ARTISAN®
TECHNOLOGY GROUP

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

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

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

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

Custom engineering
so your equipment works exactly as you specify.

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

Expert team | Trust guarantee | 100% satisfaction

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

Find the *Emerson / Motorola MVME2301* at our website: *Click HERE*

MVME2300SC
VME Processor Module

Installation and Use

V2300SCA/IH2

Edition of March 2001

© Copyright 2001 Motorola, Inc.

All rights reserved.

Printed in the United States of America.

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

PowerPC® is a registered trademark of International Business Machines Corporation and is used by Motorola with permission.

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

Safety Summary

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

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

Ground the Instrument.

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

Do Not Operate in an Explosive Atmosphere.

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

Keep Away From Live Circuits Inside the Equipment.

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

Use Caution When Exposing or Handling a CRT.

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

Do Not Substitute Parts or Modify Equipment.

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

Observe Warnings in Manual.

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



Warning

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

Flammability

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

EMI Caution



Caution

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

Lithium Battery Caution

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



Caution

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



Attention

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



Vorsicht

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

CE Notice (European Community)

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

> EN55022 “Limits and Methods of Measurement of Radio Interference Characteristics of Information Technology Equipment”; this product tested to Equipment Class B
>
> EN50082-1:1997 “Electromagnetic Compatibility—Generic Immunity Standard, Part 1. Residential, Commercial and Light Industry”

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

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

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

Notice

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

Contents

About this Manual

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

CHAPTER 1 Hardware Preparation and Installation

Getting Started 1-1
- Overview of Installation Procedure 1-1
- Equipment Required 1-3
- Guidelines for Unpacking 1-4
- ESD Precautions 1-4

Preparing the Board 1-5
- MVME2300SC 1-6
  - Flash Bank Selection (J8) 1-8
  - VMEbus System Controller (J9) 1-9
  - General-Purpose Readable Jumpers (J10) 1-9
- PMCs 1-10
- PMCspan 1-10

Installing the Hardware 1-11
- PMC Modules 1-11
- Primary PMCspan Module 1-13
- Secondary PMCspan Module 1-16
- MVME2300SC 1-18
- System Console Terminal 1-20
- Installation Considerations 1-21

CHAPTER 2 Startup and Operation

Introduction 2-1
- Switches and LEDs 2-1

Initial Conditions 2-2
Applying Power 2-3
Pre-Startup Checklist 2-4
Bringing up the Board 2-5

Autoboot 2-7
ROMboot 2-8
Network Boot 2-9
Restarting the System 2-9
Reset 2-10
Abort 2-11
Break 2-11
Diagnostic Facilities 2-11

CHAPTER 3 PPCBug Firmware

Overview 3-1
PPCBug Basics 3-1
PPCBug Implementation 3-2
Memory Requirements 3-3
Using PPCBug 3-3
Debugger Commands 3-4
Diagnostic Tests 3-8
Modifying the Environment 3-9
CNFG - Configure Board Information Block 3-10
ENV - Set Environment 3-11
Configuring the PPCBug Parameters 3-12
Configuring the VMEbus Interface 3-21

CHAPTER 4 Functional Description

Introduction 4-1
Summary of Features 4-1
General Description 4-3
MPC604 Processor 4-3
I/O Implementation 4-4
ASICs 4-5
DRAM Memory 4-5
DRAM Latency 4-7
Flash Memory 4-9
Flash Latency 4-10
Ethernet Interface 4-11
PCI Mezzanine Card (PMC) Interface 4-12
PMC Slot 1 (Single-Width PMC) 4-13
PMC Slot 2 (Single-Width PMC) 4-13
PMC Slots 1 and 2 (Double-Width PMC) 4-14

PCI Expansion 4-14
VMEbus Interface 4-14
Asynchronous Debug Port 4-15
PCI/ISA Bridge (PIB) Controller 4-15
Real-Time Clock/NVRAM/Timer Function 4-16
PCI Host Bridge 4-17
Interrupt Controller (MPIC) 4-17
Programmable Timers 4-17
Interval Timers 4-18
16/32-Bit Timers 4-18
Connectors 4-19
Status Indicators 4-20
10/100 BASET Port 4-20
DEBUG Port 4-20

CHAPTER 5 Pin Assignments

Connector Pin Assignments 5-1
VMEbus Connectors (P1, P2) 5-1
Serial Port Connector - DEBUG (J2) 5-4
Ethernet Connector - 10BASET (J3) 5-4
CPU Debug Connector - J1 5-5
PCI Expansion Connector - J6 5-10
PCI Mezzanine Card Connectors - J11 through J14 5-12
PCI Mezzanine Card Connectors - J21 through J24 5-15

APPENDIX A Specifications

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

APPENDIX B Troubleshooting

Solving Startup Problems B-1

APPENDIX C Related Documentation

Motorola Computer Group Documents C-1
Manufacturers' Documents C-2
Related Specifications C-5

Glossary

Index

List of Figures

Figure 1-1. MVME2300SC Layout 1-7
Figure 1-2. General-Purpose Software-Readable Header 1-10
Figure 1-3. Typical Single-width PMC Module Placement on MVME2300SC 1-12
Figure 1-4. PMCspan-002 Installation on an MVME2300SC 1-15
Figure 1-5. PMCspan-010 Installation on a PMCspan-002/MVME2300SC 1-17
Figure 2-1. MVME2300SC/Firmware System Startup 2-3
Figure 4-1. MVME2300SC Block Diagram 4-6
Figure 4-2. Memory Block Diagram 4-7

List of Tables

Table 1-1. Startup Overview 1-1
Table 1-2. MVME2300SC Jumper Settings 1-6
Table 2-1. MVME2300SC Front Panel Controls 2-1
Table 3-1. Debugger Commands 3-4
Table 3-2. Diagnostic Test Groups 3-8
Table 4-1. MVME2300SC Features 4-1
Table 4-2. Power Requirements 4-4
Table 4-3. PPC604-Bus-to-DRAM Timing — 60ns Page Devices 4-7
Table 4-4. PPC604-to-DRAM Timing — 50ns EDO Devices 4-8
Table 4-5. PowerPC604-Bus-to-Flash Timing — Bank B (16-bit Port) 4-11
Table 5-1. VMEbus Connector P1 Pin Assignments 5-2
Table 5-2. VMEbus Connector P2 Pin Assignment 5-3
Table 5-3. DEBUG (J2) Connector Pin Assignments 5-4
Table 5-4. 10/100 BASET (J3) Connector Pin Assignments 5-4
Table 5-5. CPU Debug (J1) Connector Pin Assignments 5-5
Table 5-6. J6 - PCI Expansion Connector (J6) Pin Assignments 5-10
Table 5-7. J11 - J12 PMC1 Connector Pin Assignments 5-13
Table 5-8. J13 - J14 PMC1 Connector Pin Assignments 5-14
Table 5-9. J21 and J22 PMC2 Connector Pin Assignments 5-16
Table 5-10. J23 and J24 PMC2 Connector Pin Assignments 5-17
Table A-1. MVME2300SC Specifications A-1
Table B-1. Troubleshooting MVME2300SC Modules B-1

About This Manual

MVME2300SC VME Processor Module Installation and Use provides information you will need to install and use your MVME2300SC VME processor module. It includes instructions for hardware preparation and installation; a board-level hardware overview; and firmware-related general information and startup instructions.

The MVME2300SC VME processor module is based on an MPC604 PowerPC microprocessor, and features dual PCI Mezzanine Card (PMC) slots with front panel and/or P2 I/O. As of the publication date, the information presented in this manual applies to the following MVME2300SC models:

Model	Processor	Memory
MVME2306SC-1	MPC604 @ 300 MHz	32MB ECC DRAM
MVME2307SC-1		64MB ECC DRAM

The MVME2300SC VME processor module is compatible with optional double-width or single-width PCI Mezzanine Cards (PMCs), and with the PMCspan PCI expansion mezzanine module. By utilizing the two onboard PMC slots and stacking PMCspan(s), the MVME2300SC provides support for up to six PMCs.

The information in this manual applies principally to the MVME2300SC module. The PMCspan and PMC modules are described briefly here but are documented in detail 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 C, *Related Documentation*.

This manual is intended for anyone who designs OEM systems, adds capability to an existing compatible system, or works in a lab environment for experimental purposes. A basic knowledge of computers and digital logic is assumed. To use this manual, you may also wish to become familiar with the publications listed in the *Related Documentation* section in Appendix C.

Summary of Changes

This is the third edition of *MVME2300SC Installation and Use*. It supersedes the April 1999 edition and incorporates the following updates.

Date	Description of Change
April 1999	Tables C-8 and C-10 were updated to supply corrected pinout information for PMC connectors J14 and J24.
March 2001	In the descriptions of the general-purpose software-readable header (J10) in Chapter 1 and 2, information on bit 1 (SRH1) has been updated to correctly reflect the functionality of that bit. In addition, the contents of the manual were reorganized in line with present Computer Group practice for board manuals.

Overview of Contents

Chapter 1, *Hardware Preparation and Installation*, provides unpacking instructions, hardware preparation guidelines, and installation instructions for the MVME2300SC VME processor module.

Chapter 2, *Startup and Operation*, provides information on powering up the MVME2300SC VME processor module after its installation in a system and describes the functionality of the switches, status indicators, and I/O ports.

Chapter 3, *PPCBug Firmware*, describes the basics of PPCBug and its architecture, describes the monitor (interactive command portion of the firmware) in detail, and gives information on using the debugger and special commands.

Chapter 4, *Functional Description*, describes the MVME2300SC VME processor module on a block diagram level.

Chapter 5, *Pin Assignments*, summarizes the pin assignments for the various groups of interconnect signals on the MVME2300SC.

Appendix A, *Specifications*, lists the general specifications for the MVME2300SC VME processor module. Subsequent sections of the appendix detail cooling requirements and EMC regulatory compliance.

Appendix B, *Troubleshooting*, includes simple troubleshooting steps to follow in the event that you have difficulty with your MVME2300SC VME processor module.

Appendix C, *Related Documentation*, lists all documentation related to the MVME2300SC.

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:

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**>

<**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**.

A character precedes a data or address parameter to specify the numeric format, as follows:

$	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.

Hardware Preparation and Installation

1

Getting Started

This chapter provides unpacking instructions, hardware preparation guidelines, and installation instructions for the MVME2300SC VME processor module. The section below supplies an overview of the preparation and startup process.

Overview of Installation Procedure

The following table lists the things you will need to do to use this board and tells where to find the information you need to perform each step. Be sure to read this entire chapter, including all Cautions and Warnings, before you begin.

