MVME2100
Single Board Computer
Installation and Use

V2100A/IH1

May 2000 Edition

Notice

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

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 The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of Motorola, Inc.

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

Limited and Restricted Rights Legend

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

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

Motorola, Inc.
Computer Group
2900 South Diablo Way
Tempe, Arizona 85282

Preface

The *MVME2100 Single Board Computer Installation and Use* manual provides the information you will need to install and use your MVME2100 Single Board Compter. The product is based on an MPC8240 microprocessor, and features one 32-bit PMC expansion slot, one Type I and two Type II PC-MIP expansion slots or two Type I and two Type II PC-MIP expansion slots, 32 or 64MB of synchronous DRAM memory, 1MB Boot FLASH ROM, 4 or 8MB expansion FLASH ROM, one 10/100Base-TX Ethernet port, and one front panel accessible asynchronous serial port.

This manual includes hardware preparation and installation instructions, along with information on: using the front panel, using the PPCBug debugging firmware, and other advanced debugger topics. Appendices provide the module's specifications and connector pin assignments. Additional manuals you may wish to obtain are listed in Appendix A.

The information in this manual applies principally to the MVME2100. The PMCspan and PMCs are described briefly; however, they are documented in separate publications, furnished with those products. Refer to the individual product documentation for complete preparation and installation instructions. These manuals are listed in Appendix A.

This manual is intended for anyone who wants to design OEM systems, supply additional capability to an existing compatible system, or work in a lab environment for experimental purposes. A basic knowledge of computers and digital logic is assumed.

Comments and Suggestions

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

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

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

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

Conventions Used in This Manual

The following typographical conventions are used in this document:

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):

$	Specifies a hexadecimal character
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.

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.

CAUTION

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.

CAUTION

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.

Contents

CHAPTER 1 Preparation and Installation

CHAPTER 2 Operating Instructions

CHAPTER 3 Functional Description

CHAPTER 4 PPCBug Firmware

List of Figures

List of Tables

Preparation and Installation 1

Introduction

This chapter provides general product information along with hardware preparation, installation, and operating instructions for the MVME2100 Single Board Computer.

Note Unless otherwise specified, the designation “MVME2100” refers to all models of the MVME2100-series Single Board Computers.

Description

The MVME2100x is a single board computer (SBC) with support for PCI mezzanine cards. It is based on the MPC8240 microprocessor.

The right front panel cutout provides access to PMC I/O. One single-width PMC can be installed.

The left front panel cutout provides access to Type II PC-MIP I/O (SCSI, VGA, etc.) Up to two PC-MIP cards can be added.

The center of the MVME2100 SBC can accommodate one Type I PC-MIP card (on connectors P21, P22, and P23). It should be noted that any Type I PC-MIP card that does not incorporate I/O (such as flash memory) can also be installed on the left side of the board where the Type II PC-MIP cards are normally installed (on either P31/P32 or P41/P42)

Note The PMC slot is shared with one of the PC-MIP Type I slots. A factory build option determines whether the slot will support a PMC board or a PC-MIP Type I board (i.e. connectors P11/P12/P13 will be removed if the PMC board is installed on J11/J12/J14)

Two RJ45 connectors on the front panel provide the interface to 10/100Base-T Ethernet, and to a debug serial port.

The following list is of equipment that is appropriate for use in an MVME2100 system:

- PMCspan PCI expansion mezzanine module
- Type I or II PC-MIP cards
- 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)

MVME2100

The MVME2100 is a “state-of-the-art” and powerful Single Board Computer. It includes support circuitry such as SDRAM, PROM/Flash memory, and bridge to the VMEbus.

The unit’s PMC carrier architecture allows flexible configuration options and easy upgrades. The unit occupies a single VME module slot (except when the optional PMCspan expansion modules are also used); and as of the printing date of this manual, is currently available in the configurations shown below.

Table 1-1. MVME2100 SBC Models

Model	MPC	Memory	Handles
MVME2101-1	MPC8240 @200MHz	32MB ECC SDRAM 5 MB Flash Memory	Scanbe
MVME2101-3		32MB ECC SDRAM 5 MB Flash Memory	IEEE 1101 (Injector/Ejector)
MVME2112-1	MPC8240 @250MHz	64MB ECC SDRAM 9 MB Flash Memory	Scanbe
MVME2112-3		64MB ECC SDRAM 9 MB Flash Memory	IEEE 1101 (Injector/Ejector)

The MVME2100 interfaces to the VMEbus via the P1 and P2 connectors, which use the new 5-row 160-pin connectors as specified in the proposed VME64 Extension Standard. It also draws +5V, +12V, and -12V power from the VMEbus backplane through these two connectors. The +3.3V power, used for the PCI bridge chip and possibly for the PMC mezzanine and PC-MIP cards is derived onboard from the +5V power.

Support for one IEEE P1386.1 PCI mezzanine card is provided via three 64-pin SMT connectors. A front panel opening is provided on the board for the PMC slot.

In addition, there are 64 pins of I/O from the PMC slot and 46 pins of I/O from the Type I PC-MIP slot. There are also two RJ45 connectors on the front panel: one for the Ethernet 10BaseT/100BaseTX interface, and one for the async serial debug port. The front panel also includes reset and abort switches and status LEDs.

PMCspan Expansion Mezzanine

An optional PCI expansion mezzanine module or PMC carrier board, PMCspan, provides the capability of adding two additional PMCs. Two PMCspans can be stacked on an MVME2100, providing four additional PMC slots, for a total of six slots including the two onboard the MVME2100. The following table lists the PMCspan models that are available for use with the MVME2100.

Table 1-2. PMCspan Models

Expansion Module	Description
PMCSPAN-002	Primary PCI expansion mezzanine module. Allows two PMC modules for the MVME2100. Includes 32-bit PCI bridge.
PMCSPAN-010	Secondary PCI expansion mezzanine module. Allows two additional PMC modules for the MVME2100. Does not include 32-bit PCI bridge; requires a PMCSPAN-002.

PCI Mezzanine Cards (PMCs)

The PMC slot on the MVME2100 is IEEE P1386.1 compliant. P2 I/O-based PMCs that follow the PMC committee recommendation for PCI I/O when using the 5-row VME64 extension connector will be pin-out compatible with the MVME2100.

VME System Enclosure

Your MVME2100 board must be installed in a VME system chassis with both P1 and P2 backplane connections. It requires a single slot, except when PMCspan carrier boards are used. Allow one extra slot for each PMCspan.

System Console Terminal

In normal operation, connection of a debug console terminal is required only if you intend to use the MVME2100's debug firmware, PPCbug, interactively. An RJ45 connector is provided on the front panel of the board for this purpose.

Unpacking the MVME2100 Hardware

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.

Preparing the MVME2100 Hardware

To produce the desired configuration and ensure proper operation of the board, it may be necessary to perform certain modifications before and after installing it. The following paragraphs discuss the preparation of the MVME2100 hardware components prior to installing them into a chassis and connecting them.

MVME2100

The MVME2100 provides software control over most options; by setting bits in control registers. After installing the it in a system, you can modify its configuration. For additional information on the board's control registers, refer to the *MVME2100 Single Board Computer Programmer's Reference Guide*.

Some options, however, are not software-programmable. Such options are controlled through manual installation or removal of jumpers or additional interface modules on the MVME2100.

Manually configured jumpers on the MVME2100 include:

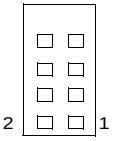
- Memory usage (soldered on or socketed Flash memory) (J9)
- System Control (J2)

Setting Flash Memory Bank A/Bank B Header (J9) (Pins 1 and 2)

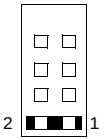
Bank A (soldered on memory) consists of four 16-bit devices that are populated with 8-Mbit flash devices (4 MB). Jumper header J9 (pins 1 and 2) provides selection between Bank A or Bank B configuration (for PPC Bug use only).

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

A jumper must be installed either between J9 pins 1 and 2 for Bank A factory configuration, or left off for Bank B configuration (factory default).



Bank A Configuration (Factory)



Bank B Configuration

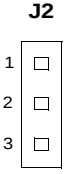
Setting System Controller Selection Header (J2)

The MVME2100 is factory-configured in automatic system controller mode (jumper is installed across pins 2 and 3 of header J2). This means that the MVME2100 determines if it is system controller at system power-up or reset by its position on the bus; if it is in slot 1 on the VME system, it configures itself as the system controller.

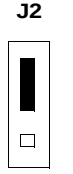
Remove the jumper from J2 if you intend to operate the MVME2100 as system controller in all cases.

Install the jumper across pins 1 and 2 if the MVME2100 will not to operate as system controller under any circumstances.

Automatic System Controller



System Controller



System Controller Disabled

PMCs

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

PMCspan

You will need to use an additional slot in the VME chassis for each PMCspan expansion module you plan to use. Before installing a PMCspan on the MVME2100, you must install the selected PMCs on the PMCspan. Refer to the PMCspan *PMC Adapter Carrier Module Installation and Use* manual for instructions.

System Console Terminal

Ensure that the appropriate jumper(s) are set in the correct position on the MVME2100 board. This is necessary when the PPCbug firmware is used. Connect the terminal via a cable to the RJ45 DEBUG connector. Refer to Appendix C for pin signal assignments. Set up the terminal as follows:

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

Installing the MVME2100 Hardware

The following section discuss installing PMCs onto the MVME2100, installing PMCspan modules onto the MVME2100, and installing the MVME2100 into a VME chassis, and connecting an optional system console terminal.

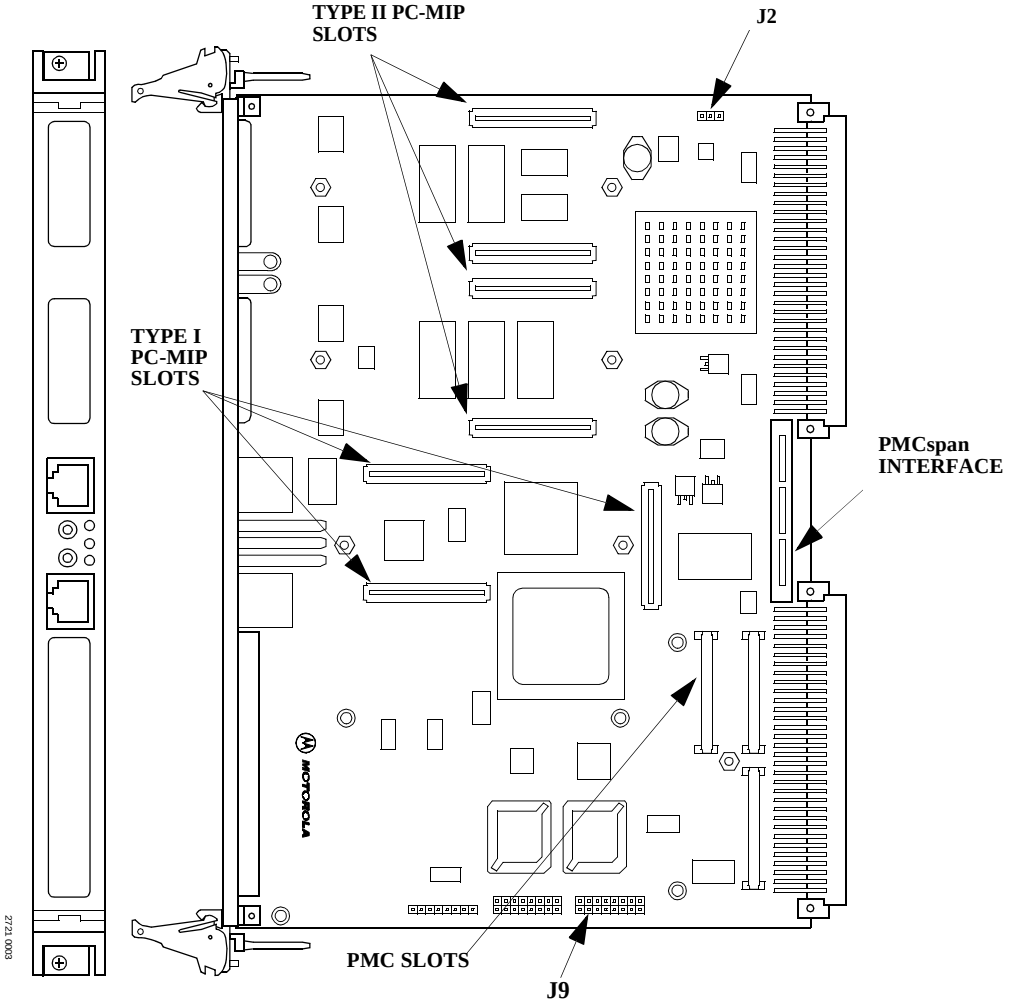


Figure 1-1. MVME2100 Layout

ESD Precautions

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 Electro-Static Discharge (ESD).

After removing the component from the system or its protective wrapper, place the component on a grounded and static-free surface (if a board, component side up). Do not slide the component over any surface.

If an ESD station is not available, you can avoid damage resulting from ESD by wearing an antistatic wrist strap (available at electronics stores) that is attached to an unpainted metal part of the system chassis.

PMCs & PC-MIPs

PMC modules and PC-MIP cards mount on top of the MVME2100 SBC. Perform the following steps to install a PMC module and/or a PC-MIP card on your MVME2100.

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

3. If the MVME2100 has already been installed in a VMEbus card slot, carefully remove it as shown in the figure below and place it with connectors P1 and P2 facing you.

