketed Bank B
		Pins 2,3 for Socketed Bank B	
J10, J17	Ethernet Port 2 Selection (set in conjunction with jumper J4)	For “Front Panel” Ethernet Port 2: Pins 1,3 and 2,4 on Both Jumpers	Front Panel Ethernet Port 2
		For “P2” Ethernet Port 2: Pins 3,5 and 4,6 on Both Jumpers	
J15	System Controller (VME)	Pins 1,2 for No SCON	Auto SCON
		Pins 2,3 for Auto SCON	
		No Jumper for ALWAYS SCON	
J16	Soldered Flash Protection	Pins 1,2 Enables Programming of Flash	Flash Prog. Enabled
		Pins 2,3 Disables Programming of Flash	

Connectors

IPMC761 Connector (J3) Pin Assignments

This connetor is used to provide an interface to the IPMC761 module signals and is located near J11. The pin assignments for this connector are as follows:

Table 5-2. IPMC761 Connector Pin Assignments

Pin	Assignment			Pin
1	I2CSCL		I2CSDA	**2**
3	GND		GND	**4**
5	DB8#		GND	**6**
7	GND		DB9#	**8**
9	DB10#		+3.3V	**10**
11	+3.3V		DB11#	**12**
13	DB12#		GND	**14**
15	GND		DB13#	**16**
17	DB14#		+3.3V	**18**
19	+3.3V		DB15#	**20**
21	DBP1#		GND	**22**
23	GND		LANINT2_L	**24**
25	PIB_INT		+3.3V	**26**
27	+3.3V		PIB_PMCREQ#	**28**
29	PIB_PMCGNT#		GND	**30**
31	GND		+3.3V	**32**
33	+5.0V		+5.0V	**34**
35	GND		GND	**36**
37	+5.0V		+5.0V	**38**
39	GND		GND	**40**

Memory Expansion Connector (J8) Pin Assignments

This connector is used to provide memory expansion capability. A single memory mezzanine card provides a maximum of 256MB of memory. Attaching another memory mezzanine to the first mezzanine provides an additional 512Mbytes of expansion memory. The pin assignments for this connector are as follows:

Table 5-3. Memory Expansion Connector Pin Assignments

Pin	Assignment		Pin
1	GND	GND	**2**
3	DQ00	DQ01	**4**
5	DQ02	DQ03	**6**
7	DQ04	DQ05	**8**
9	DQ06	DQ07	**10**
11	+3.3V	+3.3V	**12**
13	DQ08	DQ09	**14**
15	DQ10	DQ11	**16**
17	DQ12	DQ13	**18**
19	DQ14	DQ15	**20**
21	GND	GND	**22**
23	DQ16	DQ17	**24**
25	DQ18	DQ19	**26**
27	DQ20	DQ21	**28**
29	DQ22	DQ23	**30**
31	+3.3V	+3.3V	**32**
33	DQ24	DQ25	**34**
35	DQ26	DQ27	**36**
37	DQ28	DQ29	**38**
39	DQ30	DQ31	**40**
41	GND	GND	**42**
43	DQ32	DQ33	**44**
45	DQ34	DQ35	**46**

Table 5-3. Memory Expansion Connector Pin Assignments (Continued)

Pin	Assignment		Pin
47	DQ36	DQ37	**48**
49	DQ38	DQ39	**50**
51	+3.3V	+3.3V	**52**
53	DQ40	DQ41	**54**
55	DQ42	DQ43	**56**
57	DQ44	DQ45	**58**
59	DQ46	DQ47	**60**
61	GND	GND	**62**
63	DQ48	DQ49	**64**
65	DQ50	DQ51	**66**
67	DQ52	DQ53	**68**
69	+3.3V	+3.3V	**70**
71	DQ54	DQ55	**72**
73	DQ56	DQ57	**74**
75	DQ58	DQ59	**76**
77	DQ60	DQ61	**78**
79	GND	GND	**80**
81	DQ62	DQ63	**82**
83	CKD00	CKD01	**84**
85	CKD02	CKD03	**86**
87	CKD04	CKD05	**88**
89	+3.3V	+3.3V	**90**
91	CKD06	CKD07	**92**
93	BA1	BA0	**94**
95	A12	A11	**96**
97	A10	A09	**98**
99	GND	GND	**100**
101	A08	A07	**102**
103	A06	A05	**104**
105	A04	A03	**106**

Table 5-3. Memory Expansion Connector Pin Assignments (Continued)