Table 1-1. Startup Overview

What you need to do ...	Refer to ...	On page ...
Unpack the hardware.	***Guidelines for Unpacking***	1-4
Configure jumpers on the MVME2300SC module.	***Preparing the Board,*** MVME2300SC	1-6
Prepare the PMCs.	***Preparing the Board,*** PMCs	1-10
	For additional information on PMCs, refer to the PMC manuals provided with those cards.	
Prepare the PMCspan module(s), if necessary.	***Preparing the Board,*** PMCspan	1-10
	For more information on PMCspan expansion modules, refer to the ***PMCspan PMC Adapter Carrier Module Installation and Use*** manual, listed in Appendix C, ***Related Documentation.***	
Install the PMCs on the MVME2300SC module.	***Installing the Hardware,*** PMC Modules	1-11
	For additional information on PMCs, refer to the PMC manuals provided with those cards**.**	

Table 1-1. Startup Overview (Continued)

What you need to do ...	Refer to ...	On page ...
Install the primary PMCspan module (if used).	***Installing the Hardware,*** Primary PMCspan Module	1-13
	For additional information on PMCspan modules, refer to the ***PMCspan PMC Adapter Carrier Module Installation and Use*** manual, listed in Appendix C, ***Related Documentation.***	
Install the secondary PMCspan module (if used).	***Installing the Hardware,*** Secondary PMCspan Module	1-16
	For additional information on PMCspan modules, refer to the ***PMCspan PMC Adapter Carrier Module Installation and Use*** manual, listed in Appendix C, ***Related Documentation.***	
Install and connect the MVME2300SC module.	***Installing the Hardware,*** MVME2300SC	1-18
Connect a console terminal, if necessary.	***Installing the Hardware,*** System Console Terminal	1-20
	Installation Considerations	1-21
Connect any other optional devices or equipment you will be using.	***Connector Pin Assignments***	5-1
	For more information on optional devices and equipment, refer to the documentation provided with that equipment.	
Power up the system.	***Startup and Operation***	2-1
	Troubleshooting	B-1
Note that the debugger firmware initializes the MVME2300SC.	***PPCBug Firmware,*** Using PPCBug	3-3
	You may also wish to obtain the ***PPCBug Diagnostics Manual***, listed in Appendix C, ***Related Documentation***.	
Initialize the system clock.	***PPCBug Firmware,*** Debugger Commands	3-4

Table 1-1. Startup Overview (Continued)

What you need to do ...	Refer to ...	On page ...
Examine and/or change environmental parameters.	***ENV - Set Environment***	3-11
	You may also wish to obtain the ***PPCBug Firmware Package User's Manual***, listed in Appendix C, ***Related Documentation***.	
Program the MVME2300SC module and PMCs as needed for your applications.	***MVME2300 Series VME Processor Module Programmer's Reference Guide***, listed in Appendix C, ***Related Documentation***	
	For additional information on PMCs, refer to the PMC manuals provided with those cards.	

Equipment Required

The following equipment is necessary to complete an MVME2300SC system:

- ❏ VME system enclosure
- ❏ System console terminal
- ❏ Operating system (and / or application software)
- ❏ Disk drives (and / or other I/O) and controllers

Guidelines for Unpacking

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

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



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

ESD Precautions

This section applies to all hardware installations you may perform that involve the MVME2300SC board.

Motorola strongly recommends the use of an antistatic wrist strap and a conductive foam pad when you install or upgrade the board. Electronic components can be extremely sensitive to ESD. After removing the board from the chassis or from its protective wrapper, place the board flat on a grounded, static-free surface, component side up. Do not slide the board over any surface.

If no ESD station is available, you can avoid damage resulting from ESD by wearing an antistatic wrist strap (available at electronics stores). Place the strap around your wrist and attach the grounding end (usually a piece of copper foil or an alligator clip) to an electrical ground. An electrical ground can be a piece of metal that literally runs into the ground (such as an unpainted metal pipe) or a metal part of a grounded electrical appliance. An appliance is grounded if it has a three-prong plug and is plugged into a three-prong grounded outlet. You cannot use the chassis in which you are installing the MVME2300SC itself as a ground, because the enclosure is unplugged while you work on it.

Warning

Turn the system's power off before you perform these procedures. Failure to turn the power off before opening the enclosure can result in personal injury or damage to the equipment. Hazardous voltage, current, and energy levels are present in the chassis. Hazardous voltages may be present on power switch terminals even when the power switch is off. Never operate the system with the cover removed. Always replace the cover before powering up the system.

Preparing the Board

To produce the desired configuration and ensure proper operation of the MVME2300SC, you may need to reconfigure hardware to some extent before installing the module.

Most options on the MVME2300SC are under software control: By setting bits in control registers after installing the module in a system, you can modify its configuration. (The MVME2300SC registers are briefly described in Chapter 3 under *ENV – Set Environment*, with additional information in the *MVME2300 Series VME Processor Module Programmer's Reference Guide* as listed in Appendix C, *Related Documentation*.)

Some options, though, are not software-programmable. Such options are either set by configuration switches or are controlled through physical installation or removal of header jumpers or interface modules on the MVME2300SC or its associated modules.

MVME2300SC

Figure 1-1 illustrates the placement of the jumper headers, connectors, and various other components on the MVME2300SC. Manually configurable jumper headers on the MVME2300SC are listed in the following table (with default settings enclosed in brackets).

Table 1-2. MVME2300SC Jumper Settings

Jumper	Function	Settings	
J8	Flash bank selection	[1-2] 2-3	Flash bank A enabled (4MB, soldered). Flash bank B enabled (1MB, sockets).
J9	VMEbus system controller selection	No jumper 1-2 [2-3]	System controller. Not system controller. Automatic system controller.
J10	General-purpose readable jumper configuration	3-4 empty [3-4]	Firmware defaults (in Flash) selected. NVRAM selected. Other headers are user-definable. Factory configuration has all jumpers installed.

Notes Items in brackets are factory default settings.

J5 and J7 are for factory use only.

MVME2300SC boards are tested and shipped with the jumper configurations summarized in Table 1-2. The factory-installed debug monitor, the PPCBug firmware, operates with those factory settings.

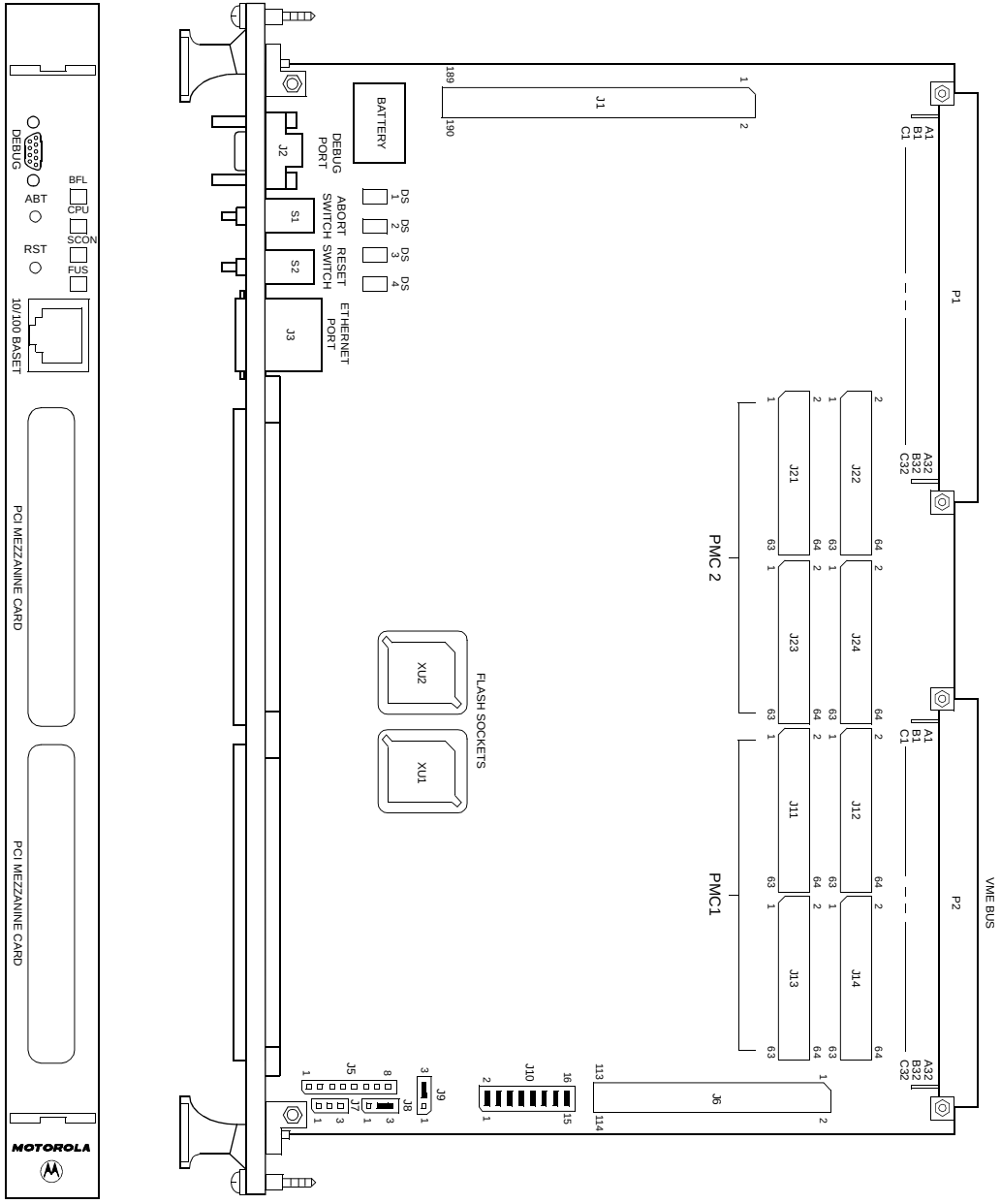


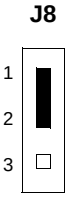
Figure 1-1. MVME2300SC Layout

Flash Bank Selection (J8)

The MVME2300SC VMEmodule has provision for 1MB of 8-bit Flash memory for the on-board firmware (or for customer-specific applications) in two 32-pin PLCC sockets that constitute Flash bank B.

In addition, the MVME2300SC accommodates 4MB of firmware resident on four soldered-in devices (Flash bank A) specifically for customer use. Bank A consists of four 16-bit Smart Voltage SMT devices populated with 8Mbit Flash devices (for a total of 4 MB).

Both banks contain the on-board firmware, PPCBug. The setting of a jumper header (J8 on the MVME2300SC) determines whether the board fetches its reset vector from bank A or from bank B. To enable Flash bank A, place a jumper across header J8 pins 1-2. To enable Flash bank B, place a jumper across header J8 pins 2-3. The factory configuration uses Flash bank B.



Flash Bank A Enabled (2MB/4MB, Soldered)



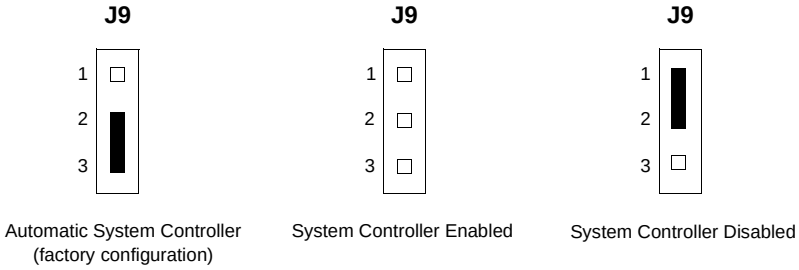
Flash Bank B Enabled (1MB, Sockets)
(factory configuration)

VMEbus System Controller (J9)

The MVME2300SC is factory-configured in ‘‘automatic’’ system controller mode (i.e., a jumper is installed across pins 2 and 3 of header J9). This means that at system power-up or reset, the MVME2300SC determines whether it is system controller by its position on the bus; if it occupies slot 1 on the VME system, it configures itself as the system controller.

Remove the jumper from J9 if you intend to operate the MVME2300SC as system controller in all cases.

Install the jumper across pins 1 and 2 if the MVME2300SC is not to operate as system controller under any circumstances.



Automatic System Controller (factory configuration) | System Controller Enabled | System Controller Disabled

General-Purpose Readable Jumpers (J10)

Header J10 provides eight software-readable jumpers. These jumpers can be read as a register at ISA I/O address $801 (hexadecimal). Bit 0 is associated with header pins 1-2; bit 7 is associated with pins 15-16. The bit values are read as a **0** when a jumper is installed, or as a **1** when the jumper is removed.