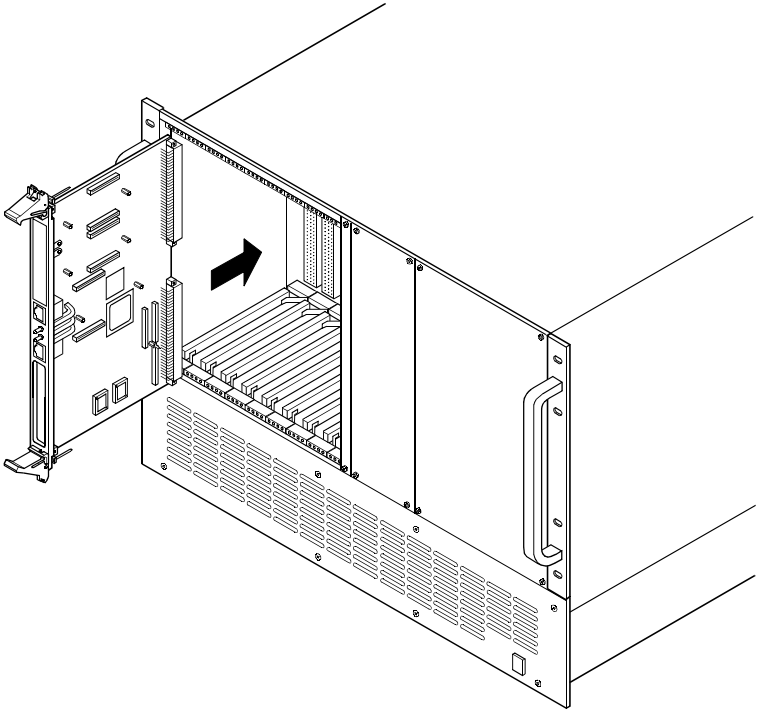


Figure 1-1. MVME2100 Installation and Removal From a VMEbus Chassis

4. Remove the filler plate(s) from the front panel of the MVME2100. If installing both PMC and PC-MIP(s), remove both filler plates.

5. If installing a PMC module, position the module's mating connectors on top of the MVME2100's mating connectors (J11/J12/J14),

   -or-

   If installing a PC-MIP, position the card's mating connectors on top of the MVME2100's Type II PC-MIP connectors (P41/P42 or P31/P32 for Type II; or P21/P22/P23 for Type I PC-MIP).

Note As a reminder, Type I PC-MIP cards without I/O can also be installed on the MVME2100's Type II PC-MIP connectors.

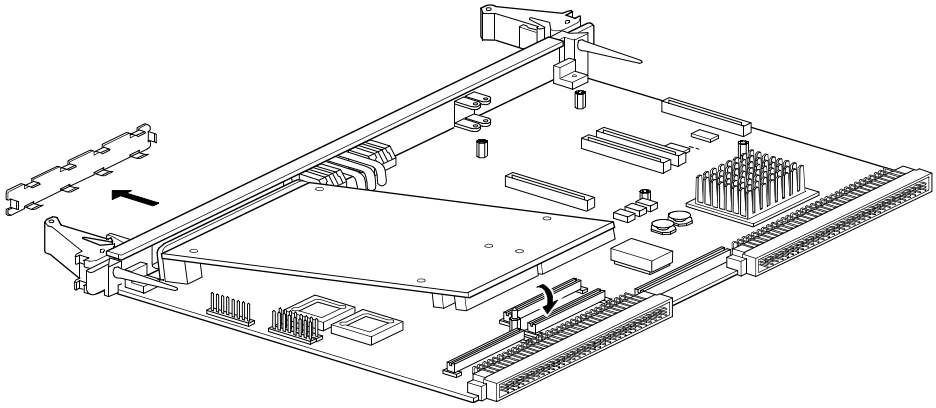


Figure 1-2. Typical Single-width PMC Module Placement on an MVME2100

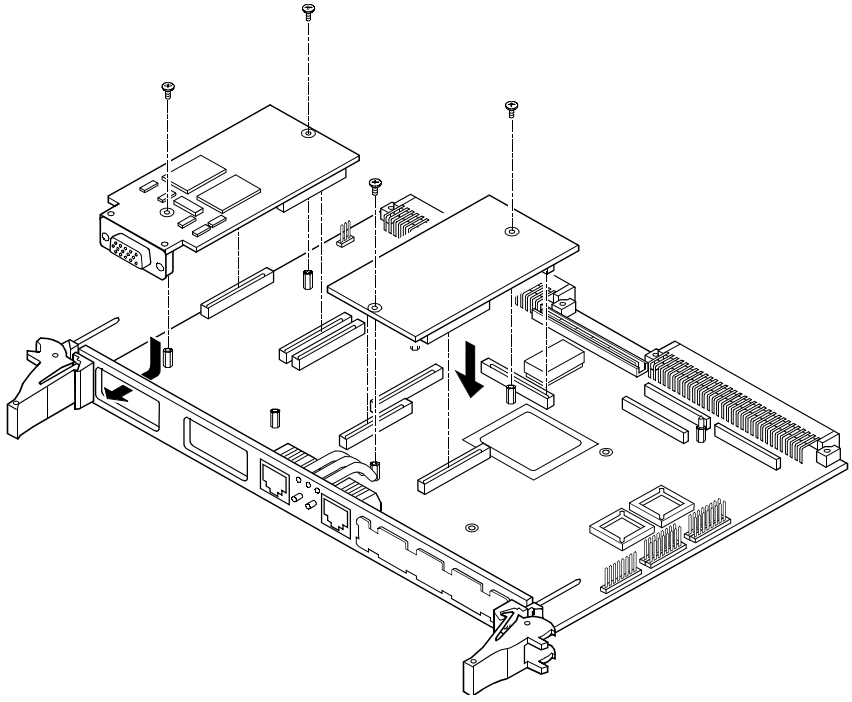


Figure 1-3. Typical Type II PC-MIP Placement on an MVME2100

6. Insert the appropriate number of phillips screws (typically 2) through the holes of the PMC module or PC-MIP card(s), into the mating standoffs on the MVME2100 and tighten the screws.

Primary PMCspan

To install a PMCspan-002 PCI expansion module on your MVME2100, refer to the figure on the next page perform the following steps:

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 PMCs on the PMCspan according to the instructions given in the PMCspan and PMC manuals.

1. Attach an ESD strap to your wrist. Attach the other end of the ESD strap to the chassis as a ground. The ESD strap must be secured to your wrist and to ground while you are performing the installation 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 card cage.

3. If the MVME2100 has already been installed in the chassis, carefully remove it from the VMEbus card slot and position it with connectors P1 and P2 facing you.

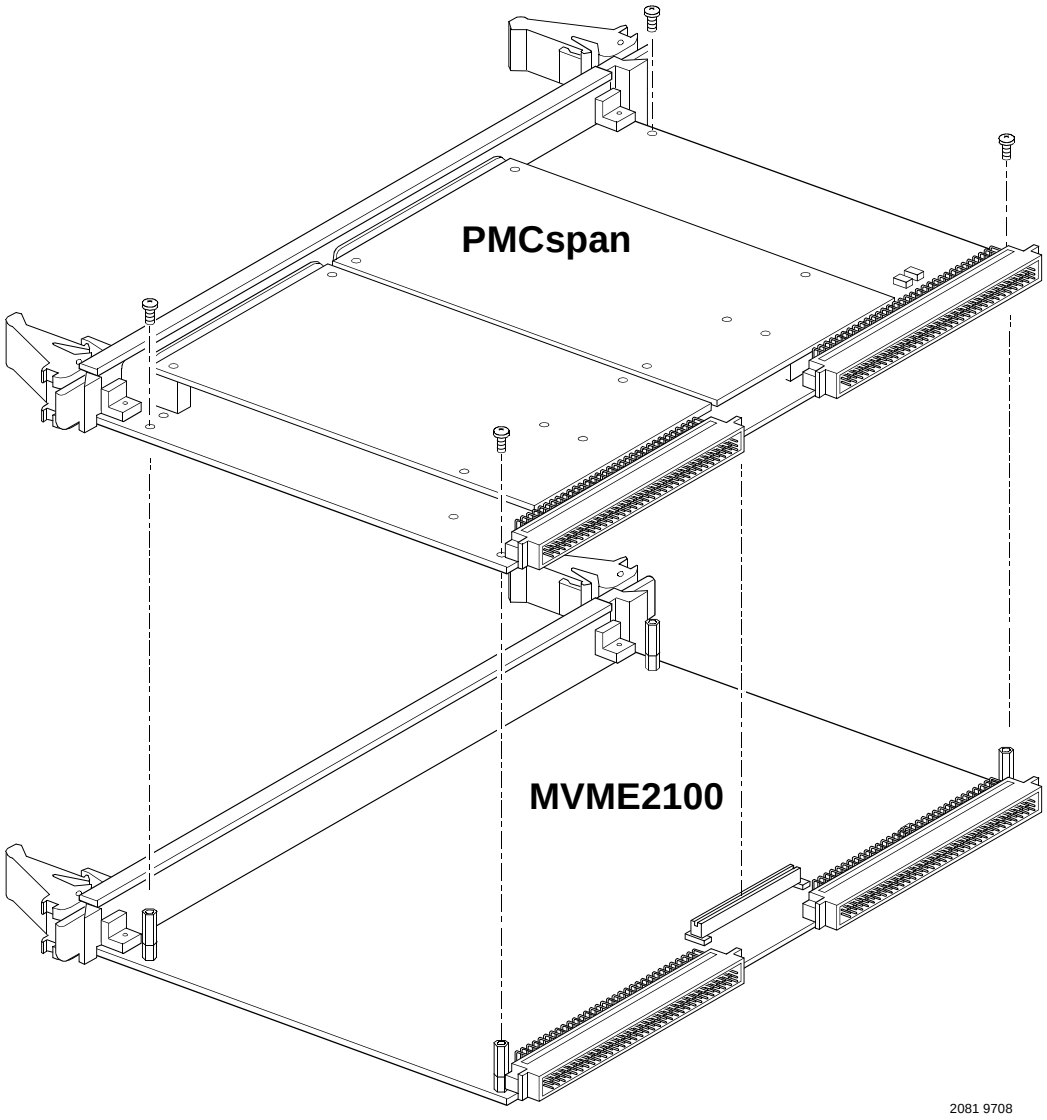


Figure 1-4. PMCspan-002 Installation on an MVME2100

4. Attach the four standoffs to the MVME2100 module. For each standoff:
   - Insert the threaded end into the standoff hole at each corner of the VME processor module.
   - Thread the locking nuts onto the standoff tips.
   - Tighten the nuts with a box-end wrench or a pair of needle nose pliers.
5. Place the PMCspan on top of the MVME2100. Align the mounting holes in each corner to the standoffs, and align PMCspan connector P4 with MVME2100 connector J4.
6. Gently press the PMCspan and MVME2100x together, making sure that P4 is fully seated in J4.
7. Insert the four short Phillips screws through the holes at the corners of the PMCspan and into the standoffs on the MVME2100. Tighten the screws.

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 MVME2100, refer to the figure on the next page and perform the following steps:

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

1. Attach an ESD strap to your wrist. Attach the other end of the ESD strap to the chassis as a ground. The ESD strap must be secured to your wrist and to ground while you are performing the installation 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 card cage.

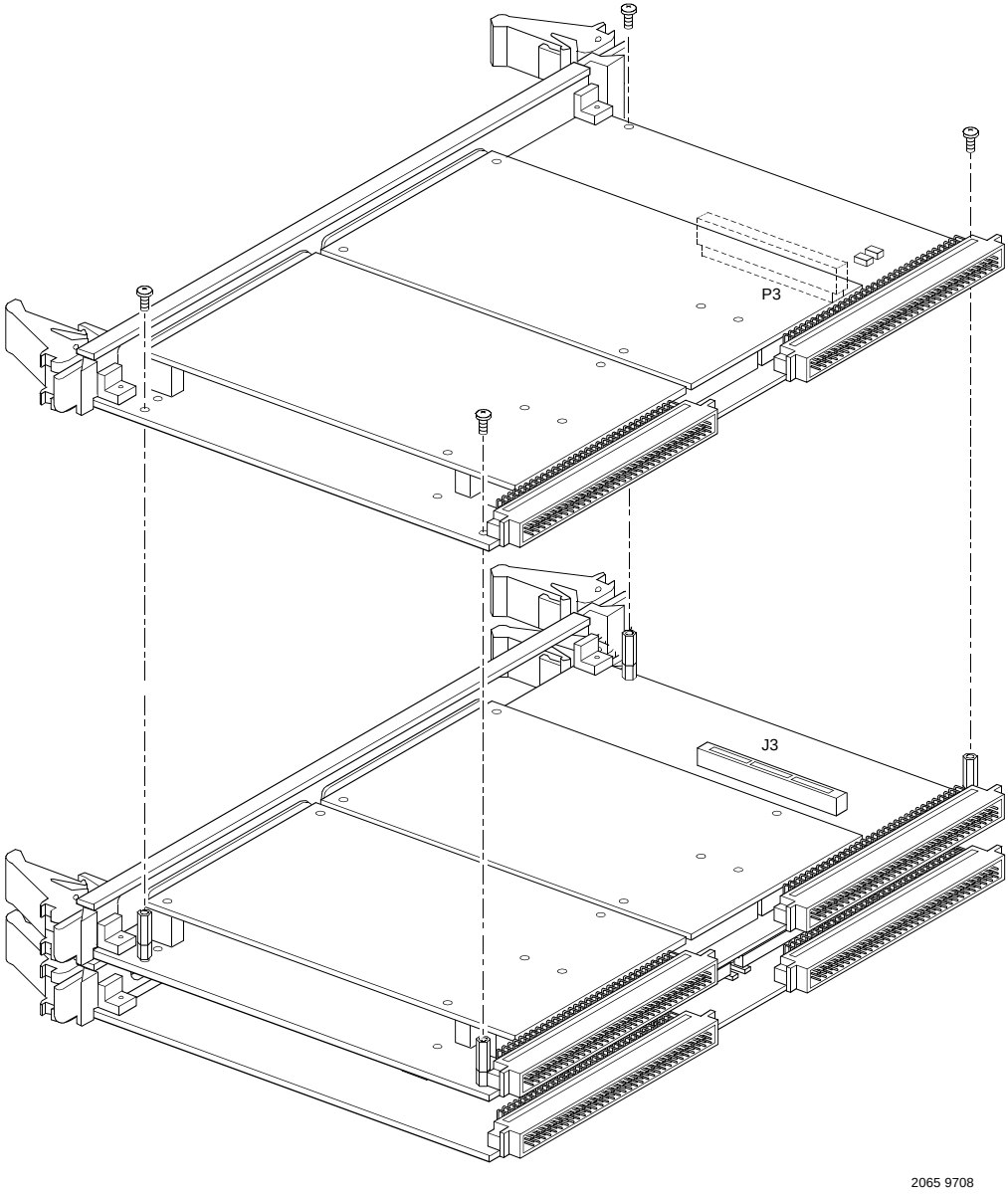


Figure 1-5. PMCspan-010 Installation on a PMCspan-002/MVME2100

Warning

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

Caution

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

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

3. If the Primary PMC Carrier Module/MVME2100 assembly is already installed in the VME chassis, carefully remove the two-board assembly from the VMEbus card slots and position it with the P1 and P2 connectors facing you.
4. Remove the four short Phillips screws from the standoffs in each corner of the primary PCI expansion module, PMCspan-002.
5. Attach the four standoffs to the PMCspan-002.
6. Place the PMCspan-010 on top of the PMCspan-002. Align the mounting holes in each corner to the standoffs, and align PMCspan-010 connector P3 with PMCspan-002 connector J3.
7. Gently press the two PMCspan modules together, making sure that P3 is fully seated in J3.
8. Insert the four short Phillips screws through the holes at the corners of PMCspan-010 and into the standoffs on the primary PMCspan-002. Tighten the screws.