Pin	Assignment			Pin
107	A02		A01	**108**
109	+3.3V		+3.3V	**110**
111	A00		CS_C0_L	**112**
113	CS_E0_L		GND	**114**
115	CS_C1_L		CS_E1_L	**116**
117	WE_L		RAS_L	**118**
119	GND		GND	**120**
121	CAS_L		+3.3V	**122**
123	+3.3V		DQMB0	**124**
125	DQMB1		SCL	**126**
127	SDA		A1_SPD	**128**
129	A0_SPD		MEZZ1_L	**130**
131	MEZZ2_L		GND	**132**
133	GND		SDRAMCLK1	**134**
135	SDRAMCLK3		+3.3V	**136**
137	SDRAMCLK4		SDRAMCLK2	**138**
139	GND		GND	**140**

Note PIN 130, 131, MEZZ1_L, MEZZ2_L, configures the board's local bus frequency. If a single mezzanine is attached to the board, MEZZ1_L will be pulled down on the board. If a second mezzanine is attached on-top to the first, MEZZ2_L will be pulled down on the board. This may cause the clock generation logic to set the local bus frequency to 83.33 MHz if necessary.

PCI Expansion Connector (J25) Pin Assignments

This connector is used to provide PCI/PMC expansion capability. The pin assignments for this connector are as follows:

Table 5-4. PCI Expansion Connector Pin Assignments

Pin	Assignment			Pin
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 5-4. PCI Expansion Connector Pin Assignments (Continued)

Pin	Assignment			Pin
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 5-4. PCI Expansion Connector Pin Assignments (Continued)

Pin	Assignment			Pin
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**

PCI Mezzanine Card (PMC) Connectors

These connectors provide 32/64-bit PCI interfaces and P2 I/O for two optional add-on PCI Mezzanine Cards (PMC). The pin assignments for these connectors are as follows:

Table 5-5. PMC Slot 1 Connector (J11) Pin Assignments

Pin	Assignment		Pin
1	TCK	-12V	**2**
3	GND	INTA#	**4**
5	INTB#	INTC#	**6**
7	PMCPRSNT1#	+5V	**8**
9	INTD#	Not Used	**10**
11	GND	Not Used	**12**
13	CLK	GND	**14**
15	GND	PMCGNT1#	**16**
17	PMCREQ1#	+5V	**18**
19	+5V (Vio)	AD31	**20**
21	AD28	AD27	**22**
23	AD25	GND	**24**
25	GND	C/BE3#	**26**
27	AD22	AD21	**28**
29	AD19	+5V	**30**
31	+5V (Vio)	AD17	**32**
33	FRAME#	GND	**34**
35	GND	IRDY#	**36**
37	DEVSEL#	+5V	**38**
39	GND	LOCK#	**40**

Table 5-5. PMC Slot 1 Connector (J11) Pin Assignments (Continued)

Pin	Assignment			Pin
41	SDONE#		SBO#	**42**
43	PAR		GND	**44**
45	+5V (Vio)		AD15	**46**
47	AD12		AD11	**48**
49	AD09		+5V	**50**
51	GND		C/BE0#	**52**
53	AD06		AD05	**54**
55	AD04		GND	**56**
57	+5V (Vio)		AD03	**58**
59	AD02		AD01	**60**
61	AD00		+5V	**62**
63	GND		REQ64#	**64**

Table 5-6. PMC Slot 1 Connector (J12) Pin Assignments

Pin	Assignment			Pin
1	+12V		TRST#	**2**
3	TMS		TDO	**4**
5	TDI		GND	**6**
7	GND		Not Used	**8**
9	Not Used		Not Used	**10**
11	Pull-up		+3.3V	**12**
13	RST#		Pull-down	**14**
15	+3.3V		Pull-down	**16**
17	Not Used		GND	**18**

Table 5-6. PMC Slot 1 Connector (J12) Pin Assignments (Continued)

Pin	Assignment			Pin
19	AD30		AD29	**20**
21	GND		AD26	**22**
23	AD24		+3.3V	**24**
25	IDSEL1		AD23	**26**
27	+3.3V		AD20	**28**
29	AD18		GND	**30**
31	AD16		C/BE2#	**32**
33	GND		Not Used	**34**
35	TDRY#		+3.3V	**36**
37	GND		STOP#	**38**
39	PERR#		GND	**40**
41	+3.3V		SERR#	**42**
43	C/BE1#		GND	**44**
45	AD14		AD13	**46**
47	GND		AD10	**48**
49	AD08		+3.3V	**50**
51	AD07		Not Used	**52**
53	+3.3V		Not Used	**54**
55	Not Used		GND	**56**
57	Not Used		Not Used	**58**
59	GND		Not Used	**60**
61	ACK64#		+3.3V	**62**
63	GND		Not Used	**64**