The PowerPC firmware, PPCBug, reserves all bits, SRH0 to SRH7. The MVME2300SC is shipped from the factory with J10 set to all **0**s (jumpers on all pins), as shown below and in Figure 1-2.

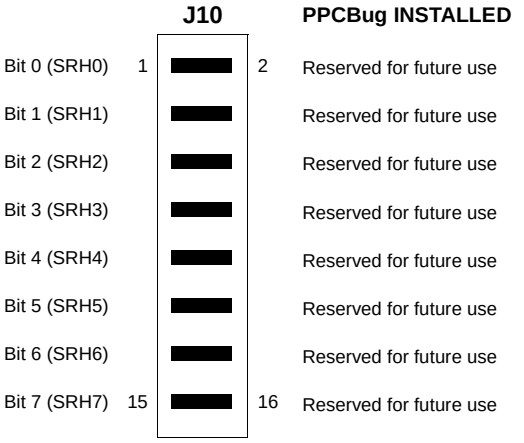


Figure 1-2. General-Purpose Software-Readable Header

PMCs

For a discussion of any configurable items on the PMCs (PCI Mezzanine Cards) that you are using, refer to the user's manual for the PMCs in question.

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 module on the MVME2300SC, you must first install the selected PMCs on the PMCspan module. Refer to the PMCspan *PMC Adapter Carrier Module Installation and Use* manual for instructions.

Installing the Hardware

This section covers:

- Installation of PMCs and PMCspan modules on the MVME2300SC
- Installation of the assembly in a VME chassis
- System considerations relevant to the installation

PMC Modules

PCI mezzanine card (PMC) modules mount on top of the MVME2300SC, and/or on a PMCspan module. Refer to Figure 1-3 and perform the following steps to install a PMC on your MVME2300SC module. This procedure assumes that you have read the user's manual that came with your PMCs.

1. 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 card cage.

Caution

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

Warning

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

2. If the MVME2300SC has already been installed in a VMEbus card slot, carefully remove it. Lay the MVME2300SC flat, with connectors P1 and P2 facing you.

Caution

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

3. Remove the PCI filler plate from the selected PMC slot in the front panel of the MVME2300SC. If you are installing a double-width PMC, remove the filler plates from both PMC slots.
4. Slide the front panel(s) of the PMC module into the front panel opening(s) from behind and place the PMC module on top of the MVME2300SC.

   The PMC front panel should fit snugly and the four connectors on the underside of the module should connect smoothly with the corresponding connectors (J11/12/13/14 or J21/22/23/24 for a single-width PMC, all eight for a double-width PMC) on the MVME2300SC.

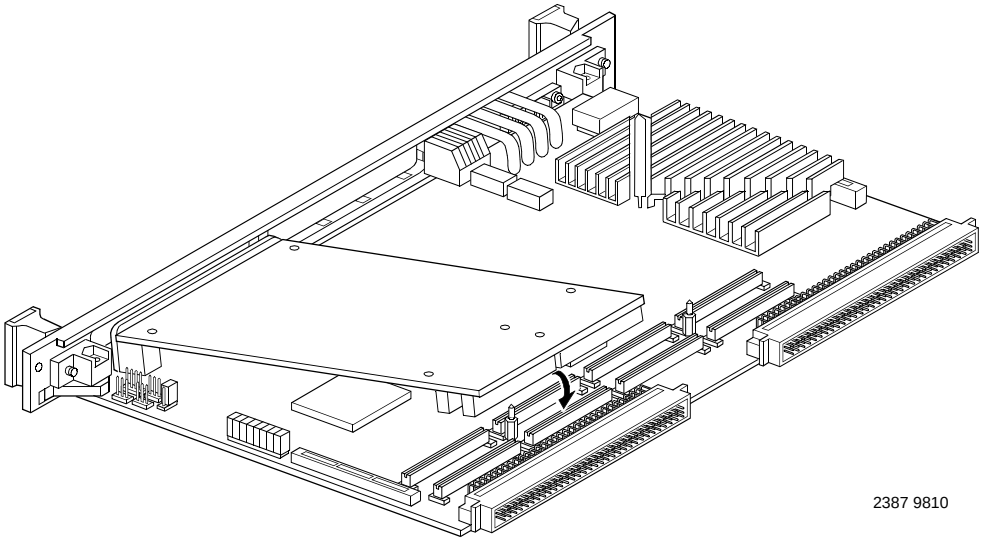


Figure 1-3. Typical Single-width PMC Module Placement on MVME2300SC

5. Insert the four short Phillips screws through the holes at the front and rear corners of the PMC module, into the standoffs on the MVME2300SC. Tighten the screws.
6. If you are installing two single-width PMCs, repeat the above procedure for the second PMC.

Primary PMCspan Module

To install a PMCspan-002 PCI expansion module on your MVME2300SC, refer to Figure 1-4 and perform the following steps. This procedure assumes that you have read the user's manual that was furnished with the PMCspan module, and that you have installed the selected PMCs on the PMCspan according to the instructions given in the PMCspan and PMC manuals.

1. 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 card cage.

Caution

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

Warning

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

2. If the MVME2300SC has already been installed in the chassis, carefully remove it from the VMEbus card slot and lay it flat, with connectors P1 and P2 facing you.

Caution

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

3. Attach the four standoffs to the MVME2300SC 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.

4. Place the PMCspan on top of the MVME2300SC module. Align the mounting holes in each corner to the standoffs, and align PMCspan connector P4 with MVME2300SC connector J6.
5. Gently press the PMCspan and MVME2300SC together, making sure that P4 is fully seated into J6.
6. Insert the four short Phillips screws through the holes at the corners of the PMCspan and into the standoffs on the MVME2300SC module. Tighten the screws.

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

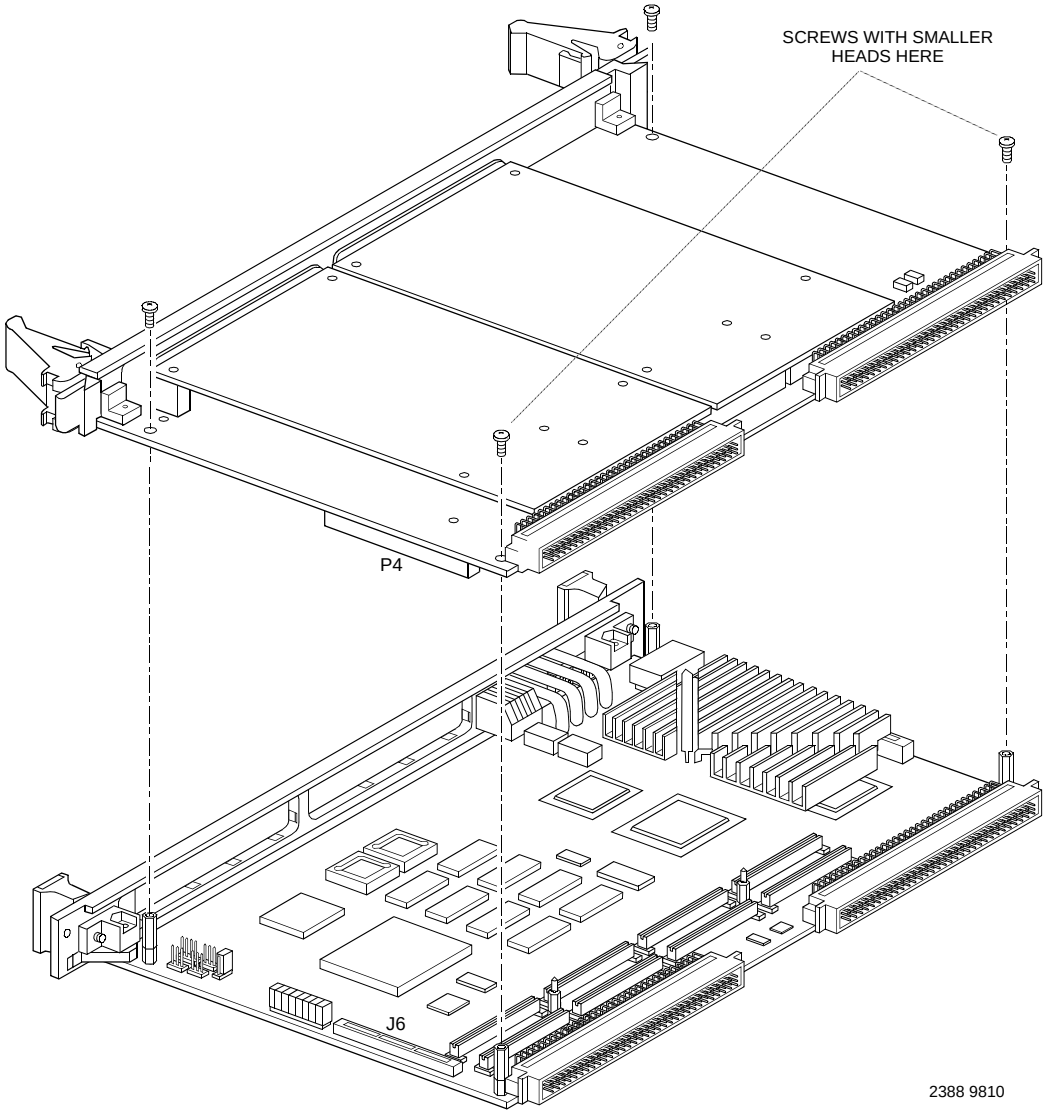


Figure 1-4. PMCspan-002 Installation on an MVME2300SC

Secondary PMCspan Module

The PMCspan-010 PCI expansion module mounts on top of a PMCspan-002 PCI expansion module. To install a PMCspan-010 on your MVME2300SC, refer to Figure 1-5 and perform the following steps. 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. 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 card cage.

Caution

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

Warning

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

2. If the Primary PMC Carrier Module/MVME2300SC assembly is already installed in the VME chassis, carefully remove the two-board assembly from the VMEbus card slots and lay it flat, with the P1 and P2 connectors facing you.

Caution

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

3. Remove the four short Phillips screws from the standoffs in each corner of the primary PCI expansion module, PMCspan-002.
4. Attach the four standoffs to the PMCspan-002.
5. 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.