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

MVME2100

Before installing the MVME2100 into your VME chassis, ensure that the jumpers on J2 and J9 configured properly. This procedure assumes that you have already installed the PMCspan(s) if desired, and any PMCs that you have selected.

Proceed as follows to install the MVME2100 in the VME chassis:

Warning

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

Caution

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

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 the chassis as a ground. The ESD strap must be secured to your wrist and to ground throughout the procedure.
2. Perform an operating system shutdown:
   a. Turn the AC or DC power off and remove the AC cord or DC power lines from the system.
   b. Remove chassis or system cover(s) as necessary for access to the VMEmodules.
3. Remove the filler panel from the card slot where you are going to install the MVME2100. If you have installed one or more PMCspan PCI expansion modules onto your MVME2100, you will need to remove filler panels from one additional card slot for each PMCspan, above the card slot for the MVME2100.
   – If you intend to use the MVME2100 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 MVME2100 as system controller, it can occupy any unused card slot.

4. Slide the MVME2100 (and PMCspans if used) into the selected card slot(s). Be sure the module or modules is/are seated properly in the P1 and P2 connectors on the backplane. Do not damage or bend connector pins.

5. Secure the MVME2100 (and PMCspans if used) in the chassis with the screws provided, making good contact with the transverse mounting rails to minimize RF emissions.

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

6. On the chassis backplane, remove the INTERRUPT ACKNOWLEDGE (IACK) and BUS GRANT (BG) jumpers from the header for the card slot occupied by the MVME2100

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 MVME2100.

Note In normal operation the host CPU controls MVME2100 operation via the VMEbus Universe registers.

8. Replace the chassis or system cover(s), cable peripherals to the panel connectors as appropriate, reconnect the system to the AC or DC power source, and turn the equipment power on.

9. The MVME2100 green **RUN** LED indicates activity as a set of confidence tests is run, and the debugger prompt `PPC1-Bug>` appears.

Installation Considerations

The MVME2100 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 MVME2100 may not function properly without its main board connected to VMEbus backplane connectors P1 and P2.

Whether the MVME2100 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 MVME2100 tries to access off-board resources in a nonexistent location and is not system controller, and if the system does not have a global bus time-out, the MVME2100 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: when the MVME2100 is not the system controller and there is no global bus time-out elsewhere in the system.

Multiple MVME2100 boards may be installed in a single VME chassis. Each must have a unique Universe address. Other MPUs on the VMEbus can interrupt, disable, communicate with, and determine the operational status of the processor(s). One register of the Universe set includes four bits that function as location monitors to allow one MVME2100 processor to broadcast a signal to any other MVME2100 processors. All eight registers are accessible from any local processor as well as from the VMEbus.

Operating Instructions 2

Introduction

This chapter provides operating instructions for the MVME2100 Single Board Computer. This includes information about powering up the system, and functionality of the switches, status indicators, and I/O ports on the front panels of the board.

Applying Power

After you have verified that all necessary hardware preparation has been done, that all connections have been made correctly, and that the installation is complete, you can power up the system. The MPU, hardware, and firmware initialization process is performed by the PPCbug firmware power-up or system reset. The firmware initializes the devices on the MVME2100 module in preparation for booting the operating system.

The firmware is shipped from the factory with an appropriate set of defaults. In most cases there is no need to modify the firmware configuration before you boot the operating system. Refer to Chapter 5 for further information about modifying defaults.

The following flowchart shows the basic initialization process that takes place during MVME2100 system start-ups.

For further information on PPCbug, refer to the following:

- Chapter 5, *Modifying the Environment*
- Appendix A, *Related Documentation*
- Appendix D, *Troubleshooting*

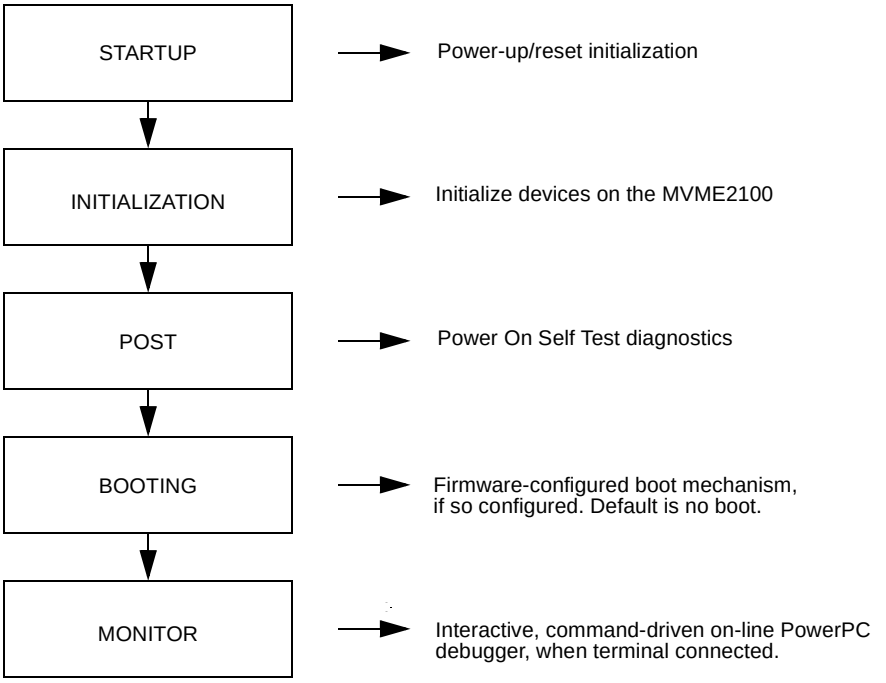


Figure 2-1. SYSTEM BOOT-UP SEQUENCE

MVME2100

The front panel of the MVME2100 SBC is shown on a following page.

Switches

There are two switches (**ABT** and **RST**) and three LED (light-emitting diode) status indicators (**BFL**, **SYS**, **RUN**) located on the MVME2100 front panel.

ABT (S1)

When activated by software, the Abort switch, **ABT**, can generate an interrupt signal from the base board to the processor at a user-programmable level. The interrupt is normally used to abort program execution and return control to the debugger firmware located in the MVME2100 Flash memory.

The interrupt signal reaches the processor module via serial interrupt 14. The signal is also available from the general purpose I/O port, which allows software to poll the Abort switch after receiving serial interrupt 14 and verify that it has been pressed.

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

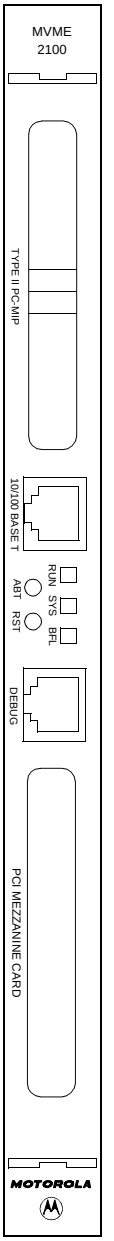
RST (S2)

The Reset switch, **RST**, resets all onboard devices and causes HRESET* to be asserted in the MPC8240. It also drives a SYSRESET* signal if the MVME2100 VME processor module is the system controller.

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

Status Indicators

There6 are Three LED (light-emitting diode) status indicators located on the MVME2100 front panel.: **BFL, SYS**, and **RUN**.



BFL (DS1)

The *yellow* **BFL** LED indicates board failure; this indicator is illuminated during a hard reset or the assertion of MCP and can only be turned off by writing to a system control
register.

SYS (DS2)

The *green* **SYS** LED indicates CPU activity; when illuminated, this indicator signifies that the MVME2100 is functioning as the VMEbus System Controller.

RUN (DS3)

The top *green* **RUN** LED illuminates when either ROM/FLASH, SDRAM, or PCI accesses are occurring.

10/100 BASET Port

The RJ45 port on the front panel of the MVME2100 labeled **10/100 BASE T** supplies the Ethernet LAN 10BaseT/100Base TX interface.

DEBUG Port

The RJ45 port labeled **DEBUG** on the front panel of the MVME2100 supplies the MVME2100 serial communications interface, implemented via a TL16C550 Universal Asynchronous Receiver/Transmitter (UART) controller chip manufactured by Texas Instruments. It is asynchronous only. For configuration information, refer to the section *Asynchronous Serial Port* found in Chapter 3.

The **DEBUG** port may be used for connecting a terminal to the MVME2100 to serve as the firmware console for the factory installed debugger, PPCbug. The port is configured as follows:

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

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

Jumper Settings

The following table describes the MVME2100 jumper configuration.

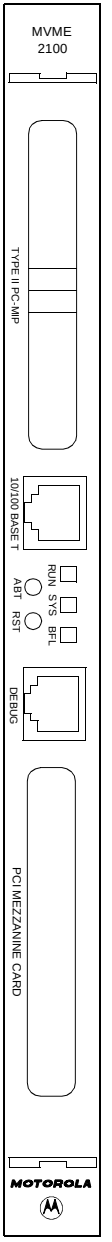
Table 2-1. Jumper Switches and Settings

Jumper	Description	Setting	Default
J1	Factory Test Header	Reserved	N/A
J2	VMEbus System Controller Functionality Select	**Pins 1 & 2 Shorted:** Disables the System Controller Function **Pins 2 & 3 Shorted:** Enables Auto-Sence Function **No Shunt on Pins Forces the System Controller On**	No Shunt on Pins
J6	Factory Test Header (I2C Signals)	Reserved	N/A
J7	Programming Header (ISPLI)	Reserved	N/A
J8	RiscWatch JTAG	Reserved	N/A
J9	Software Readable Header (Support for Processor Emulation) Pins 1 & 2 For PPCbug Use Only.	**Pins 1 & 2:** **ON**=Use Soldered On Flash Memory Devices **OFF**= Use Memory in Sockets U1, U8, U13, U15 Note: PPCbug uses Bit 0 while booting to determine whether to continue executing from socketed Flash or jump to soldered-on Flash. Pins 3 - 16 are User Definable. Pins 3/4=Bit 1 / Pins 5/6=Bit 2 / Pins 7/8= Bit 3 Pins 9/10=Bit 4 / Pins 11/12=Bit 0 Pins 13/14=Bit 6 / Pins 15/16=Bit 7	Pins 1 & 2 OFF

2

I/O EXPANSION CARDS

Two openings are located on the front panel of the MVME2100 to provide I/O expansion by allowing access to a PCI Mezzanine Card (PMC) or two Type II PC-MIP cards. Refer to Appendix C for additional information on pin assignments.



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

PCI MEZZANINE CARD

The right-most (lower) opening labeled **PCI MEZZANINE CARD** on the MVME2100 front panel provides front panel I/O access to a PMC that is connected to the 64-pin connectors J11, J12, and J14 on the MVME2100. Connector J14 allows rear panel P2 I/O.

TYPE II PC-MIP

The left-most opening labeled **TYPE II PC-MIP** on the MVME22100 front panel provides front panel I/O access to two Type II PC-MIP cards connected to the 64-pin connectors P31, P32, P41, and P42 on the MVME2100.

PMCspan

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

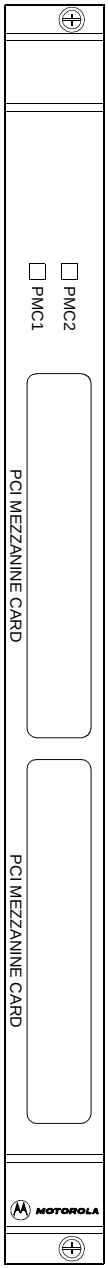
There are two PMC slots, labeled PCI MEZZANINE CARD, which support either two single-wide PMCs or one double-wide PMC.

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

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

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

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



Functional Description 3

Introduction

This chapter provides additional product information along with a general functional description for the MVME2100 Single Board Computer.

The MVME2100 is a VME based single-slot Single Board Computer based on the MPC8240™ Integrated Processor.

Key features of the MVME2100 include either one 32-bit PMC expansion slot, one Type I and two Type II PC-MIP expansion slots or two Type I and two Type II PC-MIP expansion slots, 32 or 64MB of synchronous DRAM memory, 1MB Boot FLASH ROM, 4 or 8MB expansion FLASH ROM, one 10/100Base-TX Ethernet port, and one front panel accessible asynchronous serial port.