Table 5-7. PMC Slot 1 Connector (J13) Pin Assignments

Pin	Assignment			Pin
1	Reserved		GND	**2**
3	GND		C/BE7#	**4**
5	C/BE6#		C/BE5#	**6**
7	C/BE4#		GND	**8**
9	+5V (Vio)		PAR64	**10**
11	AD63		AD62	**12**
13	AD61		GND	**14**
15	GND		AD60	**16**
17	AD59		AD58	**18**
19	AD57		GND	**20**
21	+5V (Vio)		AD56	**22**
23	AD55		AD54	**24**
25	AD53		GND	**26**
27	GND		AD52	**28**
29	AD51		AD50	**30**
31	AD49		GND	**32**
33	GND		AD48	**34**
35	AD47		AD46	**36**
37	AD45		GND	**38**
39	+5V (Vio)		AD44	**40**
41	AD43		AD42	**42**
43	AD41		GND	**44**
45	GND		AD40	**46**

Table 5-7. PMC Slot 1 Connector (J13) Pin Assignments (Continued)

Pin	Assignment			Pin
47	AD39		AD38	**48**
49	AD37		GND	**50**
51	GND		AD36	**52**
53	AD35		AD34	**54**
55	AD33		GND	**56**
57	+5V (Vio)		AD32	**58**
59	Reserved		Reserved	**60**
61	Reserved		GND	**62**
63	GND		Reserved	**64**

Table 5-8. PMC Slot 1 Connector (J14) Pin Assignments

Pin	Assignment			PIN
1	Jumper Configurable		PMC1_2 (P2-A1)	**2**
3	Jumper Configurable		PMC1_4 (P2-A2)	**4**
5	Jumper Configurable		PMC1_6 (P2-A3)	**6**
7	Jumper Configurable		PMC1_8 (P2-A4)	**8**
9	PMC1 _9 (P2-C5)		PMC1_10 (P2-A5)	**10**
11	PMC1_11 (P2-C6)		PMC1_12 (P2-A6)	**12**
13	PMC1_13 (P2-C7)		PMC1_14 (P2-A7)	**14**
15	PMC1_15 (P2-C8)		PMC1_16 (P2-A8)	**16**
17	PMC1_17 (P2-C9)		PMC1_18 (P2-A9)	**18**
19	PMC1_19 (P2-C10)		PMC1_20 (P2-A10)	**20**
21	PMC1_21 (P2-C11)		PMC1_22 (P2-A11)	**22**
23	PMC1_23 (P2-C12)		PMC1_24 (P2-A12)	**24**

Table 5-8. PMC Slot 1 Connector (J14) Pin Assignments (Continued)

Pin	Assignment		PIN
25	PMC1_25 (P2-C13)	PMC1_26 (P2-A13)	**26**
27	PMC1_27 (P2-C14)	PMC1_28 (P2-A14)	**28**
29	PMC1_29 (P2-C15)	PMC1_30 (P2-A15)	**30**
31	PMC1_31 (P2-C16)	PMC1_32 (P2-A16)	**32**
33	PMC1_33 (P2-C17)	PMC1_34 (P2-A17)	**34**
35	PMC1_35 (P2-C18)	PMC1_36 (P2-A18)	**36**
37	PMC1_37 (P2-C19)	PMC1_38 (P2-A19)	**38**
39	PMC1_39 (P2-C20)	PMC1_40 (P2-A20)	**40**
41	PMC1_41 (P2-C21)	PMC1_42 (P2-A21)	**42**
43	PMC1_43 (P2-C22)	PMC1_44 (P2-A22)	**44**
45	PMC1_45 (P2-C23)	PMC1_46 (P2-A23)	**46**
47	PMC1_47 (P2-C24)	PMC1_48 (P2-A24)	**48**
49	PMC1_49 (P2-C25)	PMC1_50 (P2-A25)	**50**
51	PMC1_51 (P2-C26)	PMC1_52 (P2-A26)	**52**
53	PMC1_53 (P2-C27)	PMC1_54 (P2-A27)	**54**
55	PMC1_55 (P2-C28)	PMC1_56 (P2-A28)	**56**
57	PMC1_57 (P2-C29)	PMC1_58 (P2-A29)	**58**
59	PMC1_59 (P2-C30)	PMC1_60 (P2-A30)	**60**
61	PMC1_61 (P2-C31)	PMC1_62 (P2-A31)	**62**
63	PMC1_63 (P2-C32)	PMC1_64 (P2-A32)	**64**

Note Jumper configuration is dependent upon P2 I/O mode chosen (PMC or MVME761 Mode).