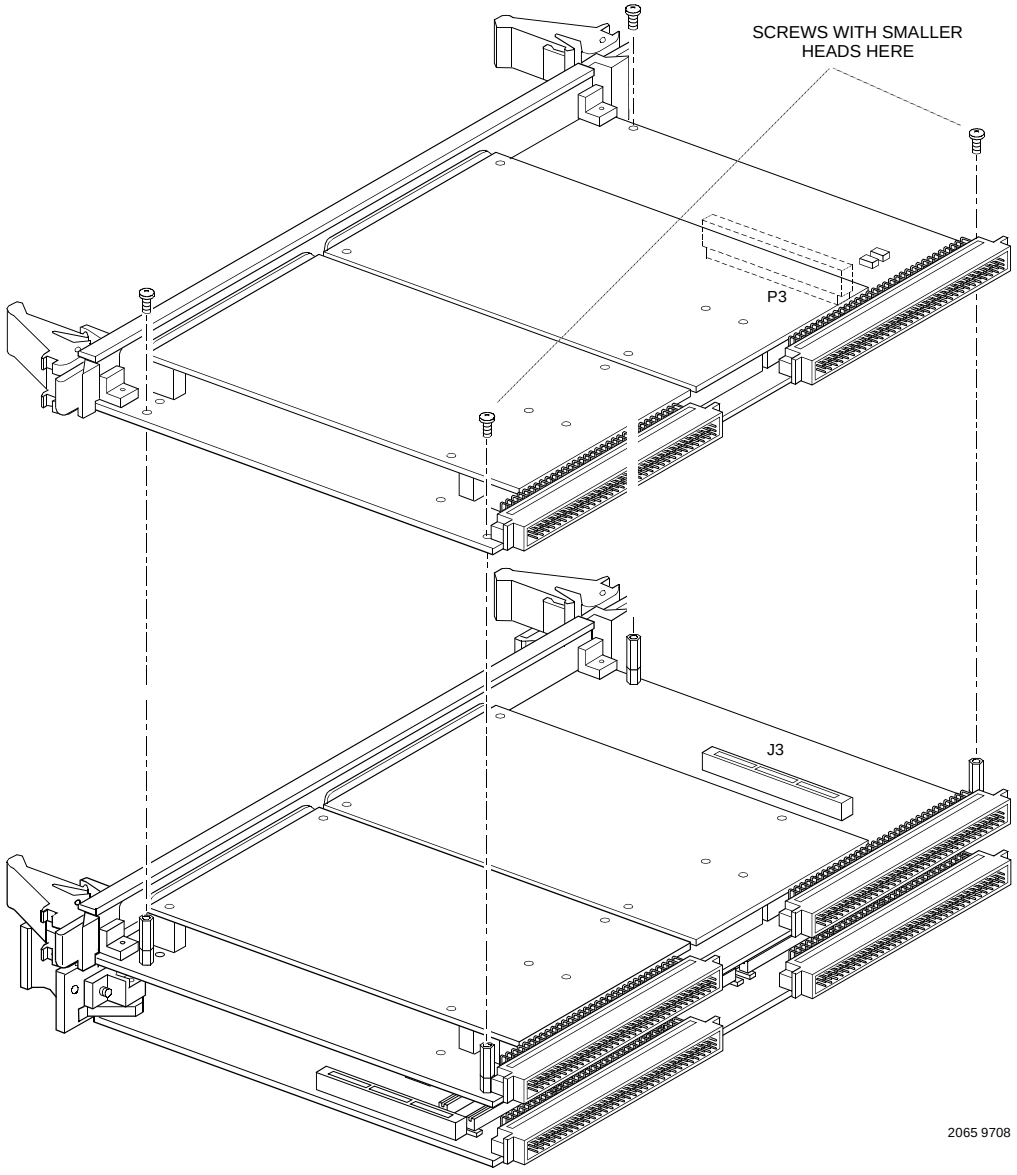


Figure 1-5. PMCspan-010 Installation on a PMCspan-002/MVME2300SC

6. Gently press the two PMCspan modules together, making sure that P3 is fully seated in J3.
7. 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.

MVME2300SC

Before installing the MVME2300SC into your VME chassis, ensure that the jumpers on the MVME2300SC J7, J8, J9, and J10 headers are configured, as previously described. This procedure assumes that you have already installed the PMCspan module(s) if desired, and any PMCs that you have selected.

Proceed as follows to install the MVME2300SC in the VME chassis:

1. Attach an ESD strap to your wrist. Attach the other end of the ESD strap to an electrical ground (refer to *Installation Preliminaries*). The ESD strap must be secured to your wrist and to ground throughout the procedure.
   a. Ensure that the AC or DC power is switched off and remove the AC cord or DC power lines from the system.



Caution Inserting or removing modules while power is applied could result in damage to module components.



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

2. Remove chassis or system cover(s) as instructed in the user's manual for the equipment.
3. Remove the filler panel from the card slot where you are going to install the MVME2300SC. If you have installed one or more

PMCspan PCI expansion modules onto your MVME2300SC, you will need to remove filler panels from one additional card slot for each PMCspan module, above the card slot for the MVME2300SC.

- If you intend to use the MVME2300SC as system controller, it must occupy the leftmost 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 MVME2300SC as system controller, it can occupy any unused card slot.



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

4. Slide the MVME2300SC (together with the PMCspan modules 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 MVME2300SC (and PMCspan modules 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 autojumpering 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 MVME2300SC.

Note For slots occupied by PMCspan modules (if used), the IACK and BG jumpers may be left in or taken out as you wish. The P1 connector on PMCspan modules passes those signals through.

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 MVME2300SC.

   In normal operation, the host CPU controls MVME2300SC 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 MVME2300SC's green **CPU** LED indicates activity as a set of confidence tests is run, and the debugger prompt `PPC1-Bug>` appears.

System Console Terminal

To ready the system console terminal for use:

1. Confirm that jumpers were installed on all bits on header J10 of the MVME2300SC board as shown in Figure 1-2. This is necessary when the PPCBug firmware is used.

2. Connect the terminal via a cable to the DB9 **DEBUG** connector, J2. See Table 5-3 for pin signal assignments. Set the terminal up as follows:
   - Eight bits per character
   - One stop bit per character
   - Parity disabled (no parity)
   - Baud rate of 9600 baud

9600 baud is the power-up default for the serial port on the MVME2300SC. After power-up you can reset the baud rate if you wish, using the PPCBug firmware's **PF** (Port Format) command via the command line interface. Whatever the baud rate, some type of hardware handshaking — either XON/XOFF or via the RTS/CTS line — is desirable if the system supports it.

Installation Considerations

The MVME2300SC draws power from VMEbus backplane connectors P1 and P2. P2 is also used for the upper 16 bits of data in 32-bit transfers, and for the upper 8 address lines in extended addressing mode. The MVME2300SC may not function properly without its main board connected to VMEbus backplane connectors P1 and P2.

In addition, the MVME2300SC routes 32 pins of I/O from PMC slot 1 and 32 pins of I/O from PMC slot 2 to a set of 32 shared pins on P2. This I/O scheme gives both PMC slots access (via P2) to an SCSA (Signal Computing System Architecture) backplane bus, if the system supports one.

Whether the MVME2300SC operates as a VMEbus master or as a VMEbus slave, it is configured for 32 bits of address and 32 bits of data (A32/D32). However, it handles A16 or A24 devices in the address ranges indicated in Chapter 4. D8 and/or D16 devices in the system must be handled by the PowerPC processor software. For elaboration, refer to the memory maps in Chapter 4.

The MVME2300SC contains shared onboard DRAM whose base address is software-selectable. Both the onboard processor and off-board VMEbus devices see this local DRAM at base physical address $00000000, as programmed by the PPCBug firmware. This may be changed via software to any other base address. Refer to the MVME2300SC *Programmer's Reference Guide* for more information.

If the MVME2300SC tries to access off-board resources in a nonexistent location and is not system controller, and if the system does not have a global bus timeout, the MVME2300SC waits indefinitely for the VMEbus cycle to complete. This will cause the system to lock up. There is only one situation in which the system might lack this global bus timeout: when the MVME2300SC is not the system controller and there is no global bus timeout elsewhere in the system.

Multiple MVME2300SC boards may be installed in a single VME chassis. Each must have a unique Universe address, selected as described in the ''VMEbus Interface'' chapter of the *Universe User Manual*. In general, hardware multiprocessor features are supported.

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

one MVME2300SC processor to broadcast a signal to any other MVME2300 series processors. All eight registers are accessible from any local processor as well as from the VMEbus.

Startup and Operation 2

Introduction

This chapter provides information on powering up the MVME2300SC VME processor module after its installation in a system, and describes the functionality of the switches, status indicators, and I/O ports on the front panels of the MVME2300SC and PMCspan modules.

For programming information, consult the *MVME2300 Series VME Processor Module Programmer's Reference Guide* as listed in Appendix C, *Related Documentation.*

Switches and LEDs

There are two switches (**ABT** and **RST**) and four LED (light-emitting diode) status indicators (**BFL**, **CPU**, **SCON** and **FUS**) located on the MVME2300SC front panel.

Table 2-1. MVME2300SC Front Panel Controls

Control/Indicator	Function
Abort Switch (**ABT**)	Sends an interrupt signal to the processor. The interrupt is normally used to abort program execution and return control to the debugger firmware located in the MVME2300SC Flash memory. The interrupter connected to the **ABT** switch is an edge-sensitive circuit, filtered to remove switch bounce.
Reset Switch (**RST**)	Resets all onboard devices. Also asserts the HRESET∗ line in the MPC604 and drives a SYSRESET∗ signal if the MVME2300SC is system controller. SYSRESET∗ signals 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. For further details, refer to Chapter 4, *Functional Description.*

Table 2-1. MVME2300SC Front Panel Controls

Control/Indicator	Function
BFL LED (yellow)	Board failure. Lights when the BRDFAIL* signal line is active.
CPU LED (green)	CPU activity. Lights when the DBB* (Data Bus Busy) signal line on the processor bus is active.
SCON LED (green)	System controller. Lights when the MVME2300SC is functioning as VMEbus system controller.
FUS LED (green)	Fuse OK. Indicates that +5Vdc and +12Vdc power is available on the board. (–12Vdc is not fused.)

Initial Conditions

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. Applying power to the system (as well as resetting it) triggers an initialization of the MVME2300SC's MPU, hardware, and firmware along with the rest of the system.

The Flash-resident firmware initializes the devices on the MVME2300SC board in preparation for booting the operating system. The firmware is shipped from the factory with a set of defaults appropriate to the board. In most cases there is no need to modify the firmware configuration before you boot the operating system. For specifics in this regard, refer to Chapter 3 and to the user documentation for the PPCBug firmware.

Applying Power

When you power up (or when you reset) the system, the firmware executes some self-checks and proceeds to the hardware initialization. The system startup flows in a predetermined sequence, following the hierarchy inherent in the processor and the MVME2300SC hardware. The figure below charts the flow of the basic initialization sequence that takes place during system startup.

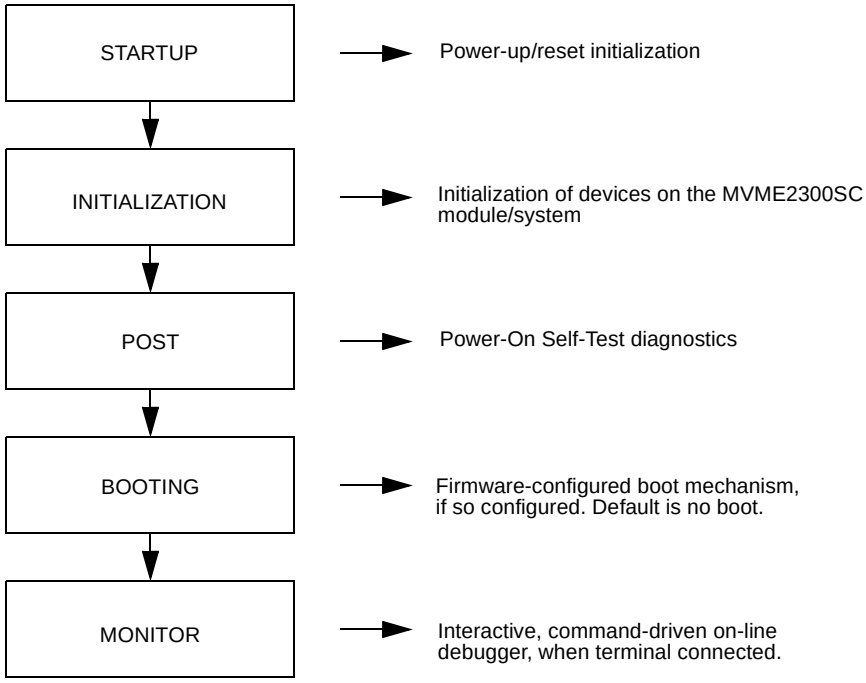


Figure 2-1. MVME2300SC/Firmware System Startup

2

Pre-Startup Checklist

Before you power up the MVME2300SC system, be sure that the following conditions exist:

1. Jumpers and/or configuration switches on the MVME2300SC VME processor module and associated equipment are set as required for your particular application.
2. The MVME2300SC board is installed and cabled up as appropriate for your particular chassis or system, as outlined in Chapter 1.
3. The terminal that you plan to use as the system console is connected to the console port (serial port 1) on the MVME2300SC module.
4. The terminal is set up as follows:
   - Eight bits per character
   - One stop bit per character
   - Parity disabled (no parity protection)
   - Baud rate 9600 baud (the default baud rate of many serial ports at power-up)
5. Any other device that you wish to use, such as a host computer system and/or peripheral equipment, is cabled to the appropriate connectors.