The following table lists the key features of the MVME2100.

Table 3-1. MVME2100 Features

Processors	• MPC8240™ • Bus Clock Frequencies of 66.67/83.33 MHz
Flash Memory	• Sockets for 1MB (8-bit) plus 4MB or 8MB (64-bit) of expansion Flash memory
System Memory	• 32 or 64MB Synchronous DRAM
LAN	• DEC21143 10/100Base-TX Ethernet • Controller • LXT970 Fast Ethernet Transceiver
Interrupt Controller	• PowerPC™ Embedded Programmable Interrupt Controller (EPIC)
DMA	• 2 Independent DMA Channels
Timers	• Four Independent Timers

Table 3-1. MVME2100 Features (Continued)

I^2C	• Integrated I^2C port with full master support
I_2O	• I_2O compliant messaging Interface
NVRAM	• 8KB (MK48T59Y)
RTC & Watchdog Timer	• MK48T59 device
Serial Interface	• One 16550-compatible async serial port
PCI Mezzanine Card$_{(1)}$	• One 32-bit PMC slot • Front panel I/O • MVME2300 compatible P2 I/O
PC-MIP$_{(1)}$	• Two 32-bit Type I PC-MIP slots (MVME2300 compatible P2 I/O) • Two 32-bit Type II PC-MIP slots (front panel I/O)
PCI Expansion	• Genesis II PCI expansion compatibility
Miscellaneous	• RESET switch • ABORT switch • Front panel status indicators
Form Factor	• Standard 6U VME

Note The PMC slot is shared with one of the PC-MIP Type I slots. A factory build option determines whether the slot will support a PMC board or a PC-MIP Type I board.

The block diagram in Figure 3-1 illustrates the architecture of the MVME2100 Single Board Computer.

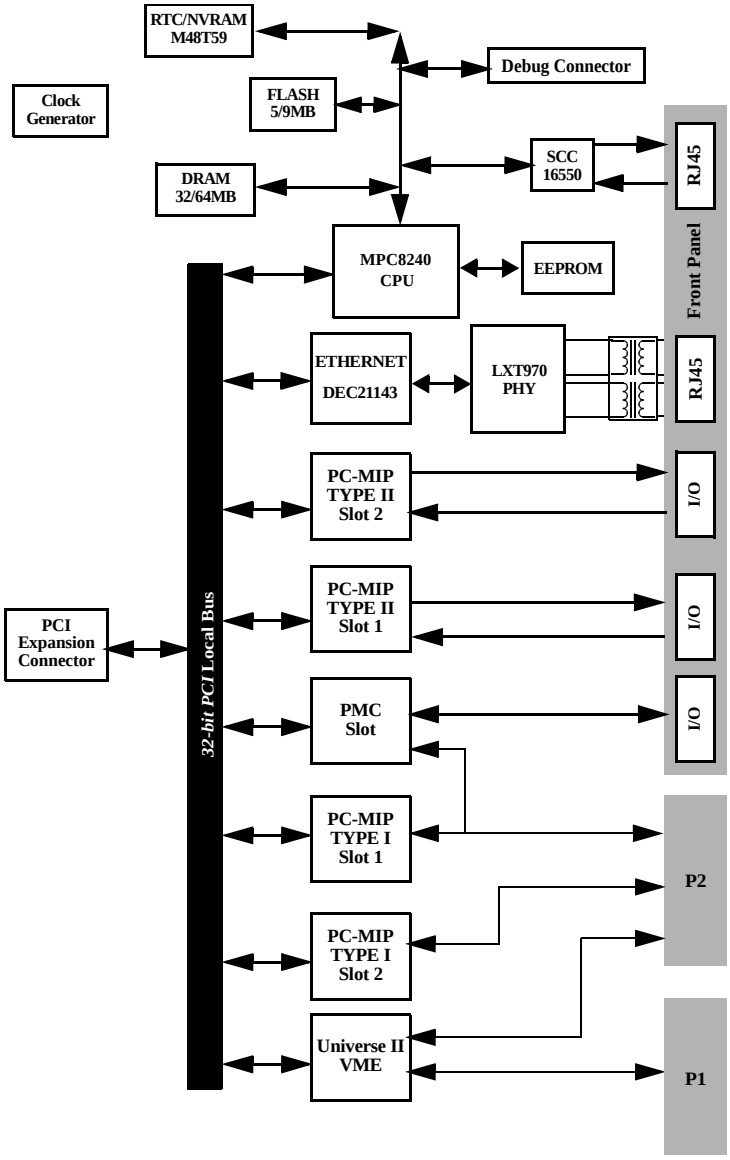


Figure 3-1. MVME2100 Block Diagram

Functional Description

The MVME2100 is a VMEbus-based single-slot Single Board Computer based on the MPC8240 processor.

Processor

The MVME2100 is designed to support the MPC8240 processor in a 352 pin TBGA package. It is also designed to support memory bus speeds of 50, 60, 66.67, and 83.33 MHZ as a board population option.

PCI Host Bridge/Memory Controller

The MPC8240 contains an integrated PCI Host Bridge and Memory Controller which provides the bridge function between the internal MPC60x bus and the external PCI Local Bus.

The processor supports a 32-bit PCI interface that is compliant with the PCI Local Bus Specification, Revision 2.1. Additional features of the processor include:

- PReP or CHRP compatible memory maps
- DRAM control/refresh
- 3.3/5.0V compatible I/O
- power management support
- Boot ROM interface.

PCI Bus Arbitration

PCI arbitration for the MVME2100 board is provided by the integrated PCI arbiter internal to the processor in conjunction with an external sub-arbiter. The processor provides support for itself and up to five external PCI masters.

Since the MVME2100 could have as many as seven potential PCI masters in addition to the processor, an onboard sub-arbiter is provided. The sub-arbiter is designed to multiplex the common PMC/PC-MIP slot and three dedicated PC-MIP slots onto one set of the processor's PCI bus request/grant pins.

Interrupt Controller

The MVME2100 uses the Embedded Programmable Interrupt Controller (EPIC) integrated into the processor to manage locally generated interrupts. The interrupt controller will operate in the serial interrupt mode.

Currently defined external interrupting devices include:

- DEC21143 Ethernet Controller
- One optional PMC/PC-MIP Type I Expansion Slot
- One dedicated PC-MIP Type I Expansion Slot
- Two dedicated PC-MIP Type II Expansion Slots
- Universe II VME-PCI Bridge
- 16550 UART
- Watchdog Timer
- Front panel Abort switch
- Four PCI Expansion Interrupts (INTA* - INTD*)

For additional information on the operation of the processor's EPIC, refer to the *MPC8240 User's manual.*

Two-Wire Serial Interface

A two-wire serial interface for the MVME2100 is provided by an I^2C compatible serial controller integrated into the processor's peripheral device. The processor's serial controller is used by the system software to read the contents of the configuration EEPROM contained on the board.

The MVME2100 also contains a four pin header (J6) providing power, ground, serial clock, and serial data signals for testing purposes.

I_2O Message Unit

I_2O compliant messaging for the MVME2100 is provided by an I_2O compliant messaging unit integrated into the processor's peripheral device. The processor's message unit can operate with either generic messages and door bell registers, or as an I_2O compliant interface.

Direct Memory Access (DMA)

The MVME2100 provides DMA capability through a two channel DMA controller integrated into the processor's peripheral device. Each DMA channel is capable of performing local memory to local memory, PCI memory to local memory, local memory to PCI memory and PCI memory to PCI memory data transfers.

Both DMA channels can be accessed by the local CPU as well as external PCI bus masters and support unaligned transfers, data chaining, and scatter gather.

Timers

Timing functions for the MVME2100 are provided by four independent 31 bit timers integrated into the processor. The four timers are clocked at 1/8 of the processor clock rate. Each timer contains four registers enabling the system software to set the count values, enable or disable the timer, enable or disable interrupt generation, set the interrupt priority level, and to generate an interrupt vector.

System Clock Generator

The system clock generator function shall generate and distribute all of the clocks required for normal system operation. The clock generator for the processor, memory, and PCI devices should be designed in such a manner as to maintain the strict edge to edge jitter and low clock to clock skew required by these devices.

Additional clocks that may be required should be generated near the individual devices requiring clocks to minimize onboard trace lengths.

Flash Memory

The MVME2100 contains two banks of Flash memory accessed via the integrated memory controller contained within the processor. Bank B consists of two 32-pin PLCC sockets which can be populated with up to 1024KB of FLASH memory, resides at address 0xFFF00000, and is restricted to 8 bits in width.

Bank A may be populated with four 512Kx16 FLASH devices to obtain 4MB of 64-bit wide expansion FLASH memory or four 1Mx16 FLASH devices to obtain 8MB of 64-bit wide FLASH memory. The expansion FLASH memory starts at address 0xFF000000.

System Memory

System memory for the MVME2100 is provided by 2 banks of synchronous DRAM. Each bank consists of five 4Mx16 SDRAM devices providing a 32MB bank organized in a 4Mx72 configuration. This allows memory configurations of 32 or 64MB that can be supported by the board.

During system initialization, the firmware will determine the presence, and configuration of each memory bank installed by reading the contents of the serial presence detect ROM located on the board. The system firmware will then initialize the MPC8240 Memory Controller for proper operation based on the contents of the serial presence detection ROM.

Note The processor does not provide support for an external L2 cache. Therefore, there will not be L2 cache present on the MVME2100.

PCI Local Bus

In addition to the processor, there may be as many as 7 additional PCI devices located on the local PCI bus. The potential PCI devices on the board are: one optional PCM/PC-MIP Type I board, one dedicated PC-MIP Type I board, two PC-MIP Type II boards, one DEC21143 Ethernet Controller, one PCI-VMEbus bridge, and one DEC21150 PCI-PCI Bridge.

Ethernet Interface

The MVME2100 provides a 10/100Base-TX Ethernet transceiver interface using a DEC21143 Ethernet Controller and a LXT970 Fast Ethernet Transceiver. The Ethernet interface is accessed via a industry standard front panel mounted RJ45 connector.

The DEC21143 will be assigned an Ethernet Station Address. The address will be $08003E2XXXXX where XXXXX is the unique number assigned to the Ethernet Controller.

The Ethernet Station Address will be displayed on a label affixed to the board. In addition, the Ethernet address will be stored in the configuration ROM interfaced to the Ethernet Controller.

Asynchronous Serial Port

The MVME2100 uses a TL16C550 or compatible Universal Asynchronous Receiver/Transmitter (UART) with a 1.8432Mhz input clock to provide an asynchronous serial interface. EIA232 drivers and receivers reside onboard and are routed to an industry standard RJ45 connector accessible from the front panel.

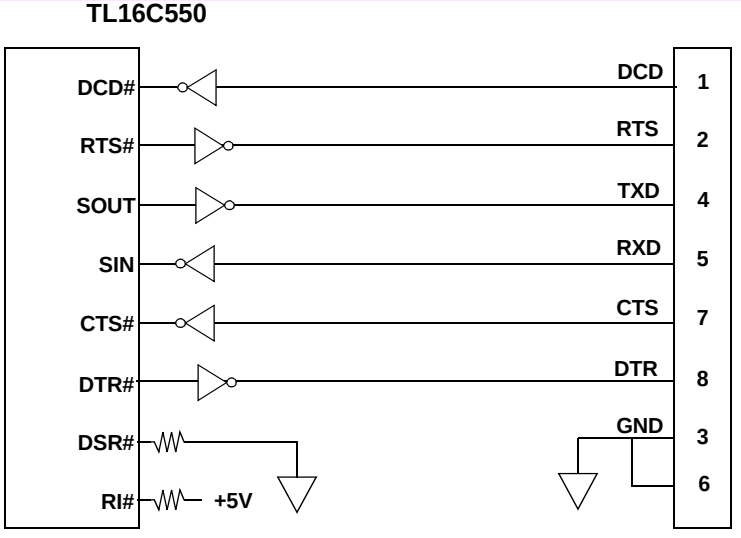


Figure 3-2. Asynchronous Serial Port Connections

VMEbus Interface

The VMEbus interface for the MVME2100 is provided by the Universe II ASIC. Refer to the *Universe II User's Manual* for additional information.

PCI Mezzanine Card Slot

The MVME2100 can support one PMC slot as a factory build option. When the board is configured for the PMC option, one PC-MIP Type I slot is unusable. With this option three EIA E700 AAAB connectors are placed on the board to interface to a single 32-bit IEEE P1386.1 PMC to add any desirable function.

PMC slot support specifications are follows:

- Mezzanine Type: PMC = PCI Mezzanine Card
- Mezzanine Size: S1B = Single width & standard depth (75mm x 150mm) with front panel
- PMC Connectors: J11, J12, J14 (32-Bit PCI with front panel and user defined I/O)
- Signalling Voltage: V_{io} = 5.0V

PC-MIP Type I Mezzanine Card Slots

The MVME2100 provides one dedicated Type I PC-MIP slot and a second optional Type I PC-MIP card slot may be added if a PMC slot is not required.

When the board is configured for two Type I PC-MIP card slots, it will accommodate either one double wide Type I PC-MIP board or two single width Type I PC-MIP boards.

Note In this configuration the PMC card slot will be unusable.

User defined I/O, as defined in the PC-MIP specification, is provided for both Type I PC-MIP card slots via the VMEbus P2 connector.

PC-MIP Type II Mezzanine Card Slots

The MVME2100 provides two Type II PC-MIP slots using front panel I/O. These slots will accommodate either one double wide Type II PC-MIP board or two single width Type II PC-MIP boards.

Note User defined I/O using P3 of the Type II PC-MIP boards is not supported by the board.

PCI/PMC Expansion Capability

The MVME2100 provides additional PCI capability through the use of a 114 pin Mictor connector that is compatible with the Genesis II series of VMEbus processor boards. By using existing PMC-Span carrier boards, up to four additional PMC boards can be used.