Table 5-9. PMC Slot 2 Connector (J21) Pin Assignments

Pin	Assignment		Pin
1	TCK	-12V	**2**
3	GND	INTA#	**4**
5	INTB#	INTC#	**6**
7	PMCPRSNT2#	+5V	**8**
9	INTD#	Not Used	**10**
11	GND	Not Used	**12**
13	CLK	GND	**14**
15	GND	PMCGNT2#	**16**
17	PMCREQ2#	+5V	**18**
19	+5V (Vio)	AD31	**20**
21	AD28	AD27	**22**
23	AD25	GND	**24**
25	GND	C/BE3#	**26**
27	AD22	AD21	**28**
29	AD19	+5V	**30**
31	+5V (Vio)	AD17	**32**
33	FRAME#	GND	**34**
35	GND	IRDY#	**36**
37	DEVSEL#	+5V	**38**
39	GND	LOCK#	**40**
41	SDONE#	SBO#	**42**
43	PAR	GND	**44**
45	+5V (Vio)	AD15	**46**
47	AD12	AD11	**48**

Table 5-9. PMC Slot 2 Connector (J21) Pin Assignments (Continued)

Pin	Assignment			Pin
49	AD09		+5V	**50**
51	GND		C/BE0#	**52**
53	AD06		AD05	**54**
55	AD04		GND	**56**
57	+5V (Vio)		AD03	**58**
59	AD02		AD01	**60**
61	AD00		+5V	**62**
63	GND		REQ64#	**64**

Table 5-10. PMC Slot 2 Connector (J22) Pin Assignments

Pin	Assignment			Pin
1	+12V		TRST#	**2**
3	TMS		TDO	**4**
5	TDI		GND	**6**
7	GND		Not Used	**8**
9	Not Used		Not Used	**10**
11	Pull-up		+3.3V	**12**
13	RST#		Pull-down	**14**
15	+3.3V		Pull-down	**16**
17	Not Used		GND	**18**
19	AD30		AD29	**20**
21	GND		AD26	**22**

Table 5-10. PMC Slot 2 Connector (J22) Pin Assignments (Continued)

Pin	Assignment			Pin
23	AD24		+3.3V	**24**
25	IDSEL2		AD23	**26**
27	+3.3V		AD20	**28**
29	AD18		GND	**30**
31	AD16		C/BE2#	**32**
33	GND		Not Used	**34**
35	TDRY#		+3.3V	**36**
37	GND		STOP#	**38**
39	PERR#		GND	**40**
41	+3.3V		SERR#	**42**
43	C/BE1#		GND	**44**
45	AD14		AD13	**46**
47	GND		AD10	**48**
49	AD08		+3.3V	**50**
51	AD07		Not Used	**52**
53	+3.3V		Not Used	**54**
55	Not Used		GND	**56**
57	Not Used		Not Used	**58**
59	GND		Not Used	**60**
61	ACK64#		+3.3V	**62**
63	GND		Not Used	**64**

Table 5-11. PMC Slot 2 Connector (J23) Pin Assignments

Pin	Assignment		Assignment	Pin
1	Reserved		GND	**2**
3	GND		C/BE7#	**4**
5	C/BE6#		C/BE5#	**6**
7	C/BE4#		GND	**8**
9	+5V (Vio)		PAR64	**10**
11	AD63		AD62	**12**
13	AD61		GND	**14**
15	GND		AD60	**16**
17	AD59		AD58	**18**
19	AD57		GND	**20**
21	+5V (Vio)		AD56	**22**
23	AD55		AD54	**24**
25	AD53		GND	**26**
27	GND		AD52	**28**
29	AD51		AD50	**30**
31	AD49		GND	**32**
33	GND		AD48	**34**
35	AD47		AD46	**36**
37	AD45		GND	**38**
39	+5V (Vio)		AD44	**40**
41	AD43		AD42	**42**
43	AD41		GND	**44**
45	GND		AD40	**46**

Table 5-11. PMC Slot 2 Connector (J23) Pin Assignments (Continued)

Pin	Assignment			Pin
47	AD39		AD38	**48**
49	AD37		GND	**50**
51	GND		AD36	**52**
53	AD35		AD34	**54**
55	AD33		GND	**56**
57	+5V (Vio)		AD32	**58**
59	Reserved		Reserved	**60**
61	Reserved		GND	**62**
63	GND		Reserved	**64**