After you complete the checks listed above, you are ready to power up the system.

Bringing up the Board

The MVME2300SC comes with PPCBug firmware installed. For the firmware to operate properly with the board, you must follow the steps below.



Caution

Inserting or removing boards with power applied may damage board components.

Turn all equipment power OFF. Refer to *Preparing the Board* on page 1-5 and verify that jumpers and switches are configured as necessary for your particular application.

1. Header J8 determines whether the on-board firmware, PPCBug, initializes from Flash memory bank A (4MB, soldered) or from Flash bank B (1MB, socketed). The factory configuration has the jumper installed on J8 pins 1-2, enabling Flash bank B. Verify that this setting is appropriate for your application.

2. Verify that header J9 is configured as appropriate for the desired system controller functionality (always system controller, never system controller, or self-regulating) on the MVME2300SC.

3. Configuration header J10 on the MVME2300SC contains eight software-readable jumper segments, which all affect the operation of the firmware. They are read as a register at ISA I/O address $801. (The *MVME2300 Series VME Processor Module Programmer's Reference Guide* supplies information on register accesses.) The bit values are read as a **0** when a jumper is installed on the corresponding header segment, or as a **1** when the jumper is removed from that segment.

   The default configuration for the board has J10 set to all **0**s (jumpers installed on all pins). The PPCBug firmware reserves/defines all bits, SRH0 to SRH7.

4. Verify that the settings of any configuration switches or jumpers that are present on the PMCs you have installed are appropriate for your application.

5. Refer to the setup procedure for your particular chassis or system for details concerning the installation of the MVME2300SC and the implementation of the SCbus.

6. Confirm that the terminal to be used as the PPCBug system console is connected to the DB9 **DEBUG** connector, J2, on the front panel of the MVME2300SC. (Other connection options may exist, but they depend on the nature of your overall system configuration.) The console terminal should be set up as follows:
   - Eight bits per character
   - One stop bit per character
   - Parity disabled (no parity)
   - Baud rate 9600 baud (the power-up default)

   After power-up, you can reconfigure the baud rate of the debug port by using the PPCBug Port Format (**PF**) command.

Note Whatever the baud rate, some form of hardware handshaking — either XON/XOFF or via the RTS/CST line — is desirable if the system supports it. If you get garbled messages and missing characters, you should check the terminal to make sure that handshaking is enabled.

7. Verify that the remaining equipment in your system is installed and configured as appropriate.

8. Power up the system. PPCBug executes some self-checks and displays the debugger prompt `PPC-Bug>` if the firmware is in Board mode.

   However, if the **ENV** command has placed PPCBug in System mode, the system performs a self-test and tries to autoboot. Refer to the **ENV** and **MENU** command descriptions (*Modifying the Environment* on page 3-9).

   If the confidence test fails, the test is aborted when the first fault is encountered. If possible, an appropriate message is displayed, and control then returns to the menu.

9. Before using the MVME2300SC after the initial installation, set the date and time using the following command line structure:

   PPC-Bug> **SET** [*mmddyyhhmm*]|[<+/-*CAL*>;**C**]

   For example, the following command line starts the real-time clock and sets the date and time to 10:37 a.m., November 7, 2001:

   PPC-Bug> **SET 1107011037**

   The board's self-tests and operating systems require that the real-time clock be running.

Autoboot

Autoboot is a software routine included in the PPCBug Flash/EPROM to provide an independent mechanism for booting operating systems. The autoboot routine automatically scans for controllers and devices in a specified sequence until a valid bootable device containing a boot media is found or the list is exhausted. If a valid bootable device is found, a boot from that device is started. The controller scanning sequence goes from the lowest controller Logical Unit Number (LUN) detected to the highest LUN detected.

At power-up, Autoboot is enabled and (provided that the drive and controller numbers encountered are valid) the following message is displayed upon the system console:

```
Autoboot in progress... To abort hit <BREAK>
```

A delay follows this message so that you can abort the Autoboot process if you wish. Then the actual I/O begins: the program designated within the volume ID of the media specified is loaded into RAM and control passes to it. If you want to gain control without Autoboot during this time, however, you can press the <BREAK> key or use the **ABORT** or **RESET** switches on the front panel.

The Autoboot process is controlled by parameters contained in the **ENV** command. These parameters allow the selection of specific boot devices and files, and allow programming of the Boot delay. Refer to the **ENV** command description in Chapter 3 for more details.



Although you can use streaming tape to autoboot, the same power supply must be connected to the tape drive, the controller, and the MVME2300SC. At power-up, the tape controller will position the streaming tape to the load point where the volume ID can correctly be read and used.

However, if the MVME2300SC loses power but the controller does not, and the tape happens to be at load point, the necessary command sequences (Attach and Rewind) cannot be given to the controller and the autoboot will not succeed.

ROMboot

The ROMboot function is configured/enabled via the **ENV** command (refer to Chapter 3) and is executed at power-up (optionally also at reset).

You can also execute the ROMboot function via the **RB** command, assuming there is valid code in the memory devices (or optionally elsewhere on the board or VMEbus) to support it. If ROMboot code is installed, a user-written routine is given control (if the routine meets the format requirements).

One use of ROMboot might be resetting the SYSFAIL∗ line on an unintelligent controller module. The **NORB** command disables the function.

For a user's ROMboot module to gain control through the ROMboot linkage, four conditions must exist:

- Power has just been applied (but the **ENV** command can change this to also respond to any reset).
- Your routine is located within the MVME2300SC Flash memory map (but the **ENV** command can change this to any other portion of the onboard memory, or even offboard VMEbus memory).
- The ASCII string "BOOT" is found in the specified memory range.
- Your routine passes a checksum test, which ensures that this routine was really intended to receive control at powerup.

Network Boot

Network Auto Boot is a software routine in the PPCBug firmware which provides a mechanism for booting an operating system using a network (local Ethernet interface) as the boot device. The Network Auto Boot routine automatically scans for controllers and devices in a specified sequence until a valid bootable device containing boot media is found or until the list is exhausted. If a valid bootable device is found, a boot from that device is started. The controller scanning sequence goes from the lowest controller Logical Unit Number (LUN) detected to the highest LUN detected.

At power-up, Network Boot is enabled and (provided that the drive and controller numbers encountered are valid) the following message is displayed upon the system console:

```
Network Boot in progress... To abort hit <BREAK>
```

After this message, there is a delay to let you abort the Auto Boot process if you wish. Then the actual I/O is begun: the program designated within the volume ID of the media specified is loaded into RAM and control passes to it. If you want to gain control without Network Boot during this time, however, you can press the <BREAK> key or use the software **ABORT** or **RESET** switches.

Network Auto Boot is controlled by parameters contained in the **NIOT** and **ENV** commands. These parameters allow the selection of specific boot devices, systems, and files, and allow programming of the Boot delay. Refer to the **ENV** command description in Chapter 3 for more details.

Restarting the System

You can initialize the system to a known state in three different ways: Reset, Abort, and Break. Each method has characteristics which make it more suitable than the others in certain situations.

A special debugger function is accessible during resets. This feature instructs the debugger to use the default setup/operation parameters in ROM instead of your own setup/operation parameters in NVRAM. To activate this function, you press the **RESET** and **ABORT** switches at the

same time. This feature can be helpful in the event that your setup/operation parameters are corrupted or do not meet a sanity check. Refer to the **ENV** command description in Chapter 3 for the ROM defaults.

Reset

Powering up the MVME2300SC initiates a system reset. You can also initiate a reset by pressing and quickly releasing the **RESET** switch on the MVME2300SC front panel, or reset the board in software.

For details on resetting the MVME2300SC board through software, refer to the *MVME2300 Series VME Processor Module Programmer's Reference Guide*.

Both "cold" and "warm" reset modes are available. By default, PPCBug is in "cold" mode. During *cold* resets, a total system initialization takes place, as if the MVME2300SC had just been powered up. All static variables (including disk device and controller parameters) are restored to their default states. The breakpoint table and offset registers are cleared. The target registers are invalidated. Input and output character queues are cleared. Onboard devices (timer, serial ports, etc.) are reset, and the two serial ports are reconfigured to their default state.

During *warm* resets, the PPCBug variables and tables are preserved, as well as the target state registers and breakpoints.

Note that when the MVME2300SC comes up in a cold reset, PPCBug runs in Board mode. Using the Environment (**ENV**) or **MENU** commands can make PPCBug run in System mode. Refer to Chapter 3 for specifics.

You will need to reset your system if the processor ever halts, or if the PPCBug environment is ever lost (vector table is destroyed, stack corrupted, etc.).

Abort

Aborts are invoked by pressing and releasing the **ABORT** switch on the MVME2300SC front panel. When you invoke an abort while executing a user program (running target code), a snapshot of the processor state is stored in the target registers. This characteristic makes aborts most appropriate for terminating user programs that are being debugged.

If a program gets caught in a loop, for instance, aborts should be used to regain control. The target PC, register contents, etc., help to pinpoint the malfunction.

Pressing and releasing the **ABORT** switch generates a local board condition which may interrupt the processor if enabled. The target registers, reflecting the machine state at the time the **ABORT** switch was pressed, are displayed on the screen. Any breakpoints installed in your code are removed and the breakpoint table remains intact. Control returns to the debugger.

Break

Pressing and releasing the <BREAK> key on the terminal keyboard generates a ‘‘power break’’. Breaks do not produce interrupts. The only time that breaks are recognized is while characters are being sent or received by the console port. A break removes any breakpoints in your code and keeps the breakpoint table intact. If the function was entered using SYSCALL, Break also takes a snapshot of the machine state. This machine state is then accessible to you for diagnostic purposes.

In many cases, you may wish to terminate a debugger command before its completion (for example, during the display of a large block of memory). Break allows you to terminate the command.

Diagnostic Facilities

The PPCBug package includes a set of hardware diagnostics for testing and troubleshooting the MVME2300SC. To use the diagnostics, switch directories to the diagnostic directory.

If you are in the debugger directory, you can switch to the diagnostic directory with the debugger command Switch Directories (**SD**). The diagnostic prompt `PPC-Diag>` appears. Refer to the *PPCBug Diagnostics Manual* for complete descriptions of the diagnostic routines available and instructions on how to invoke them. Note that some diagnostics depend on restart defaults that are set up only in a particular restart mode. The documentation for such diagnostics includes restart information.

PPCBug Firmware 3

Overview

The PPCBug firmware is the layer of software just above the hardware. The firmware supplies the appropriate initialization for devices on the MVME2300SC board 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 using the debugger and special commands. A 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 under *Related Documentation.*

PPCBug Basics

The PowerPC debug firmware, PPCBug, is a powerful evaluation and debugging tool for systems built around the Motorola PowerPC microcomputers. Facilities are available for loading and executing user programs under complete operator control for system evaluation. PPCBug provides a high degree of functionality, user friendliness, portability, and ease of maintenance.

PPCBug includes:

- Commands for display and modification of memory
- Breakpoint and tracing capabilities
- A powerful assembler/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 MVME2300SC 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 `PPC1-Bug>` is displayed and you have all of the debugger commands at your disposal.
- If you are in the diagnostic directory, the diagnostic prompt `PPC1-Diag>` is displayed and you have all of the diagnostic commands at your disposal as well as all of the debugger commands.