Real-Time Clock & NVRAM

The SGS-Thomson M48T59 is used by the MVME2100 to provide 8KB of non-volatile static RAM and a real-time clock. It consists of two parts:

- A 28-pin 330mil SO device which contains the RTC, the oscillator, 8KB of SRAM, and gold-plated sockets for the SNAPHAT battery.
- A SNAPHAT battery that houses the crystal and the battery.

Note Refer to the MK48T59 Data Sheets for programming information.

The SNAPHAT battery package is to be mounted on top of the SO MT48T59 device after the completion of the surface mount process. The battery housing is keyed to prevent reverse insertion.

PPCBug Firmware 4

PPCBug Overview

The PPCBug firmware is the layer of software just above the hardware. The firmware provides the proper initialization for the devices on the MVME2100 module upon power-up or reset.

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

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

PPCBug Basics

The PowerPC debug firmware (known as the "PPCBug") is a powerful evaluation and debugging tool for systems built around the Motorola PowerPC microcomputers. Facilities are available for loading and executing user programs under complete operator control for system evaluation. 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 power-up 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*. It is hereafter referred to as "the debugger" or "PPCBug".
- A command-driven *diagnostics package* for the MVME2100 hardware, hereafter referred to as "the diagnostics." The diagnostics package is described in the *PPCBug Diagnostics Manual.*
- 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 `PPC5-Bug>` is displayed and you have all of the debugger commands at your disposal.
- If you are in the diagnostic directory, the diagnostic prompt `PPC5-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.

Memory Requirements

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 ($04000000) of read/write memory will place the PPCBug memory page at locations $03F40000 to $03FFFFFF.

PPCBug Implementation

PPCBug is written largely in the C programming language, providing benefits of portability and maintainability. Where necessary, assembly language has been used in the form of separately compiled program modules containing only assembler code.

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.

MPU, Hardware, and Firmware Initialization

The debugger performs the MPU, hardware, and firmware initialization process. This process occurs each time the MVME2100 is reset or powered up. The steps below are a high-level outline; not all of the detailed steps are listed.

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

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

27. Calculates and displays the MPU clock speed, verifies that the MPU clock speed matches the configuration data, and displays a warning message if the verification fails.
28. Displays the BUS clock speed, verifies that the BUS clock speed matches the configuration data, and displays a warning message if the verification fails.
29. Probes PCI bus for supported network devices.
30. Probes PCI bus for supported mass storage devices.
31. Initializes the memory/IO addresses for the supported PCI bus devices.
32. Executes Self-Test, if so configured. (Default is no Self-Test.)
33. Extinguishes the board fail LED, if Self-Test passed, and outputs any warning messages.
34. Executes boot program, if so configured. (Default is no boot.)
35. Executes the debugger monitor (i.e., issues the `PPC5-Bug>` prompt).

Using PPCBug

PPCBug is command-driven; it performs its various operations in response to commands that you enter at the keyboard. When the `PPC5-Bug` prompt appears on the screen, the debugger is ready to accept debugger commands. When the `PPC5-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*.

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*, Chapter 5). For more about this, refer to the **GD**, **GO**, and **GT** command descriptions in the *PPCBug Firmware Package User's Manual*, Chapter 3.

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.

Debugger Commands

The individual debugger commands are listed in the following table. The commands are described in detail in the *PPCBug Firmware Package User's Manual*, Chapter 3.

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 4-1. 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
DS	Disassembler
DU	Dump S-Records
ECHO	Echo String
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)

Table 4-1. Debugger Commands (Continued)

Command	Description
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
M	“Alias” for “MM” Command
MA	Macro Define/Display
MAE	Macro Edit
MAL	Enable Macro Expansion Listing
MAR	Macro Load

Table 4-1. Debugger Commands (Continued)

Command	Description
MAW	Macro Save
MD	Memory Display
MDS	Memory Display
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
NOPF	Port Detach
NORB	No ROM Boot
NOSYM	Detach Symbol Table
NPING	Network Ping

Table 4-1. Debugger Commands (Continued)

Command	Description
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
SYM	Symbol Table Attach
SYMS	Symbol Table Display/Search
T	Trace
TA	Terminal Attach
TIME	Display Time and Date
TM	Transparent Mode
TT	Trace to Temporary Breakpoint
VE	Verify S-Records Against Memory
VER	Revision/Version Display
WL	Write Loop

Caution

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

Diagnostic Tests

The PPCBug hardware diagnostics are intended for testing and troubleshooting the MVME2100.

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 `PPC5-Bug>` is displayed, and all of the debugger commands are available. Diagnostics commands cannot be entered at the `PPC5-Bug>` prompt.

If you are in the diagnostic directory, the diagnostic prompt `PPC5-Diag>` is displayed, and all of the debugger and diagnostic commands are available.

PPCBug's diagnostic test groups are listed in the Table 5-2. Note that not all tests are performed on the MVME2100. Using the **HE** command, you can list the diagnostic routines available in each test group. Refer to the *PPCBug Diagnostics Manual* for complete descriptions of the diagnostic routines and instructions on how to invoke them.

Table 4-2. 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 You may enter command names in either uppercase or lowercase.

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.

Test Sets marked with an asterisk (*) are not available on the MVME2100.

Modifying the Environment

5

Overview

You can use the factory-installed debug monitor, PPCBug, to modify certain parameters contained in the MVME2100 Non-Volatile RAM (NVRAM), also known as Battery Backed-up RAM (BBRAM).

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

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

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

CNFG - Configure Board Information Block

Use this command to display and configure the Board Information Block, which is resident within the NVRAM. This data block contains various elements detailing specific operational parameters of the MVME2100. The structure for the board is shown in the following example:

```
Board (PWA) Serial Number     = MOT00xxxxxxx
Board Identifier              = MVME2100
Artwork (PWA) Identifier      = 01-W3403FxxC
MPU Clock Speed               = 250
Bus Clock Speed               = 083
Ethernet Address              = 08003E2A0A57
Primary SCSI Identifier       = 07
System Serial Number          = nnnnnnn
System Identifier             = Motorola MVME2101
License Identifier            = nnnnnnnn
```

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

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

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

ENV - Set Environment

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

Refer to the *PPCBug Firmware Package User's Manual* for a description of the use of **ENV**. Additional information on registers in the Universe ASIC that affect these parameters is contained in your *MVME2100 Programmer's Reference Guide*.

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

Configuring the PPCBug Parameters

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

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

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

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

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. Rather, 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 MVME2100 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 (Global Environment Variable) boot takes priority over all other boots, including Autoboot and Network Boot.

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

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

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

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

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

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

Y The Autoboot function is enabled.

N The Autoboot function is disabled. (Default)

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

Y Autoboot is attempted at power-up reset only.

N Autoboot is attempted at any reset. (Default)

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

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

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


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

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

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

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

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

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

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

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

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

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

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

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


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

Y The ROMboot function is enabled.

N The ROMboot function is disabled. (Default)

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

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

N ROMboot is attempted at any reset.

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

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

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

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

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

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

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

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

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

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

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

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

Y	NETboot is attempted at power-up reset only.
N	NETboot is attempted at any reset. (Default)

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

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

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

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

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

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

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

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

Caution

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

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


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

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

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

The default Starting Address is $00000000.