Table 5-12. PMC Slot 2 Connector (J24) Pin Assignments

Pin	Assignment			Pin
1	PMC2_1 (P2-D1)		PMC2_2 (P2-Z1)	**2**
3	PMC2_3 (P2-D2)		PMC2_4 (P2-D3)	**4**
5	PMC2_5 (P2-Z3)		PMC2_6 (P2-D4)	**6**
7	PMC2_7 (P2-D5)		PMC2_8 (P2-Z5)	**8**
9	PMC2_9 (P2-D6)		PMC2_10 (P2-D7)	**10**
11	PMC2_11 (P2-Z7)		PMC2_12 (P2-D8)	**12**
13	PMC2_13 (P2-D9)		PMC2_14 (P2-Z9)	**14**
15	PMC2_15 (P2-D10		PMC2_16 (P2-D11)	**16**
17	PMC2_17 (P2-Z11)		PMC2_18 (P2-D12)	**18**
19	PMC2_19 (P2-D13)		PMC2_20 (P2-Z13)	**20**
21	PMC2_21 (P2-D14)		PMC2_22 (P2-D15)	**22**
23	PMC2_23 (P2-Z15)		PMC2_24 (P2-D16)	**24**

Table 5-12. PMC Slot 2 Connector (J24) Pin Assignments (Continued)

Pin	Assignment		Pin
25	PMC2_25 (P2-D17)	PMC2_26 (P2-Z17)	**26**
27	PMC2_27 (P2-D18)	PMC2_28 (P2-D19)	**28**
29	PMC2_29 (P2-Z19)	PMC2_30 (P2-D20)	**30**
31	PMC2_31 (P2-D21)	PMC2_32 (P2-Z21)	**32**
33	PMC2_33 (P2-D22	PMC2_34 (P2-D23)	**34**
35	PMC2_35 (P2-Z23)	PMC2_36 (P2-D24)	**36**
37	PMC2_37 (P2-D25)	PMC2_38 (P2-Z25	**38**
39	PMC2_39 (P2-D26)	PMC2_40 (P2-D27)	**40**
41	PMC2_41 (P2-Z27)	PMC2_42 (P2-D28)	**42**
43	PMC2_43 (P2-D29)	PMC2_44 (P2-Z29)	**44**
45	PMC2_45 (P2-D30)	PMC2_46 (P2-Z31)	**46**
47	Not Used	Not Used	**48**
49	Not Used	Not Used	**50**
51	Not Used	Not Used	**52**
53	Not Used	Not Used	**54**
55	Not Used	Not Used	**56**
57	Not Used	Not Used	**58**
59	Not Used	Not Used	**60**
61	Not Used	Not Used	**62**
63	Not Used	Not Used	**64**

VMEbus Connectors P1 and P2 Pin Assignments (PMC mode)

The VMEbus connector P1 provides power and VME signals for 24-bit address and 16-bit data. The pin assignments for the connector are specified by the IEEE P1014-1987 VMEbus Specification and the VME64 Extension Standard.

Row B of connector P2 provides power to the MVME5100, and to the upper eight VMEbus address lines, and additional 16 VMEbus data lines. Rows A, C, Z, and D provide power and interface signals to the MVME762 transition module. The pin assignments for connector P2 in PMC mode are as follows:

Table 5-13. VMEbus Connector P2 Pin Assignments (PMC Mode)

Pin	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)
2	GND	PMC1_4 (J14-4)	GND	PMC1_3 (J14-3)	PMC2_3 (J24-3)
3	PMC2_5 (J24-5)	PMC1_6 (J14-6)	RETRY#	PMC1_5 (J14-5)	PMC2_4 (J24-4)
4	GND	PMC1_8 (J14-8)	VA24	PMC1_7 (J14-7)	PMC2_6 (J24-6)
5	PMC2_8 (J24-8)	PMC1_10 (J14-10)	VA25	PMC1_9 (J14-9)	PMC2_7 (J24-7)
6	GND	PMC1_12 (J14-12)	VA26	PMC1_11 (J14-11)	PMC2_9 (J24-9)
7	PMC2_11(J24-11)	PMC1_14 (J14-14)	VA27	PMC1_13 (J14-13)	PMC2_10 (J24-10)
8	GND	PMC1_16 (J14-16)	VA28	PMC1_15 (J14-15)	PMC2_12 (J24-12)
9	PMC2_14 (J24-14)	PMC1_18 (J14-18)	VA29	PMC1_17 (J14-17)	PMC2_13 (J24-13)
10	GND	PMC1_20 (J14-20)	VA30	PMC1_19 (J14-19)	PMC2_15 (J24-15)
11	PMC2_17 (J24-17)	PMC1_22 (J14-22)	VA31	PMC1_21 (J14-21)	PMC2_16 (J24-16)
12	GND	PMC1_24 (J14-24)	GND	PMC1_23 (J14-23)	PMC2_18 (J24-18)
13	PMC2_20 (J24-20)	PMC1_26 (J14-26)	+5V	PMC1_25 (J14-25)	PMC2_19 (J24-19)

Table 5-13. VMEbus Connector P2 Pin Assignments (PMC Mode) (Continued)

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

VMEbus P1 & P2 Connectors Assignments (MVME761 Mode)

The VMEbus connector P1 provides power and VME signals for 24-bit address and 16-bit data. The pin assignments for the connector are specified by the IEEE P1014-1987 VMEbus Specification and the VME64 Extension Standard.