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

PPCBug Implementation

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

Memory Requirements

PPCBug requires a maximum of 512KB of read/write memory (i.e., DRAM). The debugger allocates this space from the top of memory. For example, a system containing 64MB ($04000000) of read/write memory will place the PPCBug memory page at locations $03F80000 to $03FFFFFF.

Using PPCBug

PPCBug is command-driven; it performs its various operations in response to commands that you enter at the keyboard. When the `PPC1-Bug` prompt appears on the screen, the debugger is ready to accept debugger commands. When the `PPC1-Diag` prompt appears on the screen, the debugger is ready to accept diagnostics commands.

To switch from one mode to the other, enter **SD** (Switch Directories). To examine the commands in the directory that you are currently in, use the Help command (**HE**).

What you key in is stored in an internal buffer. Execution begins only after you press the Return or Enter key. This allows you to correct entry errors, if necessary, with the control characters described in the *PPCBug Firmware Package User's Manual*, Chapter 1.

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

For example, if a breakpoint has been specified, then control returns to the debugger when the breakpoint is encountered during execution of the user program. Alternatively, 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 the 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 PPCBug debugger commands are summarized in the following table. The commands are described in detail in the *PPCBug Firmware Package User's Manual.*

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

Table 3-1. Debugger Commands

Command	Description
AS	One Line Assembler
BC	Block of Memory Compare
BF	Block of Memory Fill
BI	Block of Memory Initialize
BM	Block of Memory Move
BR	Breakpoint Insert
NOBR	Breakpoint Delete
BS	Block of Memory Search

Table 3-1. Debugger Commands (Continued)

Command	Description
BV	Block of Memory Verify
CM	Concurrent Mode
NOCM	No Concurrent Mode
CNFG	Configure Board Information Block
CS	Checksum
CSAR	PCI Configuration Space READ Access
CSAW	PCI Configuration Space WRITE Access
DC	Data Conversion
DMA	Block of Memory Move
DS	One Line Disassembler
DU	Dump S-Records
ECHO	Echo String
ENV	Set Environment
FORK	Fork Idle MPU at Address
FORKWR	Fork Idle MPU with Registers
GD	Go Direct (Ignore Breakpoints)
GEVBOOT	Global Environment Variable Boot
GEVDEL	Global Environment Variable Delete
GEVDUMP	Global Environment Variable(s) Dump
GEVEDIT	Global Environment Variable Edit
GEVINIT	Global Environment Variable Initialization
GEVSHOW	Global Environment Variable(s) Display
GN	Go to Next Instruction
GO	Go Execute User Program
GT	Go to Temporary Breakpoint
HE	Help
IDLE	Idle Master MPU
IOC	I/O Control for Disk
IOI	I/O Inquiry

Table 3-1. Debugger Commands (Continued)

Command	Description
IOP	I/O Physical (Direct Disk Access)
IOT	I/O Teach for Configuring Disk Controller
IRD	Idle MPU Register Display
IRM	Idle MPU Register Modify
IRS	Idle MPU Register Set
LO	Load S-Records from Host
MA	Macro Define/Display
NOMA	Macro Delete
MAE	Macro Edit
MAL	Enable Macro Listing
NOMAL	Disable Macro Listing
MAR	Load Macros
MAW	Save Macros
MD, MDS	Memory Display
MENU	System Menu
MM	Memory Modify
MMD	Memory Map Diagnostic
MS	Memory Set
MW	Memory Write
NAB	Automatic Network Boot
NAP	Nap MPU
NBH	Network Boot Operating System, Halt
NBO	Network Boot Operating System
NIOC	Network I/O Control
NIOP	Network I/O Physical
NIOT	Network I/O Teach (Configuration)
NPING	Network Ping
OF	Offset Registers Display/Modify
PA	Printer Attach

Table 3-1. Debugger Commands (Continued)

Command	Description
NOPA	Printer Detach
PBOOT	Bootstrap Operating System
PF	Port Format
NOPF	Port Detach
PFLASH	Program FLASH Memory
PS	Put RTC into Power Save Mode
RB	ROMboot Enable
NORB	ROMboot Disable
RD	Register Display
REMOTE	Remote
RESET	Cold/Warm Reset
RL	Read Loop
RM	Register Modify
RS	Register Set
RUN	MPU Execution/Status
SD	Switch Directories
SET	Set Time and Date
SROM	SROM Examine/Modify
SYM	Symbol Table Attach
NOSYM	Symbol Table Detach
SYMS	Symbol Table Display/Search
T	Trace
TA	Terminal Attach
TIME	Display Time and Date
TM	Transparent Mode
TT	Trace to Temporary Breakpoint
VE	Verify S-Records Against Memory
VER	Revision/Version Display
WL	Write Loop

Caution

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

Note, however, that Flash bank A and Flash bank B both contain the PPCBug debugger.

Diagnostic Tests

The PPCBug hardware diagnostics are intended for testing and troubleshooting the MVME2300SC module.

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

If you are in the debugger directory, the debugger prompt `PPC1-Bug>` displays, and all of the debugger commands are available. Diagnostics commands cannot be entered at the `PPC1-Bug>` prompt.

If you are in the diagnostic directory, the diagnostic prompt `PPC1-Diag>` displays, and all of the debugger and diagnostic commands are available.

PPCBug's diagnostic test groups are listed in Table 3-2. Note that not all tests are performed on the MVME2300SC. 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 3-2. Diagnostic Test Groups

Command	Description
CL1283	Parallel Interface (CL1283) Tests*
DEC	DEC21x43 Ethernet Controller Tests
ISABRDGE	PCI/ISA Bridge Tests
KBD8730x	PC8730x Keyboard/Mouse Tests*
L2CACHE	Level 2 Cache Tests*

Table 3-2. Diagnostic Test Groups (Continued)

Command	Description
NCR	NCR 53C8xx SCSI-2 I/O Processor Tests*
PAR8730x	Parallel Interface (PC8730x) Test*
UART	Serial Input/Output Tests
PCIBUS	PCI/PMC Generic Tests
RAM	Local RAM Tests
RTC	M48Txxx Timekeeping Tests
SCC	Serial Communications Controller (Z85C230) Tests*
VGA543x	Video Diagnostics Tests*
VME2	VMEchip2 VME Interface ASIC Tests*
Z8536	Z8536 Counter/Timer Tests*

Notes You may enter command names in either uppercase or lowercase characters.

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 MVME2300SC.

Modifying the Environment

You can use the factory-installed debug monitor, PPCBug, to modify certain parameters contained in the MVME2300SC's Non-Volatile RAM (NVRAM), also known as Battery Backed-Up RAM (BBRAM).

- The Board Information Block in NVRAM contains various entries that define operating parameters of the board 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 supplementary information on **CNFG** and **ENV** that is specific to the PPCBug firmware, along with the parameters that you can modify with the **ENV** command.

CNFG - Configure Board Information Block

Use this command to display and configure the Board Information Block which resides within the NVRAM. The board information block contains various elements that correspond to specific operational parameters of the MVME2300SC board.

The board structure for MVME2300SCs resembles that shown in the following example:

```
PPC-Bug>cnfg
Board (PWA) Serial Number = "MOT00xxxxxxx"
Board Identifier = "MVME2307SC"
Artwork (PWA) Identifier = "01-W3430FxxB"
MPU Clock Speed = "300"
Ethernet Address = 0001AF26A464
Local SCSI Identifier = "07"
System Serial Number = "nnnnnnn"
System Identifier = "Motorola MVME2300SC"
License Identifier = "nnnnnnnn"
PPC-Bug>
```

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 zeros 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 *MVME2300 Series VME Processor Module Programmer's Reference Guide* for the actual location and other information about the Board Information Block. Refer to the *PPCBug Firmware Package User's Manual* for a **CNFG** description 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 *MVME2300 Series VME Processor Module Programmer's Reference Guide*.

Listed and described below are the parameters that you can configure using **ENV**. The default values shown are those that were in effect when this document was published.

Note In the event of difficulty with the MVME2300SC, you may wish to use **env;d** <CR> to restore the factory defaults as a troubleshooting aid (see Appendix B).

Configuring the PPCBug Parameters

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

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

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

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

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

Y Display the field service menu.

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

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

The Remote Start Method Switch is used when the MVME2300SC is cross-loaded from another VME-based CPU, to start execution of the cross-loaded program.

G Use the Global Control and Status Register to pass and start execution of the cross-loaded program. *This selection is not applicable to the MVME2300SC boards.*

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

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

N Do not use any Remote Start Method.

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

Y Accesses will be made to the appropriate system buses (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 = 04000000?
```

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 = 50?
```

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

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

This is the value programmed into the“ROMFAL” field (Memory Control Configuration Register 8: bits 23-27) to indicate the number of clock cycles used in accessing the ROM. The lowest allowable ROMFAL setting is $00; the highest allowable is $1F. The value to enter depends on processor speed; refer to Chapter 1 or Appendix A for appropriate values. The default value varies according to the system’s bus clock speed.

Note ROM First Access Length is not applicable to the MVME2300SC. The configured value is ignored by PPCBug.

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

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

Note ROM Next Access Length is not applicable to the MVME2300SC. The configured value is ignored by PPCBug.

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

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

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

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

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

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

Initializes the PIRQ*x* (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. For details on PCI/ISA interrupt assignments and for suggested values to enter for this parameter, refer to the *8259 Interrupts* section of Chapter 5 in the *MVME2300 Series VME Processor Module Programmer's Reference Guide*.

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

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

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

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

The list of LED/serial codes is included in the section on *MPU, Hardware, and Firmware Initialization* in 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 MVME2300SC modules. To perform this configuration, you should have a working knowledge of the Universe ASIC as described in your *MVME2300 Series VME Processor Module Programmer's Reference Guide*.

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

Y	Set up and enable the VMEbus Interface. (Default)
N	Do not set up or enable the VMEbus Interface.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

`Special PCI Slave Image Register = 00000000?`

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

`Master Control Register = 80C00000?`

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

`Miscellaneous Control Register = 52060000?`

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

`User AM Codes = 00000000?`

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

Functional Description 4

Introduction

This chapter describes the MVME2300SC VME processor module on a block diagram level. The *Summary of Features* provides an overview of the MVME2300SC, followed by a detailed description of several blocks of circuitry. Figure 4-1 shows a block diagram of the overall board architecture.

Detailed descriptions of other MVME2300SC blocks, including programmable registers in the ASICs and peripheral chips, can be found in the *MVME2300 Series VME Processor Module Programmer's Reference Guide* (part number V2300A/PG). Refer to that manual for a functional description of the MVME2300SC in greater depth.

Summary of Features

The following table summarizes the features of the MVME2300SC VME processor module.