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

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

```
DRAM Speed in NANO Seconds = 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 (above) also applies to enabling ECC for DRAM.

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

O	L2 Cache parity is enabled upon detection. (Default)
A	L2 Cache parity is always enabled.
N	L2 Cache parity is never enabled.

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

Initializes the PIRQx (PCI Interrupts) route control registers in the IBC (PCI/ISA bus bridge controller). The **ENV** parameter is a 32-bit value that is divided by 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. Should the debugger fail to come up to a prompt, the last code displayed will indicate how far the initialization sequence had progressed before stalling. The codes are enabled by an **ENV** parameter:

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

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

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

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

Configuring the VMEbus Interface

ENV asks the following series of questions to set up the VMEbus interface for the MVME2100. To perform this configuration, you should have a working knowledge of the Universe ASIC as described in your *MVME2100 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 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 will 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.

Related Documentation A

Motorola Computer Group Documents

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

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

Table A-1. Motorola Computer Group Documents

Document Title	Motorola Publication Number
MVME2100 Single Board Computer Programmer's Reference Manual	MVME2100A/PG
PPCBug User's Manual	PPCBUGA/UM
PPCBug Diagnostics Manual	PPCDIAA/UM
MPC8240 Integrated Processor Training Manual	N/A*
PMCspan PMC Adapter Carrier Module Installation and Use	PMCSPANA/IH

* Courseware provided in training class. Contact Motorola Computer Group Training Department for Manual and course availability.

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 A-2. Manufacturers' Documents

Document Title	Publication Number
MPC8240 Integrated Processor User's Manual Motorola Literature Distribution Center Telephone: (800) 441-2447 or (303) 675-2140	MPC8240UM/D
PowerPC 603 RISC Microprocessor User's Manual Motorola Literature Distribution Center Telephone: (800) 441-2447 or (303) 675-2140	MPC603EUM/AD
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
TL16C550C Universal Asynchronous Receiver/Transmitter Texas Instruments Dallas, Texas	SLLS177C
AM29LV800B 8Megabit CMOS 3.0 Volt-only Boot Sector Flash Memory Advanced Micro Devices Inc. P.O. Box 3453 Sunnyvale, California 94088-3453 (408) 732-2400	21490

Table A-2. Manufacturers' Documents (Continued) (Continued)

Document Title	Publication Number
AM29LV160B 16Megabit CMOS 3.0 Volt-only Boot Sector Flash Memory Advanced Micro Devices Inc. P.O. Box 3453 Sunnyvale, California 94088-3453 (408) 732-2400	21358
LXT970 Fast Ethernet Transceiver Level One Communications, Inc. 9760 Goethe Road Sacramento, CA 95827	N/A
AT24C01A/02/04/08/16 2-Wire Serial CMOS E^2PROM Atmel Corporation 2325 Orchard Parkway San Jose, CA 95131 (408) 441-0311	0180C
M48T59Y CMOS 8Kx8 Timekeeper SRAM SGS Thomson Microelectronics 1000 East Bell Road Phoenix, AZ 85022	M48T59Y
DIGITAL Semiconductor 21143 PCI/CardBus 10/100-Mb/s Ethernet LAN Controller, Hardware Reference Manual Digital Equipment Corp. Maynard, Massachusetts	EC-QWC4E-TE
DIGITAL Semiconductor 21143 PCI/CardBus 10/100-Mb/s Ethernet LAN Controller Data Sheet Digital Equipment Corp. Maynard, Massachusetts	EC-QWC3C-TE

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 A-3. Related Specifications

Document Title and Source	Publication Number
IEEE - PCI Mezzanine Card Specification (PMC) Institute of Electrical and Electronics Engineers, Inc. Publication and Sales Department 345 East 47th Street New York, New York 10017-21633 Telephone: 1-800-678-4333	P1386.1 Draft 2.0
Peripheral Component Interconnect (PCI) Local Bus Specification, Revision 2.0 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
Intelligent I/O (I_2O) Architecture Specification Version 1.5 March 1997 I_2O Special Interest Group 404 Balboa Street San Francisco, CA 94118 Voice: 415-750-8352 Fax: 415-751-4829	N/A

Table A-3. Related Specifications (Continued) (Continued)

Document Title and Source	Publication Number
PC-MIP Specification VITA Standards Organization 7825 East Gelding Drive, Suite 104, Scottsdale AZ 85260	VITA 29 Draft 0.9a
PCI Mezzanine Card Specification IEEE Standards Department 445 Hoes Lane P.O Box 1331 Piscataway, NJ 08855-1331	P1386.1 Draft 2.0
Common Mezzanine Card Specification IEEE Standards Department 445 Hoes Lane P.O Box 1331 Piscataway, NJ 08855-1331	P1386 Draft 2.0
PCI Interface Specification Rev 2.1 PCI Special Interest Group 503-696-2000	PCI Rev 2.1

MCG Customer Services

The Motorola Computer Group Customer Services organization provides numerous services to support the needs of our OEM customers throughout the qualification, development, deployment, and continued service phases of their product life cycles. Specific areas of support include:

- Helping you identify and address your unique needs for providing outstanding products to your customers
- Extended warranties
- 24 x7 access to the high-level technical assistance that is crucial for mission critical applications around the world
- 24 x7 access to the latest technical information on MCG products, including known problems and a solutions database
- Customized training available at the MCG campus or at any of your sites across the world
- Customized documentation and 24 x7 Internet access to product documentation
- Services Central, a one-stop information source about customer services — program content, pricing, and availability

For information on what services are available, or to purchase a support contract, call us at Services Central, at 1-800-624-6745 or 602-438-5875, or visit our web site at http://www.mcg.mot.com/support.

Specifications

The following table lists the general specifications for the MVME2100 Single Board Computer. The subsequent sections detail cooling requirements and EMC regulatory compliance.

A complete functional description of the MVME2100 boards appears in Chapter 3. Specifications for the optional PMCs can be found in the documentation for those modules.

Table B-1. MVME2100 Specifications

Processors	• MPC8240 based • Bus Clock Frequencies of 66.67/83.33 MHz
Flash Memory	• Sockets for 1MB (8-bit) plus 4MB or 8MB (64-bit) of expansion Flash memory
System Memory	• 32 or 64MB ECC Synchronous DRAM
LAN	• DEC21143 10/100Base-TX Ethernet • Controller • LXT970 Fast Ethernet Transceiver
Interrupt Controller	• PowerPC™ Embedded Programmable Interrupt Controller (EPIC)
DMA	• 2 Independent DMA Channels
Timers	• Four Independent Timers
I^2C	• Integrated I^2C port with full master support
I_2O	• I_2O compliant messaging Interface
NVRAM	• 8KB (MK48T59Y)
RTC & Watchdog Timer	• MK48T59 device

B

Table B-1. MVME2100 Specifications (Continued)

Serial Interface	• One 16550-compatible async serial port
PCI Mezzanine Card(1)	• One 32-bit PMC slot • Front panel I/O • MVME2300 compatible P2 I/O
PC-MIP(1)	• Two 32-bit Type I PC-MIP slots (MVME2300 compatible P2 I/O) • Two 32-bit Type II PC-MIP slots (front panel I/O)
PCI Expansion	• Genesis II PCI expansion compatibility
Miscellaneous	• RESET switch • ABORT switch • Front panel status indicators
Form Factor	• Standard 6U VME

Note The power requirement listed for the MVME2100 does not include the power requirements for the PMC slots. The PMC specification allows for 7.5 watts per PMC slot. The 15 watts total can be drawn from any combination of the four voltage sources provided by the MVME2100: +3.3V, +5V, +12V, and -12V.

Cooling Requirements

The MVME2100 VME processor Module is specified, designed, and tested to operate reliably with an incoming air temperature range from 0° to 55° C (32° to 131° F) with forced air cooling of the entire assembly (base board and modules) at a velocity typically achievable by using a 100 CFM axial fan.

Temperature qualification is performed in a standard Motorola VMEsystem chassis. Twenty-five-watt load boards are inserted in two card slots, one on each side, adjacent to the board under test, to simulate a high power density system configuration. An assembly of three axial fans, rated at 100 CFM per fan, is placed directly under the VME card cage. The incoming air temperature is measured between the fan assembly and the card cage, where the incoming airstream first encounters the module under test. Test software is executed as the module is subjected to ambient temperature variations. Case temperatures of critical, high power density integrated circuits are monitored to ensure component vendors' specifications are not exceeded.

While the exact amount of airflow required for cooling depends on the ambient air temperature and the type, number, and location of boards and other heat sources, adequate cooling can usually be achieved with 10 CFM and 490 LFM flowing over the module. Less airflow is required to cool the module in environments having lower maximum ambients.

Under more favorable thermal conditions, it may be possible to operate the module reliably at higher than 55° C with increased airflow. It is important to note that there are several factors, in addition to the rated CFM of the air mover, which determine the actual volume and speed of air flowing over a module.

B

EMC Regulatory Compliance

The MVME2100 was tested in an EMC compliant chassis and meets the requirements for Class B equipment. Compliance was achieved under the following conditions:

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

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

The MVME2100 is a board level product and meant to be used in standard VME applications. As such, it is the responsibility of the OEM to meet the regulatory guidelines as determined by its application.

All external I/O connectors are shielded to aid in meeting EMC emissions standards. MVME2100 boards are tested in an MCG chassis for EMC compliance.

Connector Pin Assignments

C

Introduction

This chapter provides information on pin assignments for various connectors on board the MVME2100 Single Board Computer.

Connectors

VMEbus Connectors

VMEbus connectors P1 and P2 are the160-pin DIN type. Connector P1 provides power and VME signals for 24-bit address and 16-bit data. The pin assignments for this connector are specified by the IEEE P1014-1987 VMEbus Specification and the VME64 Extension Standard.

Row B of connector P2 provides power to the MVME2100, the upper 8 VMEbus address lines, and additional 16 VMEbus data lines. Rows A, C, Z, and D provide power and interface signals for the Type I PC-MIP and/or PMC mezzanine boards. The pin assignments for connector P2 are as follows:

Table C-1. P2 Connector Pin Assignment

PIN	ROW Z	ROW A	ROW B	ROW C	ROW D
1	PCMIP2 P3-59	PCMIP1 P3-63 PMC1 J14-2	+5V	PCMIP1 P3-64 PMC1 J14-1	PCMIP2 P3-60
2	GND	PCMIP1 P3-61 PMC1 J14-4	GND	PCMIP1 P3-62 PMC1 J14-3	PCMIP2 P3-57
3	PCMIP2 P3-55	PCMIP1 P3-59 PMC1 J14-6	NC	PCMIP1 P3-60 PMC1 J14-5	PCMIP2 P3-56
4	GND	PCMIP1 P3-57 PMC1 J14-8	VA24	 PMC1 J14-7	PCMIP2 P3-54

C

Table C-1. P2 Connector Pin Assignment (Continued)

PIN	ROW Z	ROW A	ROW B	ROW C	ROW D
5	PCMIP2 P3-51	PCMIP1 P3-55 PMC1 J14-10	VA25	PCMIP1 P3-56 PMC1 J14-9	PCMIP2 P3-52
6	GND	PMC1 J14-12	VA26	PCMIP1 P3-54 PMC1 J14-11	PCMIP2 P3-49
7	PCMIP2 P3-47	PCMIP1 P3-51 PMC1 J14-14	VA27	PCMIP1 P3-52 PMC1 J14-13	PCMIP2 P3-48
8	GND	PCMIP1 P3-49 PMC1 J14-16	VA28	PMC1 J14-15	PCMIP2 P3-46
9	PCMIP2 P3-43	PCMIP1 P3-47 PMC1 J14-18	VA29	PCMIP1 P3-48 PMC1 J14-17	PCMIP2 P3-44
10	GND	PMC1 J14-20	VA30	PCMIP1 P3-46 PMC1 J14-19	PCMIP2 P3-41
11	PCMIP2 P3-39	PCMIP1 P3-43 PMC1 J14-22	VA31	PCMIP1 P3-44 PMC1 J14-21	PCMIP2 P3-40
12	GND	PCMIP1 P3-41 PMC1 J14-24	GND	PMC1 J14-23	PCMIP2 P3-38
13	PCMIP2 P3-35	PCMIP1 P3-39 PMC1 J14-26	+5V	PCMIP1 P3-40 PMC1 J14-25	PCMIP2 P3-36
14	GND	PMC1 J14-28	VD16	PCMIP1 P3-38 PMC1 J14-27	PCMIP2 P3-33
15	PCMIP2 P3-31	PCMIP1 P3-35 PMC1 J14-30	VD17	PCMIP1 P3-36 PMC1 J14-29	PCMIP2 P3-32
16	GND	PCMIP1 P3-33 PMC1 J14-32	VD18	PMC1 J14-31	PCMIP2 P3-30
17	PCMIP2 P3-27	PCMIP1 P3-31 PMC1 J14-34	VD19	PCMIP1 P3-32 PMC1 J14-33	PCMIP2 P3-28
18	GND	PMC1 J14-36	VD20	PCMIP1 P3-30 PMC1 J14-35	PCMIP2 P3-25

Table C-1. P2 Connector Pin Assignment (Continued)

PIN	ROW Z	ROW A	ROW B	ROW C	ROW D
19	PCMIP2 P3-23	PCMIP1 P3-27 PMC1 J14-38	VD21	PCMIP1 P3-28 PMC1 J14-37	PCMIP2 P3-24
20	GND	PCMIP1 J3-25 PMC1 J14-40	VD22	 PMC1 J14-39	PCMIP2 J3-22
21	PCMIP2 J3-19	PCMIP1 J3-23 PMC1 J14-42	VD23	PCMIP1 J3-24 PMC1 J14-41	PCMIP2 J3-20
22	GND	 PMC1 J14-44	GND	PCMIP1 J3-22 PMC1 J14-43	PCMIP2 J3-17
23	PCMIP2 J3-15	PCMIP1 J3-19 PMC1 J14-46	VD24	PCMIP1 J3-20 PMC1 J14-45	PCMIP2 J3-16
24	GND	PCMIP1 J3-17 PMC1 J14-48	VD25	 PMC1 J14-47	PCMIP2 J3-14
25	PCMIP2 J3-11	PCMIP1 J3-15 PMC1 J14-50	VD26	PCMIP1 J3-16 PMC1 J14-49	PCMIP2 J3-12
26	GND	 PMC1 J14-52	VD27	PCMIP1 J3-14 PMC1 J14-51	PCMIP2 J3-9
27	PCMIP2 J3-7	PCMIP1 J3-11 PMC1 J14-54	VD28	PCMIP1 J3-12 PMC1 J14-53	PCMIP2 J3-8
28	GND	PCMIP1 J3-9 PMC1 J14-56	VD29	 PMC1 J14-55	PCMIP2 J3-6
29	PCMIP2 J3-3	PCMIP1 J3-7 PMC1 J14-58	VD30	PCMIP1 J3-8 PMC1 J14-57	PCMIP2 J3-4
30	GND	 PMC1 J14-60	VD31	PCMIP1 J3-6 PMC1 J14-59	PCMIP2 J3-2
31	PCMIP2 J3-1	PCMIP1 J3-3 PMC1 J14-62	GND	PCMIP1 J3-4 PMC1 J14-61	GND
32	GND	PCMIP1 J3-1 PMC1 J14-64	+5V	PCMIP1 J3-2 PMC1 J14-63	NC

Note PCMIP1 and PMC1 are factory build option slots. Either PCMIP1-P3 or PMC1-J14 will be populated, but never both.

PC-MIP PCI Interface Connectors

There are two 64-pin SMT connectors on the MVME2100 for each of the four PC-MIP board slots that are used to provide a 32-bit PCI interface for the optional add-on PC-MIP boards. The pin assignments are as follows:

Table C-2. PC-MIP P1/P2 Pin Assignments

PIN	P1		PIN	PIN	P2		PIN
1	RSVD	RSVD	2	1	RSVD	RSVD	2
3	RSVD	RSVD	4	3	RSVD	RSVD	4
5	-12V	TRST#	6	5	+5V	+5V	6
7	TCK	+12V	8	7	+5V	+5V	8
9	GND	TMS	10	9	REQ64#	ACK64#	10
11	TDO	TDI	12	11	+3.3V	+3.3V	12
13	+5V	+5V	14	13	AD00	AD01	14
15	+5V	INTA#	16	15	AD02	GND	16