Row B of connector P2 provides power to the MVME5100 and to the upper 8 VMEbus address lines and additional 16 VMEbus data lines. Rows A, C, Z, and D provide power and interface signals to the MVME761 transition module in 761 mode.

It is important to note that the PMC I/O routing to row D and Z are not the same as MVME2600/2700. The PMC I/O routing for row D and row Z is the same as the PMC mode with the exception of pins Z1, 3, 5, 7, 9, 11, 13, 15, and 17 which are used for extended SCSI.

Note A PMC card installed in slot 2 of a MVME5100 in 761 mode MUST NOT connect to J24-2, 5, 8, 11, 14, 17, 20, 23, and 26 since they are connected to the extended SCSI signals of the MVME5100.

The pin assignments for connector P2 in MVME761 mode are as follows:

Table 5-14. VMEbus Connector P2 Pin Assignments (MVME761 Mode)

Pin	Row Z	Row A	Row B	Row C	Row D
1	DB8#	DB0#	+5V	RD- (10/100)	PMC2_1 (J24-1)
2	GND	DB1#	GND	RD+ (10/100)	PMC2_3 (J24-3)
3	DB9#	DB2#	RETRY#	TD- (10/100)	PMC2_4 (J24-4)
4	GND	DB3#	VA24	TD+ (10/100)	PMC2_6 (J24-6)
5	DB10#	DB4#	VA25	Not Used	PMC2_7 (J24-7)
6	GND	DB5#	VA26	Not Used	PMC2_9 (J24-9)
7	DB11#	DB6#	VA27	+12VF	PMC2_10 (J24-10)
8	GND	DB7#	VA28	PRSTB#	PMC2_12 (J24-12)
9	DB12#	DBP#	VA29	PRD0	PMC2_13 (J24-13)

Table 5-14. VMEbus Connector P2 Pin Assignments (MVME761 Mode) (Continued)

Pin	Row Z	Row A	Row B	Row C	Row D
10	GND	ATN#	VA30	PRD1	PMC2_15 (J24-15)
11	DB13#	BSY#	VA31	PRD2	PMC2_16 (J24-16)
12	GND	ACK#	GND	PRD3	PMC2_18 (J24-18)
13	DB14#	RST#	+5V	PRD4	PMC2_19 (J24-19)
14	GND	MSG#	VD16	PRD5	PMC2_21 (J24-21)
15	DB15#	SEL#	VD17	PRD6	PMC2_22 (J24-22)
16	GND	D/C#	VD18	PRD7	PMC2_24 (J24-24)
17	DBP1#	REQ#	VD19	PRACK#	PMC2_25 (J24-25)
18	GND	O/I#	VD20	PRBSY	PMC2_27 (J24-27)
19	PMC2_29 (J24-29)	AFD#	VD21	PRPE	PMC2_28 (J24-28)
20	GND	SLIN#	VD22	PRSEL	PMC2_30 (J24-30)
21	PMC2_32 (J24-32)	TXD3	VD23	INIT#	PMC2_31 (J24-31)
22	GND	RXD3	GND	PRFLT#	PMC2_33 (J24-33)
23	PMC2_35 (J24-35)	RTXC3	VD24	TXD1_232	PMC2_34 (J24-34)
24	GND	TRXC3	VD25	RXD1_232	PMC2_36 (J24-36)
25	PMC2_38 (J24-38)	TXD3	VD26	RTS1_232	PMC2_37 (J24-37)
26	GND	RXD3	VD27	CTS1_232	PMC2_39 (J24-39)
27	PMC2_41 (J24-41)	RTXC4	VD28	TXD2_232	PMC2_40 (J24-40)
28	GND	TRXC4	VD29	RXD2_232	PMC2_42 (J24-42)
29	PMC2_44 (J24-44)		VD30	RTS2_232	PMC2_43 (J24-43)
30	GND	-12VF	VD31	CTS2_232	PMC2_45 (J24-45)
31	PMC2_46 (J24-46)	MSYNC#	GND	MDO	GND
32	GND	MCLK	+5V	MDI	VPC

Note Rows A and C and Z's (Z1, 3, 5 , 7, 9, 11, 13, 15, and 17) functionality is provided by the IPMC761 in slot 1 and the MVME5100 Ethernet port 2.