Table 4-1. MVME2300SC Features

Feature	Description
Microprocessor	300 MHZ MPC604 PowerPC® processor
Form factor	6U VMEbus
ECC DRAM	Two-way interleaved, ECC-protected 32MB or 64MB
Flash memory	Bank B: Two 32-pin PLCC sockets populated with 1MB 8-bit Flash devices
	Bank A: Four 16-bit Smart Voltage SMT devices populated with 8Mbit Flash devices (4MB)
Real-time clock	8KB NVRAM with RTC, battery backup, and watchdog function (SGS-Thomson M48T559)
Switches	Reset (**RST**) and Abort (**ABT**)

Table 4-1. MVME2300SC Features (Continued)

Feature	Description
Status LEDs	Four: Board Fail (**BFL**), **CPU**, System Controller (**SCON**), Fuses (**FUS**)
Timers	One 16-bit timer in W83C553 PCI/ISA bridge; four 32-bit timers in Raven (MPIC) device
	Watchdog timer provided in SGS-Thomson M48T559
Interrupts	Software interrupt handling via Raven (PCI/MPU bridge) and Winbond (PCI/ISA bridge) controllers
VME I/O	VMEbus P2 connector
Serial I/O	One asynchronous debug port via DB9 connector on front panel, also via P2 and transition module
Ethernet I/O	10Base-T/100Base-TX connections via RJ45 connector on front panel; AUI connections via P2 and transition module
PCI interface	Two IEEE P1386.1 PCI Mezzanine Card (PMC) slots for one double-width or two single-width PMCs
	Front panel and/or VMEbus P2 I/O on both PMC slots
	One 114-pin Mictor connector for optional PMCspan expansion module
SCSA I/O	Connections from both PMC slots to SCSA backplane TDM bus (if present in system) via shared pins on P2 connector
VMEbus interface	VMEbus system controller functions
	VMEbus-to-local-bus interface (A24/A32, D8/D16/D32/block transfer [D8/D16/D32/D64])
	Local-bus-to-VMEbus interface (A16/A24/A32, D8/D16/D32)
	VMEbus interrupter
	VMEbus interrupt handler
	Global Control/Status Register (GCSR) for interprocessor communications
	DMA for fast local memory/VMEbus transfers (A16/A24/A32, D16/D32/D64)

General Description

The MVME2300SC is a VME processor module equipped with a PowerPC® 604 microprocessor.

As described in the *Features* section, the MVME2300SC offers many standard features desirable in a computer system — among them Ethernet and debug ports, Boot ROM, Flash memory, DRAM, and an interface for two PCI Mezzanine Cards (PMCs) — all contained in a one-slot VME package. Its flexible mezzanine architecture allows relatively easy upgrades of the I/O.

There are four standard buses on the MVME2300SC:

PowerPC Processor Bus	ISA Bus
PCI Local Bus	VMEbus

In addition, the MVME2300SC supplies connectivity from both PMC slots to an SCSA (Signal Computing System Architecture) backplane TDM bus, if the system supports one, via shared pins on VME connector P2.

As shown in Figure 4-1, a Raven PCI Bridge ASIC provides the interface from the Processor Bus to PCI. A W83C553 PCI/ISA Bridge (PIB) Controller device performs the bridge function between PCI and ISA. The Universe ASIC device provides the interface between the PCI Local Bus and the VMEbus. A Falcon chipset is the ECC memory controller.

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

MPC604 Processor

The MVME2300SC is built with a PowerPC 604 processor chip. It can be ordered with 32MB or 64MB of ECC DRAM, and up to 5MB of Flash memory.

The PowerPC 604 is a 64-bit processor with 32 KB on-chip caches (32KB data cache and 32KB instruction cache).

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

The power requirements for the MVME2300SC are shown in Table 4-2.

Table 4-2. Power Requirements

Configuration	+5V Power	+12V and –12V Power
300MHz PowerPC 604	4.5A typical 5.5A maximum	PMC-dependent (Refer to Appendix B)

I/O Implementation

The MVME2300SC offers many standard features desirable in a computer system — among them Ethernet and debug ports, Boot ROM, Flash memory, DRAM, and an interface for two PCI Mezzanine Cards (PMCs) — all contained in a one-slot VME package. Its flexible mezzanine architecture allows relatively easy I/O upgrades.

There are four standard buses on the MVME2300SC:

PowerPC Processor Bus	ISA Bus
PCI Local Bus	VMEbus

In addition, the MVME2300SC supplies connectivity from both PMC slots to an SCSA (Signal Computing System Architecture) backplane TDM bus, if the system supports one, via shared pins on VME connector P2.

As shown in Figure 4-1, a Raven PCI Bridge ASIC provides the interface from the Processor Bus to PCI. A W83C553 PCI/ISA Bridge (PIB) Controller device performs the bridge function between PCI and ISA. The Universe ASIC device provides the interface between the PCI Local Bus and the VMEbus. A Falcon chipset is the ECC memory controller.

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

ASICs

The following ASICs are used on the MVME2300SC:

- **Universe ASIC** (VMEbus interface). Provides the PCI-bus-to-VMEbus interface, the VMEbus-to-PCI-bus interface, and the DMA controller functions of the local VMEbus.

  Processor-to-VMEbus transfers are D8, D16, or D32. VMEchip2 DMA transfers to the VMEbus, however, are D16, D32, D16/BLT, D32/BLT, or D64/MBLT.

- **Raven ASIC**. Provides the bridge function between the MPC604 processor bus and the PCI Local Bus. It provides 32-bit addressing and 64-bit data.

The block diagram in Figure 4-1 on page 4-6 illustrates the MVME2300SC's overall architecture.

DRAM Memory

Current models of the MVME2300SC have either 32MB or 64MB of DRAM.

The DRAM blocks are controlled by the Falcon chipset, which performs two-way interleaving and provides single- and double-bit error correction. ECC is calculated over a span of 72 bits.

There are one or two blocks of DRAMs that provide 32MB or 64MB of ECC DRAM. The DRAM blocks consists of 9 devices each. Either 1Mx16 (Page) 50-pin TSOPII DRAM or 4Mx16 (EDO) 50-pin TSOPII DRAM are used to provide 32/64MB. When populated, these blocks appears as Block A and Block B to the Falcon chipset.

Refer to the *MVME2300 Series VME Processor Module Programmer's Reference Guide* for additional information and programming details.

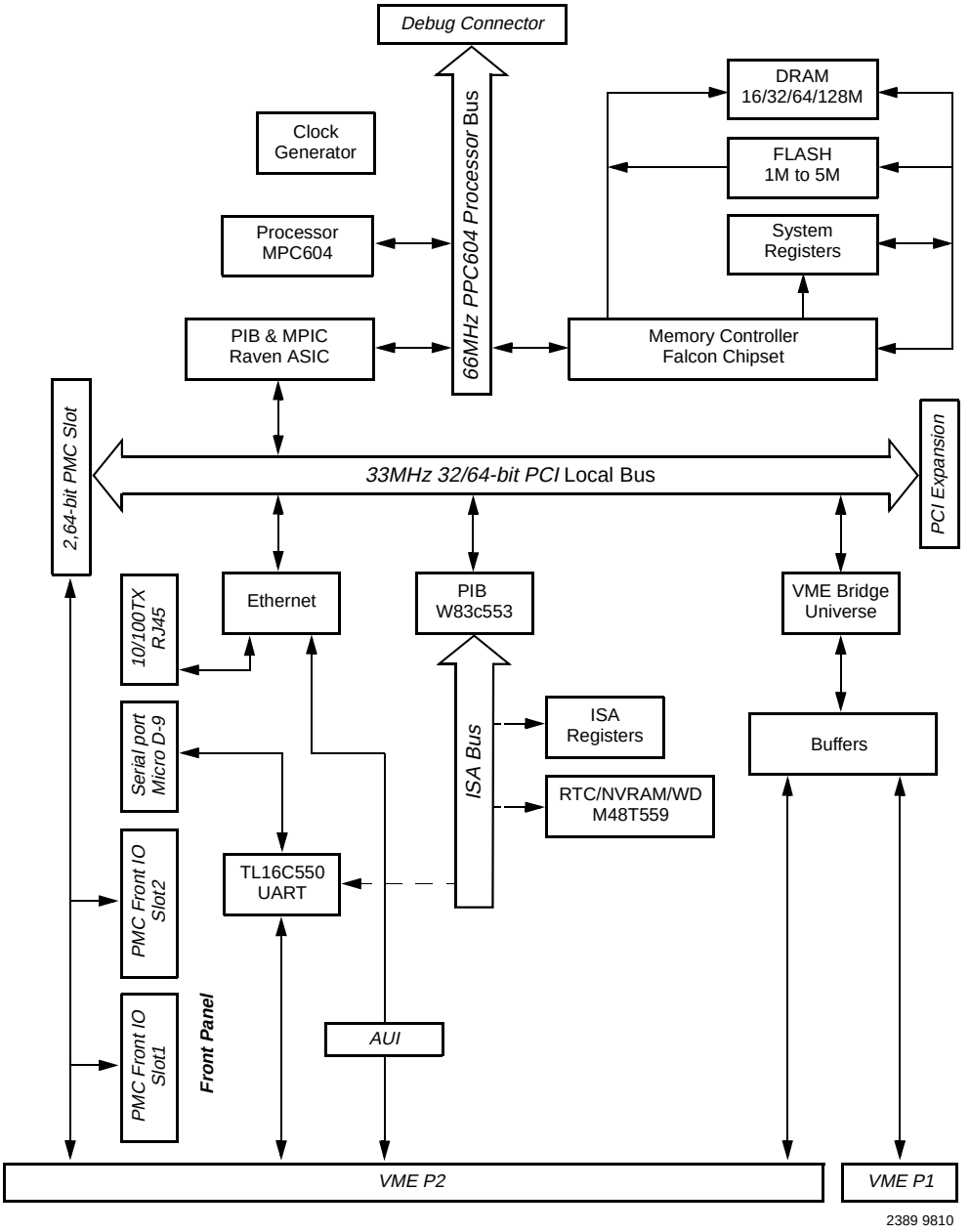


Figure 4-1. MVME2300SC Block Diagram

The block diagram for the memory interface is shown in the following figure.

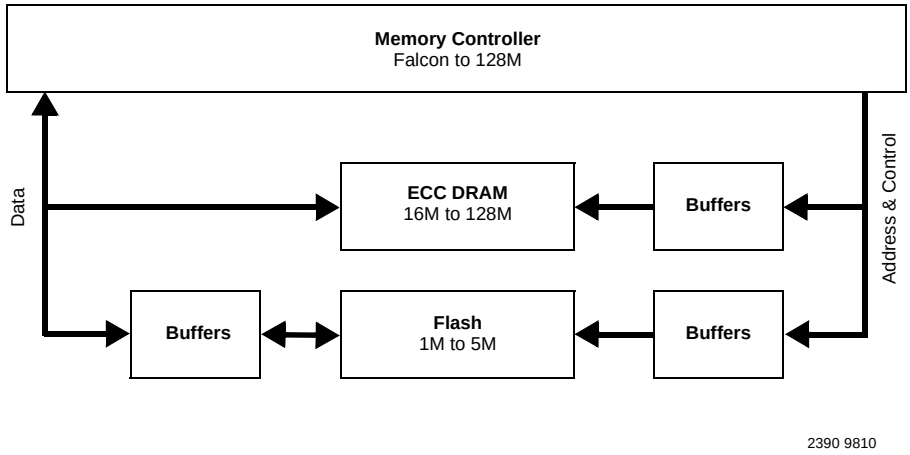


Figure 4-2. Memory Block Diagram

DRAM Latency

The ECC memory access latency times for 60ns, fast page DRAMs are shown in the following table.

Table 4-3. PPC604-Bus-to-DRAM Timing — 60ns Page Devices

Access Type	Clock Periods Required for:				Total Clocks
	1st Beat	2nd Beat	3rd Beat	4th Beat	
4-Beat Read after Idle (Quad-word aligned)	9	1	2	1	13
4-Beat Read after Idle (Quad-word misaligned)	9	3	1	1	14
4-Beat Read after 4-Beat Read (Quad-word aligned)	7/3 [1]	1	2	1	11/7

Table 4-3. PPC604-Bus-to-DRAM Timing — 60ns Page Devices (Continued)

Access Type	Clock Periods Required for:				Total Clocks
	1st Beat	2nd Beat	3rd Beat	4th Beat	
4-Beat Read after 4-Beat Read (misaligned)	6/2 [1]	3	1	1	11/7
4-Beat Write after Idle	4	1	1	1	7
4-Beat Write after 4-Beat Write (Quad-word aligned)	7/3 [1]	1	1	1	10/6
1-Beat Read after Idle	9	-	-	-	9
1-Beat Read after 1-Beat Read	8/6 [1]	-	-	-	8/6
1-Beat Write after Idle	4	-	-	-	4
1-Beat Write after 1-Beat Write	12/10 [1]	-	-	-	12/10

Notes 1. These numbers assume that the PowerPC 604 bus master is doing address pipelining with TS∗ occurring at the minimum time after AACK∗ is asserted. Also the two numbers shown in the 1st beat column are for page miss/page hit.