17	INTB#	INTC#	18	17	GND	AD03	18
19	INTD#	+5V	20	19	AD04	AD05	20
21	PRSNT1#	RSVD	22	21	AD06	+3.3V	22
23	RSVD	+3.3V	24	23	+3.3V	AD07	24
25	PRSNT2#	RSVD	26	25	C/BE0#	AD08	26
27	GND	GND	28	27	GND	GND	28
29	RSVD	RSVD	30	29	AD09	M66EN	30
31	GND	RST#	32	31	GND	AD10	32
33	CLK	+3.3V	34	33	AD11	AD12	34

Table C-2. PC-MIP P1/P2 Pin Assignments (Continued)

PIN	P1		PIN	PIN	P2		PIN
35	GND	GNT#	36	35	AD13	GND	36
37	REQ#	GND	38	37	+3.3V	AD14	38
39	+3.3V	RSVD	40	39	AD15	C/BE1#	40
41	AD31	AD30	42	41	PAR	+3.3V	42
43	AD29	+3.3V	44	43	GND	SERR#	44
45	GND	AD28	46	45	SBO#	+3.3V	46
47	AD27	AD26	48	47	SDONE	PERR#	48
49	AD25	GND	50	49	+3.3V	LOCK#	50
51	+3.3V	AD24	52	51	STOP#	GND	52
53	C/BE3#	IDSEL	54	53	GND	DEVSEL#	54
55	AD23	+3.3V	56	55	TRDY#	+3.3V	56
57	GND	AD22	58	57	GND	IRDY#	58
59	AD21	AD20	60	59	FRAME#	GND	60
61	AD19	GND	62	61	+3.3V	C/BE2#	62
63	+3.3V	AD18	64	63	AD16	AD17	64

PC-MIP USER Defined I/O Connectors

There is one 64-pin SMT connector on the MVME2100 for each of the two PC-MIP Type I board slots that are used to provide a user defined interface for the optional add-on PC-MIP Type I boards. The pin assignments are as follows:

Table C-3. PC-MIP Slot 1 User Defined I/O Connector P3 Pin Assignments

PIN	PC-MIP Slot 1 User Defined I/O P3		PIN
1	PC-MIP IO (P2-A32)	PC-MIP IO (P2-C32)	2
3	PC-MIP IO (P2-A31)	PC-MIP IO (P2-C31)	4
5	GND	PC-MIP IO (P2-C30)	6
7	PC-MIP IO (P2-A29)	PC-MIP IO (P2-C29)	8
9	PC-MIP IO (P2-A28)	GND	10
11	PC-MIP IO (P2-A27)	PC-MIP IO (P2-C27)	12
13	GND	PC-MIP IO (P2-C26)	14
15	PC-MIP IO (P2-A25)	PC-MIP IO (P2-C25)	16
17	PC-MIP IO (P2-A24)	GND	18
19	PC-MIP IO (P2-A23)	PC-MIP IO (P2-C23)	20
21	GND	PC-MIP IO (P2-C22)	22
23	PC-MIP IO (P2-A21)	PC-MIP IO (P2-C21)	24
25	PC-MIP IO (P2-A20)	GND	26
27	PC-MIP IO (P2-CA19)	PC-MIP IO (P2-C19)	28
29	GND	PC-MIP IO (P2-C18)	30
31	PC-MIP IO (P2-A17)	PC-MIP IO (P2-C17)	32

Table C-3. PC-MIP Slot 1 User Defined I/O Connector P3 Pin Assignments (Continued)

PIN	PC-MIP Slot 1 User Defined I/O P3		PIN
33	PC-MIP IO (P2-A16)	GND	34
35	PC-MIP IO (P2-CA15)	PC-MIP IO (P2-C15)	36
37	GND	PC-MIP IO (P2-C14)	38
39	PC-MIP IO (P2-A13)	PC-MIP IO (P2-C13)	40
41	PC-MIP IO (P2-A12)	GND	42
43	PC-MIP IO (P2-A11)	PC-MIP IO (P2-C11)	44
45	GND	PC-MIP IO (P2-C10)	46
47	PC-MIP IO (P2-A9)	PC-MIP IO (P2-C9)	48
49	PC-MIP IO (P2-A8)	GND	50
51	PC-MIP IO (P2-A7)	PC-MIP IO (P2-C7)	52
53	GND	PC-MIP IO (P2-C6)	54
55	PC-MIP IO (P2-A5)	PC-MIP IO (P2-C5)	56
57	PC-MIP IO (A4)	GND	58
59	PC-MIP IO (P2-A3)	PC-MIP IO (P2-C3)	60
61	PC-MIP IO (P2-A2)	PC-MIP IO (P2-C2)	62
63	PC-MIP IO (P2-A1)	PC-MIP IO (P2-C1)	64

C

Table C-4. PC-MIP Slot 2 User Defined I/O Connector P3 Pin Assignments

PIN	PC-MIP Slot 2 User Defined I/O P3		PIN
1	PC-MIP IO (P2-Z31)	PC-MIP IO (P2-D30)	2
3	PC-MIP IO (P2-Z29)	PC-MIP IO (P2-D29)	4
5	GND	PC-MIP IO (P2-D28)	6
7	PC-MIP IO (P2-Z27)	PC-MIP IO (P2-D27)	8
9	PC-MIP IO (P2-D26)	GND	10
11	PC-MIP IO (P2-Z25)	PC-MIP IO (P2-D25)	12
13	GND	PC-MIP IO (P2-D24)	14
15	PC-MIP IO (P2-Z23)	PC-MIP IO (P2-D23)	16
17	PC-MIP IO (P2-D22)	GND	18
19	PC-MIP IO (P2-Z21)	PC-MIP IO (P2-D21)	20
21	GND	PC-MIP IO (P2-D20)	22
23	PC-MIP IO (P2-Z19)	PC-MIP IO (P2-D19)	24
25	PC-MIP IO (P2-D18)	GND	26
27	PC-MIP IO (P2-Z17)	PC-MIP IO (P2-D17)	28
29	GND	PC-MIP IO (P2-D16)	30
31	PC-MIP IO (P2-Z15)	PC-MIP IO (P2-D15)	32
33	PC-MIP IO (P2-D14)	GND	34
35	PC-MIP IO (P2-Z13)	PC-MIP IO (P2-D13)	36
37	GND	PC-MIP IO (P2-D12)	38

Table C-4. PC-MIP Slot 2 User Defined I/O Connector P3 Pin Assignments (Continued)

PIN	PC-MIP Slot 2 User Defined I/O P3		PIN
39	PC-MIP IO (P2-Z11)	PC-MIP IO (P2-D11)	40
41	PC-MIP IO (P2-D10)	GND	42
43	PC-MIP IO (P2-Z9)	PC-MIP IO (P2-D9)	44
45	GND	PC-MIP IO (P2-D8)	46
47	PC-MIP IO (P2-Z7)	PC-MIP IO (P2-D7)	48
49	PC-MIP IO (P2-D6)	GND	50
51	PC-MIP IO (P2-Z5)	PC-MIP IO (P2-D5)	52
53	GND	PC-MIP IO (P2-D4)	54
55	PC-MIP IO (P2-Z3)	PC-MIP IO (P2-D3)	56
57	PC-MIP IO (P2-D2)	GND	58
59	PC-MIP IO (P2-Z1)	PC-MIP IO (P2-D1)	60
61	Not Connected	Not Connected	62
63	Not Connected	Not Connected	64

PCI Mezzanine Card (PMC) PCI Interface Connectors

There are two 64-pin SMT connectors on the MVME2100 to provide a 32-bit PCI interface for one optional add-on PCI Mezzanine Card (PMC). The pin assignments are as follows:

Table C-5. PMC Connector J11/J12 Pin Assignments

PIN	J11		PIN	PIN	J12		PIN
1	TCK	-12V	2	1	+12V	TRST#	2
3	GND	INTA#	4	3	TMS	TDO	4
5	INTB#	INTC#	6	5	TDI	GND	6
7	PMCPRSNT#	+5V	8	7	GND	Not Used	8
9	INTD#	Not Used	10	9	Not Used	Not Used	10
11	GND	Not Used	12	11	Pull-up	+3.3V	12
13	CLK	GND	14	13	RST#	Pull-down	14
15	GND	PMCGNT#	16	15	+3.3V	Pull-down	16
17	PMCREQ#	+5V	18	17	Not Used	GND	18
19	+5V (Vio)	AD31	20	19	AD30	AD29	20
21	AD28	AD27	22	21	GND	AD26	22
23	AD25	GND	24	23	AD24	+3.3V	24
25	GND	C/BE3#	26	25	IDSEL	AD23	26
27	AD22	AD21	28	27	+3.3V	AD20	28
29	AD19	+5V	30	29	AD18	GND	30
31	+5V (Vio)	AD17	32	31	AD16	C/BE2#	32
33	FRAME#	GND	34	33	GND	Not Used	34

Table C-5. PMC Connector J11/J12 Pin Assignments (Continued)

PIN	J11		PIN	PIN	J12		PIN
35	GND	IRDY#	36	35	TRDY#	+3.3V	36
37	DEVSEL#	+5V	38	37	GND	STOP#	38
39	GND	LOCK#	40	39	PERR#	GND	40
41	SDONE#	SBO#	42	41	+3.3V	SERR#	42
43	PAR	GND	44	43	C/BE1#	GND	44
45	+5V (Vio)	AD15	46	45	AD14	AD13	46
47	AD12	AD11	48	47	GND	AD10	48
49	AD09	+5V	50	49	AD08	+3.3V	50
51	GND	C/BE0#	52	51	AD07	Not Used	52
53	AD06	AD05	54	53	+3.3V	Not Used	54
55	AD04	GND	56	55	Not Used	GND	56
57	+5V (Vio)	AD03	58	57	Not Used	Not Used	58
59	AD02	AD01	60	59	GND	Not Used	60
61	AD00	+5V	62	61	ACK64#	+3.3V	62
63	GND	REQ64#	64	63	GND	Not Used	64

PMC USER Defined I/O Connector

There is one 64-pin SMT connector on the MVME2100 for the PMC board slot that is used to provide a user defined interface for the optional add-on PMC board. The pin assignments are as follows:

Table C-6. PMC User Defined I/O Connector J14 Pin Assignments

PIN	PMC User Defined I/O J14		PIN
1	PMC IO (P2-C1)	PMC IO (P2-A1)	2
3	PMC IO (P2-C2)	PMC IO (P2-A2)	4
5	PMC IO (P2-C3)	PMC IO (P2-A3)	6
7	PMC IO (P2-C4)	PMC IO (P2-A4)	8
9	PMC IO (P2-C5)	PMC IO (P2-A5)	10
11	PMC IO (P2-C6)	PMC IO (P2-A6)	12
13	PMC IO (P2-C7)	PMC IO (P2-A7)	14
15	PMC IO (P2-C8)	PMC IO (P2-A8)	16
17	PMC IO (P2-C9)	PMC IO (P2-A9)	18
19	PMC IO (P2-C10)	PMC IO (P2-A10)	20
21	PMC IO (P2-C11)	PMC IO (P2-A11)	22
23	PMC IO (P2-C12)	PMC IO (P2-A12)	24
25	PMC IO (P2-C13)	PMC IO (P2-A13)	26
27	PMC IO (P2-C14)	PMC IO (P2-A14)	28
29	PMC IO (P2-C15)	PMC IO (P2-A15)	30
31	PMC IO (P2-C16)	PMC IO (P2-A16)	32

Table C-6. PMC User Defined I/O Connector J14 Pin Assignments (Continued)

PIN	PMC User Defined I/O J14		PIN
33	PMC IO (P2-C17)	PMC IO (P2-A17)	34
35	PMC IO (P2-C18)	PMC IO (P2-A18)	36
37	PMC IO (P2-C19)	PMC IO (P2-A19)	38
39	PMC IO (P2-C20)	PMC IO (P2-A20)	40
41	PMC IO (P2-C21)	PMC IO (P2-A21)	42
43	PMC IO (P2-C22)	PMC IO (P2-A22)	44
45	PMC IO (P2-C23)	PMC IO (P2-A23)	46
47	PMC IO (P2-C24)	PMC IO (P2-A24)	48
49	PMC IO (P2-C25)	PMC IO (P2-A25)	50
51	PMC IO (P2-C26)	PMC IO (P2-A26)	52
53	PMC IO (P2-C27)	PMC IO (P2-A27)	54
55	PMC IO (P2-C28)	PMC IO (P2-A28)	56
57	PMC IO (P2-C29)	PMC IO (P2-A29)	58
59	PMC IO (P2-C30)	PMC IO (P2-A30)	60
61	PMC IO (P2-C31)	PMC IO (P2-A31)	62
63	PMC IO (P2-C32)	PMC IO (P2-A32)	64

Debug Connector

One 190-pin *Mictor* connector with a center row of power and ground pins is used to provide access to the memory bus and some miscellaneous signals. Pin assignments for this connector are as follows:

Table C-7. Debug Connector Pin Assignments

PIN	ASSIGNMENT		ASSIGNMENT	PIN
1	MA0		MA1	2
3	MA2		MA3	4
5	MA4		MA5	6
7	MA6		MA7	8
9	MA8		MA9	10
11	MA10		MA11	12
13	MA12		MA13	14
15	NC.	**GND**	NC.	16
17	DQM0		DQM1	18
19	DQM2		DQM3	20
21	DQM4		DQM5	22
23	DQM6		DQM7	24
25	NC.		NC.	26
27	CS0#		CS1#	28
29	NC		NC.	30
31	NC		NC.	32
33	NC.		NC.	34
35	SDRAS#		SDCAS#	36
37	WE#		CKE	38

Table C-7. Debug Connector Pin Assignments (Continued)

PIN	ASSIGNMENT		ASSIGNMENT	PIN
39	DH0		DH1	40
41	DH2		DH3	42
43	DH4		DH5	44
45	DH6		DH7	46
47	DH8	**+5V**	DH9	48
49	DH10		DH11	50
51	DH12		DH13	52
53	DH14		DH15	54
55	DH16		DH17	56
57	DH18		DH19	58
59	PA20		DH21	60
61	DH22		DH23	62
63	DH24		DH25	64
65	DH26		DH27	66
67	DH28		DH29	68
69	DH30		DH31	70
71	DL32		DL33	72
73	DL34		DL35	74
75	DL36		DL37	76

Table C-7. Debug Connector Pin Assignments (Continued)

PIN	ASSIGNMENT		ASSIGNMENT	PIN
77	DL38	GND	DL39	78
79	DL40		DL41	80
81	DL42		DL43	82
83	DL44		DL45	84
85	DL46		DL47	86
87	DL48		DL49	88
89	PA50		DL51	90
91	DL52		DL53	92
93	DL54		DL55	94
95	DL56		DL57	96
97	DL58		DL59	98
99	DL60		DL61	100
101	DL62		DL63	102
103	CB0		CB1	104
105	CB2		CB3	106
107	CB4		CB5	108
109	CB6		CB7	110
111	FWE#		FOE#	112
113	RCS0#		RCS1#	114

Table C-7. Debug Connector Pin Assignments (Continued)

PIN	ASSIGNMENT		ASSIGNMENT	PIN
115	AS#		NC.	116
117	NC.		NC.	118
119	NC.		NC.	120
121	NC.		NC.	122
123	REQ0#		GNTO#	124
125	REQ1#		GNT1#	126
127	REQ2#		GNT2#	128
129	REQ3#	+3.3V	GNT3#	130
131	REQ4#		GNT4#	132
133	NC.		NC.	134
135	INTOUT#		NC.	136
137	SINT#		NC.	138
139	INT_SYNC#		NC.	140
141	INT_RST		NC.	142
143	IRQ0		NC.	144
145	IRQ1		NC.	146
147	IRQ2		NC.	148
149	IRQ3		NC.	150
151	IRQ4		NC.	152

C

Table C-7. Debug Connector Pin Assignments (Continued)

PIN	ASSIGNMENT		ASSIGNMENT	PIN
153	IRQ5		NC.	154
155	IRQ6		MCP#	156
157	IRQ7		SMI#	158
159	IRQ8		NC.	160
161	IRQ9		NC.	162
163	IRQ10		NC.	164
165	IRQ11		NC.	166
167	IRQ12	GND	NC.	168
169	IRQ13		SUSPEND#	170
171	IRQ14		NC.	172
173	IRQ15		NC.	174
175	NC.		NC.	176
177	NC.		NC.	178
179	NC.		NC.	180
181	HRESET#		TDO	182
183	GND		TDI	184
185	CPUCLK		TCK	186
187	CPUCLK		TMS	188
189	CPUCLK		TRST#	190

PCI Expansion Connector

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

Table C-8. PCI Expansion Connector Pin Assignments

PIN	ASSIGNMENT		ASSIGNMENT	PIN
1	+3.3V		+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#	**GND**	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 C-8. PCI Expansion Connector Pin Assignments (Continued)

PIN	ASSIGNMENT		ASSIGNMENT	PIN
39	PAR		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	**+5V**	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 C-8. PCI Expansion Connector Pin Assignments (Continued)

PIN	ASSIGNMENT		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

10/100Base-TX Connector

One 10/100Base-TX RJ45 connector is located on the front panel of the CPU board. The pin assignments for this connector are as follows:

Table C-9. 10/100Base-TX Connector Pin Assignments

PIN	ASSIGNMENT
1	TD+
2	TD-
3	RD+
4	No Connect
5	No Connect
6	RD-
7	No Connect
8	No Connect

Asynchronous Serial Port Connector

A standard RJ45 connector located on the front panel of the CPU board provides the interface to the asynchronous serial port. The pin assignments for this connector are as follows:

Table C-10. Asynchronous Serial Connector Pin Assignments

PIN	ASSIGNMENT
1	DCD (input)
2	RTS (output)
3	GND
4	TXD (output)
5	RXD (input)
6	GND
7	CTS (input)
8	DTR (output)

Two-Wire Serial Interface Header

A 4 pin header on the CPU board is used to support external two-wire serial devices as a test aide. The pin assignments for this connector are as follows:

Table C-11. Two-Wire Serial Interface Header Pin Assignments

PIN	ASSIGNMENT
1	+5V
2	SCLK
3	SDATA
4	GND

Solving Startup Problems

In the event of difficulty with your MVME2100 Single Board Computer, try the simple troubleshooting steps on the following pages before calling for help or sending the board back for repair. Some of the procedures will return the board to the factory debugger environment. Please note that the board was tested under these conditions before it left the factory. The self tests may not run in all user-customized environments.

Table D-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 correct power.	1. Make sure the system is plugged in. 2. Check that the board is securely installed in its backplane or chassis. 3. Check that all necessary cables are connected to the board, per this manual. 4. Check for compliance with Installation Considerations, per this manual. 5. Review the Installation and Startup procedures, per this manual. They include a step-by-step powerup routine. Try it.
	B. If the LEDs are lit, the board may be in the wrong slot.	1. The VME processor module should be in the first (leftmost) slot. 2. Also check that the “system controller” function on the board is enabled, per this manual.
	C. The “system console” terminal may be configured incorrectly.	Configure the system console terminal per this manual.

Table D-1. Troubleshooting Problems (Continued)

Condition	Possible Problem	Possible Resolution:
II. There is a display on the terminal, but input from the keyboard and/or mouse has no effect.	A. The keyboard or mouse may be connected incorrectly.	Recheck the keyboard and/or mouse connections and power.
	B. Board jumpers may be configured incorrectly.	Check the board jumpers per this manual.
	C. You may have invoked flow control by pressing a HOLD or PAUSE key, or by typing: **<CTRL>-S**	Press the HOLD or PAUSE key again. If this does not free up the keyboard, type in: **<CTRL>-Q**
III. Debug prompt PPC5-Bug> does not appear at powerup, and the board does not autoboot.	A. Debugger Flash may be missing	1. Disconnect *all* power from your system. 2. Check that the proper debugger devices are installed. 3. Reconnect power. 4. Restart the system by "double-button reset": press the RST and ABT switches at the same time; release RST first, wait seven seconds, then release ABT. 5. If the debug prompt appears, go to step IV or step V, as indicated. If the debug prompt does not appear, go to step VI.
	B. The board may need to be reset.	
IV. Debug prompt PPC5-Bug> appears at powerup, but the board does not autoboot.	A. The initial debugger environment parameters may be set incorrectly.	1. Start the onboard calendar clock and timer. Type: **set *mmddyyhhmm* <CR>** where the characters indicate the month, day, year, hour, and minute. The date and time will be displayed. **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.	

(continues>)

Table D-1. Troubleshooting Problems (Continued)

Condition	Possible Problem	Possible Resolution:
IV. *Continued*		2. At the command line prompt, type in: **env;d <CR>** This sets up the default parameters for the debugger environment. 3. When prompted to Update Non-Volatile RAM, type in: **y <CR>** 4. When prompted to Reset Local System, type in: **y <CR>** 5. After clock speed is displayed, immediately (within five seconds) press the Return key: **<CR>** or **BREAK** to exit to the System Menu. Then enter a 3 for "Go to System Debugger" and Return: **3 <CR>** Now the prompt should be: `PPC5-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.