P2 Input/Output Connector Pin Assignments

The table below provides connector P2 Input/Output (I/O) mode jumper settings for the PMC P2 I/O mode and the MVME761 P2 I/O mode.

For PMC mode, place all jumpers (4) to connect the top row and the middle row. For MVME 761 mode, place all jumpers (4) to connect the bottom row and the middle row.

Table 5-15. P2 I/O mode Connector

TOP ROW		MIDDLE ROW	BOTTOM ROW
PIN	**PMC MODE JUMPER**	**MVME5100 P2 I/O Pin**	**MVME761 MODE JUMPER**
1	PMC1_1 (J14-1)	C1	RD- (10/100)
2	PMC1_3 (J14-3)	C2	RD+ (10/100)
3	PMC1_5 (J14-5)	C3	TD- (10/100)
4	PMC1_7 (J14-7)	C4	TD+ (10/100)

10 Base-T/100 Base-Tx Connector Pin Assignments

The board's dual 10 Base-T/100 Base-Tx RJ45 connectors are located on the front plate. The pin assignments for these connector's are as follows:

Table 5-16. 10 Base-T/100 Base-Tx Connector Pin Assignment

Pin	Assignment
1	TD+
2	TD-
3	RD+
4	AC Terminated
5	AC Terminated
6	RD-
7	AC Terminated
8	AC Terminated

COM1 and COM2 Connector Pin Assignments

A standard RJ45 connector located on the front panel and a 9-pin header located near the bottom edge of the MVME5100 provides the interface to the serial debug ports. The RJ45 connector is for COM1 and the 9-pin header is for COM2.

The pin assignments for these connectors are as follows:

Table 5-17. COM1 Connector Pin Assignments

Pin	Assignment
1	DCD
2	RTS
3	GNDC
4	TXD
5	RXD
6	GNDC
7	CTS
8	DTR
9	GNDC

Table 5-18. COM2 Connector Pin Assignments

Pin	Assignment
1	DCD
2	DSR
3	RXD
4	RTS
5	TXD
6	CTS
7	DTR
8	RI
9	GND

Specifications

A

This appendix lists general specifications and power characteristics for the MVME5100 Single Board Computer. It also provides information on cooling requirements.

A complete functional description of the MVME5100 Single Board Computer appears in Chapter 4, *Functional Description*. Specifications for the optional PMC modules can be found in the documentation for those modules.

General Specifications

The following table lists general specifications for MVME5100 Single Board Computer.

Table A-1. MVME5100 Specifications

Characteristic	Specification
Operating Temperature	0° C to 55° C (commercial) and - 20° C to 71° C (industrial) inlet air temperature with forced air cooling. 400 LFM (Linear Feet per Minute) of forced air cooling is recommended for operation in the upper temperature range.
Storage Temperature	- 40° C to +85° C
Relative Humidity	5% to 90% Non-Condensing
Physical Dimensions	
Height	233.4 mm (9.2 in.)
Depth	160 mm (6.3 in.)
Front Panel Height	261.8 mm (10.3 in.)
Width	19.8 mm (0.8 in.)
Max. Component Height	14.8 mm (0.58 in.)

Power Requirements

Power requirements for the MVME5100 Single Board Computer depend on the configuration of the board. The table below lists the typical and maximum power consumption of the board.

Table A-2. Power Consumption

Supply Voltage	Amps (Typical)	Amps (Maximum)
+5V (±5%)	3.2 A	4.0 A
+12V (±10%)	0.2 A	0.5 A
-12V (±10%)	0.1 A	0.3 A

Note The power requirements for the MVME5100 do not include the power requirements for the PMC or IMPC761Modules. The PMC specification allows for 7.5 watts per PMC slot. The 15 watts total can be drawn from any combination of the three voltage sources provided by the MVME5100: +5V, +12V, and -12V.

Cooling Requirements

The MVME5100 is specified, designed, and tested to operate reliably with an incoming air temperature range from 0° C to 55° C (commercial) or -20° C to 71° C (industial) with forced air cooling of the entire assembly (board and expansion modules) at a velocity typically achievable by using a 100 CFM axial fan. Note that 400 LFM (Linear Feet per Minute) of forced air cooling is recommended for operation in the upper temperature range.

Temperature qualification is performed in a Motorola development 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 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 400 LFM flowing over the module. Less airflow is required to cool the module in environments having lower maximum ambient temperature. Under more favorable thermal conditions, it may be possible to operate the module reliably at higher than 71° 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.