V. The debugger is in system mode and the board autoboots, or the board has passed selftests.	A. No apparent problems — troubleshooting is done.	No further troubleshooting steps are required.

Table D-1. Troubleshooting Problems (Continued)

Condition	Possible Problem	Possible Resolution:
VI. The board has failed one or more of the tests listed above, and cannot be corrected using the steps given.	A. There may be some fault in the board hardware or the on-board debugging and diagnostic firmware.	1. Document the problem and return the board for service. 2. Phone 1-800-222-5640.
TROUBLESHOOTING PROCEDURE COMPLETE.		

Glossary

Abbreviations, Acronyms, and Terms to Know

This glossary defines some of the abbreviations, acronyms, and key terms used in this document.

10Base-T An Ethernet implementation in which the physical medium is an unshielded twisted pair (UTP) of wires capable of carrying data at 10 Mbps for a maximum distance of 185 meters. Also known as twisted-pair Ethernet.

100Base-TX An Ethernet implementation in which the physical medium is an unshielded twisted pair (UTP) of wires capable of carrying data at 100 Mbps for a maximum distance of 100 meters. Also known as fast Ethernet.

ACIA **A**synchronous **C**ommunications **I**nterface **A**dapter

AIX **A**dvanced **I**nteractive e**X**ecutive (IBM version of UNIX)

architecture The main overall design in which each individual hardware component of the computer system is interrelated. The most common uses of this term are 8-bit, 16-bit, or 32-bit architectural design systems.

ASCII **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange. This is a 7-bit code used to encode alphanumeric information. In the IBM-compatible world, this is expanded to 8-bits to encode a total of 256 alphanumeric and control characters.

ASIC **A**pplication-**S**pecific **I**ntegrated **C**ircuit

AUI **A**ttachment **U**nit **I**nterface

BBRAM Battery Backed-up Random Access Memory

bi-endian Having big-endian and little-endian byte ordering capability.

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

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

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

BLT — **BL**ock **T**ransfer

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

bpi — **b**its **p**er **i**nch

bps — **b**its **p**er **s**econd

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

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

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

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

CHRP — See Common Hardware Reference Platform (CHRP).

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

CHRP Spec — See Common Hardware Reference Platform (CHRP).

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

CODEC — **CO**der/**DEC**oder

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

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

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

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

cpl **c**haracters **p**er **l**ine

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

fast Ethernet — See 100Base-TX.

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

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

FIFO — **F**irst-**I**n, **F**irst-**O**ut. A memory that can temporarily hold data so that the sending device can send data faster than the receiving device can accept it. The sending and receiving devices typically operate asynchronously.

firmware — The program or specific software instructions that have been more or less permanently burned into an electronic component, such as a ROM (read-only memory) or an EPROM (erasable programmable read-only memory).

frame — One complete television picture frame consists of 525 horizontal lines with the NTSC system. One frame consists of two Fields.

graphics controller — On EGA and VGA, a section of circuitry that can provide hardware assist for graphics drawing algorithms by performing logical functions on data written to display memory.

HAL — **H**ardware **A**bstraction **L**ayer. The lower level hardware interface module of the Windows NT operating system. It contains platform specific functionality.

hardware	A computing system is normally spoken of as having two major components: hardware and software. Hardware is the term used to describe any of the physical embodiments of a computer system, with emphasis on the electronic circuits (the computer) and electromechanical devices (peripherals) that make up the system.
HCT	**H**ardware **C**onformance **T**est. A test used to ensure that both hardware and software conform to the Windows NT interface.
I/O	**I**nput/**O**utput
IBC	PCI/**I**SA **B**ridge **C**ontroller
IDC	**I**nsulation **D**isplacement **C**onnector
IDE	**I**ntegrated **D**rive **E**lectronics. A disk drive interface standard. Also known as **ATA** (**A**dvanced **T**echnology **A**ttachment).
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
interlaced	A graphics system in which the even scanlines are refreshed in one vertical cycle (field), and the odd scanlines are refreshed in another vertical cycle. The advantage is that the video bandwidth is roughly half that required for a non-interlaced system of the same resolution. This results in less costly hardware. It also may make it possible to display a resolution that would otherwise be impossible on given hardware. The disadvantage of an interlaced system is flicker, especially when displaying objects that are only a few scanlines high.
IQ Signals	Similar to the color difference signals (R-Y), (B-Y) but using different vector axis for encoding or decoding. Used by some USA TV and IC manufacturers for color decoding.
ISA (bus)	**I**ndustry **S**tandard **A**rchitecture (bus). The de facto standard system bus for IBM-compatible computers until the introduction of VESA and PCI. Used in the reference platform specification. (IBM)
ISASIO	**ISA S**uper **I**nput/**O**utput device
ISDN	**I**ntegrated **S**ervices **D**igital **N**etwork. A standard for digitally transmitting video, audio, and electronic data over public phone networks.
LAN	**L**ocal **A**rea **N**etwork
LED	**L**ight-**E**mitting **D**iode
LFM	**L**inear **F**eet per **M**inute

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

MBLT **M**ultiplexed **BL**ock **T**ransfer

MCA (bus) **M**icro **C**hannel **A**rchitecture

MCG **M**otorola **C**omputer **G**roup

MFM **M**odified **F**requency **M**odulation

MIDI **M**usical **I**nstrument **D**igital **I**nterface. The standard format for recording, storing, and playing digital music.

MPC **M**ultimedia **P**ersonal **C**omputer

MPC105 The PowerPC-to-PCI bus bridge chip developed by Motorola for the Ultra 603/Ultra 604 system board. It provides the necessary interface between the MPC603/MPC604 processor and the Boot ROM (secondary cache), the DRAM (system memory array), and the PCI bus.

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

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

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

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

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

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

multisession	The ability to record additional information, such as digitized photographs, on a CD-ROM after a prior recording session has ended.
non-interlaced	A video system in which every pixel is refreshed during every vertical scan. A non-interlaced system is normally more expensive than an interlaced system of the same resolution, and is usually said to have a more pleasing appearance.
nonvolatile memory	A memory in which the data content is maintained whether the power supply is connected or not.
NTSC	**N**ational **T**elevision **S**tandards **C**ommittee (USA)
NVRAM	**N**on-**V**olatile **R**andom **A**ccess **M**emory
OEM	**O**riginal **E**quipment **M**anufacturer
OMPAC	**O**ver - **M**olded **P**ad **A**rray **C**arrier
OS	**O**perating **S**ystem. The software that manages the computer resources, accesses files, and dispatches programs.
OTP	**O**ne-**T**ime **P**rogrammable
palette	The range of colors available on the screen, not necessarily simultaneously. For VGA, this is either 16 or 256 simultaneous colors out of 262,144.
parallel port	A connector that can exchange data with an I/O device eight bits at a time. This port is more commonly used for the connection of a printer to a system.
PCI (local bus)	**P**eripheral **C**omponent **I**nterconnect (local bus) (Intel). A high-performance, 32-bit internal interconnect bus used for data transfer to peripheral controller components, such as those for audio, video, and graphics.
PCMCIA (bus)	**P**ersonal **C**omputer **M**emory **C**ard **I**nternational **A**ssociation (bus). A standard external interconnect bus which allows peripherals adhering to the standard to be plugged in and used without further system modification.
PCR	**P**CI **C**onfiguration **R**egister
PDS	**P**rocessor **D**irect **S**lot
PHB	**P**CI **H**ost **B**ridge

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

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

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

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

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

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

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

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

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

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

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

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

PRP See PowerPC Reference Platform (PRP).

PRP-compliant See PowerPC Reference Platform (PRP).

PRP Spec See PowerPC Reference Platform (PRP).

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

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

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

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

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

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

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

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

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

RISC See Reduced Instruction Set Computer (RISC).

ROM	**R**ead-**O**nly **M**emory
RTC	**R**eal-**T**ime **C**lock
SBC	**S**ingle **B**oard **C**omputer
SCSI	**S**mall **C**omputer **S**ystems **I**nterface. An industry-standard high-speed interface primarily used for secondary storage. While the oldest standard, SCSI-1 provides up to 5 Mbps data transfer, the most current (as of the printing date of this manual) ULTRA-160 provides transfer rates of 160 Mbps (a two-fold increase over ULTRA-2 LVD which stands at 80 Mbps).
serial port	A connector that can exchange data with an I/O device one bit at a time. It may operate synchronously or asynchronously, and may include start bits, stop bits, and/or parity.
SIM	**S**erial **I**nterface **M**odule
SIMM	**S**ingle **I**nline **M**emory **M**odule. A small circuit board with RAM chips (normally surface mounted) on it designed to fit into a standard slot.
SIO	**S**uper **I/O** controller
SMP	**S**ymmetric **M**ulti**P**rocessing. A computer architecture in which tasks are distributed among two or more local processors.
SMT	**S**urface **M**ount **T**echnology. A method of mounting devices (such as integrated circuits, resistors, capacitors, and others) on a printed circuit board, characterized by not requiring mounting holes. Rather, the devices are soldered to pads on the printed circuit board. Surface-mount devices are typically smaller than the equivalent through-hole devices.
software	A computing system is normally spoken of as having two major components: hardware and software. Software is the term used to describe any single program or group of programs, languages, operating procedures, and documentation of a computer system. Software is the real interface between the user and the computer.
SRAM	**S**tatic **R**andom **A**ccess **M**emory
SSBLT	**S**ource **S**ynchronous **BL**ock **T**ransfer
standard(s)	A set of detailed technical guidelines used as a means of establishing uniformity in an area of hardware or software development.

SVGA — **S**uper **V**ideo **G**raphics **A**rray (IBM). An improved VGA monitor standard that provides at least 256 simultaneous colors and a screen resolution of 800 x 600 pixels.

Teletext — One way broadcast of digital information. The digital information is injected in the broadcast TV signal, VBI, or full field, The transmission medium could be satellite, microwave, cable, etc. The display medium is a regular TV receiver.

thick Ethernet — See 10base-5.

thin Ethernet — See 10base-2.

twisted-pair Ethernet — See 10Base-T.

UART — **U**niversal **A**synchronous **R**eceiver/**T**ransmitter

Universe — ASIC developed by Tundra in consultation with Motorola, that provides the complete interface between the PCI bus and the 64-bit VMEbus.

UV — **U**ltra**V**iolet

UVGA — **U**ltra **V**ideo **G**raphics **A**rray. An improved VGA monitor standard that provides at least 256 simultaneous colors and a screen resolution of 1024 x 768 pixels.

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

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

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

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

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

VMEchip2 — MCG second generation VMEbus interface ASIC (Motorola)

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

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

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

Windows NT™ The trademark representing **Windows N**ew **T**echnology, a computer operating system developed by the Microsoft Corporation.

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

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

Index

D

E

F

G

H

I

INDEX

J

L

M

N

O

P

R

S

T

INDEX

U

V