Solving Startup Problems

In the event of difficulty with your MVME5100, perform the simple troubleshooting steps listed in the table below before calling for help or sending the board back for repair.

Some of the procedures will return the board to the factory debugger environment. It is important to note that the Board was tested under these conditions before it left the factory. The selftests may not run in all user-customized environments.

Table B-1. Troubleshooting Problems

Condition	Possible Problem	Possible Resolution:
I. Nothing works; no display on the terminal.	A. If the LEDs are not lit, the board may not be getting 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. 4. Review the Installation and Startup procedures in this manual. They include a step-by-step powerup routine.
	B. If the LEDs are lit, the board may be in the wrong slot.	1. The MVME5100 should be in the first (leftmost) slot. 2. Check if the “system controller” function on the board is enabled per the instructions this manual.
	C. The “system console” terminal may be configured incorrectly.	Configure the system console terminal per the instructions this manual.

Table B-1. Troubleshooting Problems (Continued)

Condition	Possible Problem	Possible Resolution:
II. There is a display on the terminal; however, keyboard and/or mouse input 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 the instructions in 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 PPC6-Bug> does not appear at powerup; 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 using the ABT/RST switch (press and hold switch down, approximately 3 - 5 seconds). 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.	
IV. Debug prompt PPC6-Bug> appears at powerup; 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. **CAUTION:** Performing the next step (**env;d**) will change some parameters that may affect your system's operation.
	B. There may be some fault in the board hardware.	

Table B-1. Troubleshooting Problems (Continued)

Condition	Possible Problem	Possible Resolution:
IV. Debug prompt PPC6-Bug> appears at powerup; the board does not autoboot *(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: PPC6-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 selftests 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 selftest is a valuable tool in isolating defects.) 8. The system may indicate that it has passed all the selftests. 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 B-1. Troubleshooting Problems (Continued)

Condition	Possible Problem	Possible Resolution:
V. The debugger is in system mode; the board autoboots, 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; 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 C

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 Motorola Computer Group's World Wide Web literature site, http://www.motorola.com/computer/literature

Table C-1. Motorola Computer Group Documents

Document Title	Motorola Publication Number
MVME5100 Single Board Computer Programmer's Reference Guide	V5100A/PG
PPCBug Firmware Package User's Manual, Part 1 of 2	PPCBUGA1/UM
PPCBug Firmware Package User's Manual, Part 2 of 2	PPCBUGA2/UM
PPCBug Diagnostics Manual	PPCDIAA/UM
PMCspan PMC Adapter Carrier Module Installation and Use	PMCSPANA/IH

To obtain the most up-to-date product information in PDF or HTML format, visit http://www.motorola.com/computer/literature.

Manufacturers' Documents

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

Note In many cases, the information is preliminary and the revision levels of the documents are subject to change without notice.

Table C-2. Manufacturers' Documents

Document Title	Publication Number
MPC750 RISC Microprocessor Users Manual Motorola Literature Distribution Center Telephone: (800) 441-2447 or (303) 675-2140	MPC750UM/AD
MPC7400 RISC Microprocessor Users Manual Motorola Literature Distribution Center Telephone: (800) 441-2447 or (303) 675-2140	MPC7400UM/D
Universe II User Manual Tundra Semiconductor Corporation 603 March Road, Kanata, ON, Canada K2K 2M5 1-800-267-7231, (613) 592-0714, Fax: (613) 592-1320	N/A
PowerPlus II Vital Product Data Engineering Specification (Revision 0.1) Motorola Literature Distribution Center Telephone: (800) 441-2447 or (303) 675-2140	N/A

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.

Note In many cases, the information is preliminary and the revision levels of the documents are subject to change without notice.

Table C-3. Related Specifications

Document Title and Source	Publication Number
Peripheral Component Interconnect (PCI) Interface Specification, Revision 2.1 PCI Special Interest Group P.O. Box 14070 Portland, Oregon 97214-4070 Marketing/Help Line: Telephone: (503) 696-6111 Document/Specification Ordering: Telephone: 1-800-433-5177or (503) 797-4207 FAX: (503) 234-6762	PCI Local Bus Specification
Common Mezzanine Card Specification IEEE Standards Department 445 Hoes Lane, P.O Box 1331 Piscataway, NJ 08855-1331	P1386 Draft 2.0
PCI Mezzanine Card Specification IEEE Standards Department 445 Hoes Lane, P.O Box 1331 Piscataway, NJ 08855-1331	P1386.1 Draft 2.0

Index

INDEX

R

S

T

U

V

INDEX