2. In some cases, the numbers shown are averages and specific instances may be longer or shorter.

If all blocks of DRAMs are 50ns, EDO devices then the latency times for the ECC memory would be as follows:

Table 4-4. PPC604-to-DRAM Timing — 50ns EDO Devices

Access Type	Clock Periods Required for:				Total Clocks
	1st Beat	2nd Beat	3rd Beat	4th Beat	
4-Beat Read after Idle (Quad-word aligned)	8	1	1	1	11

Table 4-4. PPC604-to-DRAM Timing — 50ns EDO Devices (Continued)

Access Type	Clock Periods Required for:				Total Clocks
	1st Beat	2nd Beat	3rd Beat	4th Beat	
4-Beat Read after Idle (Quad-word misaligned)	8	2	1	1	12
4-Beat Read after 4-Beat Read (Quad-word aligned)	5/2 [1]	1	1	1	8/5
4-Beat Read after 4-Beat Read (misaligned)	4/2 [1]	2	1	1	8/6
4-Beat Write after Idle	4	1	1	1	7
4-Beat Write after 4-Beat Write (Quad-word aligned)	4/3 [1]	1	1	1	7/6
1-Beat Read after Idle	8	-	-	-	8
1-Beat Read after 1-Beat Read	7/5 [1]	-	-	-	7/5
1-Beat Write after Idle	4	-	-	-	4
1-Beat Write after 1-Beat Write	9/7 [1]	-	-	-	9/7

Notes 1. These numbers assume that the PowerPC 604 bus master is doing address pipelining with TS∗ occurring at the minimum time after AACK∗ is asserted. Also the two numbers shown in the 1st beat column are for page miss/page hit.

2. In some cases, the numbers shown are averages and specific instances may be longer or shorter.

Flash Memory

The MVME2300SC base board has provision for up to 5MB of Flash memory.

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

Bank A consists of four 16-bit Smart Voltage SMT devices populated with 8Mbit Flash devices (for a total of 4MB). Only 32-bit writes are supported for this bank of Flash. Both banks contain the on-board firmware, PPCBug. The address of the reset vector, 0xFFF00100, is jumper-selectable. A jumper must be installed either between J8 pins 1 and 2 for Bank A, or between J8 pins 2 and 3 for Bank B (the factory configuration). The Falcon chipset maps 0xFFF00100 to Bank A or B depending on the jumper setting.

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

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

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

Flash Latency

There is one 16-bit port bank of Flash on the MVME2300SC. The access times for this bank are shown in the following table.

Table 4-5. PowerPC604-Bus-to-Flash Timing — Bank B (16-bit Port)

Access type	Clock Periods Required for:				Total Clocks
	1st Beat	2nd Beat	3rd Beat	4th Beat	
4-Beat Read	68	64	64	64	260
4-Beat Write	N/A	N/A	N/A	N/A	N/A
1-Beat Read (2 bytes to 8 bytes)	68	-	-	-	68
1-Beat Read (1 byte)	20	-	-	-	20
1-Beat Write	19	-	-	-	19

Ethernet Interface

The MVME2300SC module uses Digital Equipment's DECchip 21143 PCI Fast Ethernet LAN controller to implement an Ethernet interface that supports 10Base-T/100Base-TX connections, via an RJ45 connector on the front panel. Ethernet AUI signals are routed to the P2 connector. The balanced differential transceiver lines are coupled via on-board transformers.

Every MVME2300SC is assigned an Ethernet station address. The address is $0001AF2*xxxxx*, where *xxxxx* is the unique 5-nibble number assigned to the board (i.e., every board has a different value for *xxxxx*).

Each MVME2300SC displays its Ethernet station address on a label attached to the base board in the PMC connector keepout area just behind the front panel. In addition, the six bytes including the Ethernet station address are stored in the NVRAM (BBRAM) configuration area specified by boot ROM. That is, the value 0001AF2*xxxxx* is stored in NVRAM. The MVME2300SC debugger, PPCBug, has the capability to retrieve the Ethernet station address via the **CNFG** command.

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

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

For the pin assignments of the 10Base-T/100Base-TX connector, refer to Chapter 5.

At the physical layer, the Ethernet interface bandwidth is 10Mbit/second for 10Base T. For the 100Base TX, it is 100Mbit/second. Refer to the description of the NVRAM/RTC and Watchdog Timer registers in the *MVME2300 Series VME Processor Module Programmer's Reference Guide* for detailed programming information.

PCI Mezzanine Card (PMC) Interface

A key feature of the MVME2300SC family is the PCI bus. In addition to the on-board local bus devices (Ethernet, etc.), the PCI bus supports an industry-standard mezzanine interface, IEEE P1386.1 PCI Mezzanine Card (PMC).

PMC modules offer a variety of possibilities for I/O expansion such as FDDI (Fiber Distributed Data Interface), ATM (Asynchronous Transfer Mode), graphics, Ethernet, or SCSI ports. For a complete listing of available PMCs, go to the GroupIPC World Wide Web site at URL
The MVME2300SC supports PMC
front panel and rear P2 I/O. There is also provision for stacking one or two PMC carrier boards, or PMCspan PCI expansion modules, on the MVME2300SC for additional expansion.

The MVME2300SC supports two PMC slots. Two sets of four 64-pin connectors on the base board (J11 - J14, and J21 - J24) interface with 32-bit/64-bit IEEE P1386.1 PMC-compatible mezzanines to add any desirable function.

Refer to Chapter 5 for the pin assignments of the PMC connectors. For detailed programming information, refer to the PCI bus descriptions in the *MVME2300 Series VME Processor Module Programmer's Reference Guide* and to the user documentation for the PMC modules you intend to use.

PMC Slot 1 (Single-Width PMC)

PMC slot 1 has the following characteristics:

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

For P2 I/O configurations, pins 1 through 32 of J14 are routed to row C and row A of P2.

PMC Slot 2 (Single-Width PMC)

PMC slot 2 has the following characteristics:

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

For P2 I/O configurations, pins 1 through 32 of J24 (as with J14) are routed to row C and row A of P2.

PMC Slots 1 and 2 (Double-Width PMC)

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

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

PCI Expansion

The PMCspan expansion module connector, J6, is a 114-pin Mictor connector. It is located near P2 on the primary side of the MVME2300SC. Its interrupt lines are routed to the Raven MPIC.

VMEbus Interface

The VMEbus interface is implemented with the CA91C042 Universe ASIC. The Universe chip interfaces the 32/64-bit PCI local bus to the VMEbus.

The Universe ASIC provides:

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

The Universe chip includes Universe Control and Status Registers (UCSRs) for interprocessor communications. It can provide the VMEbus system controller functions as well. For detailed programming information, refer to the *Universe User's Manual* and to the discussions in the *MVME2300 Series VME Processor Module Programmer's Reference Guide*.

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

The MVME2300SC interfaces to the VMEbus via the P1 and P2 backplane connectors, which use three-row 96-pin connectors as specified in the VMEbus standard. It also draws +5V, +12V, and –12V power from the VMEbus backplane through these two connectors. 3.3V and 2.5V supplies are regulated onboard from the +5 power.

Asynchronous Debug Port

A Texas Instruments TL16C550 Universal Asynchronous Receiver/Transmitter (UART) provides the asynchronous debug port. TTL-level signals for the port are routed through appropriate EIA-232-D drivers and receivers to a DB9 connector on the front panel, and to the P2 backplane connector. The external signals are ESD protected.

This serial port can support 19.2 KBaud I/O. For detailed programming information, refer to the *MVME2300 Series VME Processor Module Programmer's Reference Guide* and to the vendor documentation for the UART device.

PCI/ISA Bridge (PIB) Controller

The MVME2300SC uses a Winbond W83C553 PCI/ISA Bridge (PIB) Controller to supply the interface between the PCI local bus and the ISA system I/O bus (diagrammed in Figure 4-1).

The PIB controller provides the following functions:

- PCI bus arbitration for:
  - ISA (Industry Standard Architecture) bus DMA (not functional on the MVME2300SC)
  - The PHB (PCI Host Bridge) MPU/local bus interface function, implemented by the Raven ASIC
  - All on-board PCI devices

  - The PMC slot

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

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

Real-Time Clock/NVRAM/Timer Function

The MVME2300SC employs an SGS-Thomson surface-mount M48T559 RAM and clock chip to provide 8KB of non-volatile static RAM, a real-time clock, and a watchdog timer function. This chip supplies a clock, oscillator, crystal, power failure detection, memory write protection, 8KB of NVRAM, and a battery in a package consisting of two parts:

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

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

The clock furnishes seconds, minutes, hours, day, date, month, and year in BCD 24-hour format. Corrections for 28-, 29- (leap year), and 30-day months are made automatically. The clock generates no interrupts. Although the M48T559 is an 8-bit device, 8-, 16-, and 32-bit accesses from the ISA bus to the M48T559 are supported. Refer to the *MVME2300 Series*

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

PCI Host Bridge

The Raven ASIC provides the bridge function between the MPC604 processor bus and the PCI Local Bus. It provides 32-bit addressing and 64-bit data. 64-bit addressing (dual address cycle) is not supported. The Raven supports various PowerPC processor external bus frequencies up to 66MHz and PCI frequencies up to 33MHz.

There are four programmable map decoders for each direction to provide flexible address mappings between the MPC and the PCI Local Bus. Refer to the *MVME2300 Series VME Processor Module Programmer's Reference Guide* for additional information and programming details.

Interrupt Controller (MPIC)

The Raven ASIC provides an MPIC Interrupt Controller to handle various interrupt sources. The interrupt sources are:

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

Programmable Timers

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

Interval Timers

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

- Counter 0 is associated with interrupt request line IRQ0. It can be used for system timing functions, such as a timer interrupt for a time-of-day function.
- Counter 1 generates a refresh request signal for ISA memory. This timer is not used in the MVME2300SC.
- Counter 2 provides the tone for the speaker output function on the PIB controller (the SPEAKER_OUT signal which can be cabled to an external speaker via the remote reset connector). This function is not used on the MVME2300SC.

The interval timers use the OSC clock input as their clock source. The MVME2300SC drives the OSC pin with a 14.31818MHz clock source.

16/32-Bit Timers

The MVME2300SC has one 16-bit timer and four 32-bit timers. The 16-bit timer is provided by the PIB. Raven device provides the four 32-bit timers that may be used for system timing or for generation of periodic interrupts. For information on programming these timers, refer to the data sheet for the W83C553 PIB controller and to the *MVME2300 Series VME Processor Module Programmer's Reference Guide*.

Connectors

The MVME2300SC interfaces to the VMEbus via P1 and P2, which are implemented with the three-row 96-pin connectors specified in the IEEE P1014-1987 VMEbus specification. The board also draws +5V, +12V, and –12V power from the VMEbus backplane through these two connectors. (The 3.3V supply is regulated on-board from the +5V power.)

Front panel connectors on the MVME2300SC include an RJ45 connector for the Ethernet 10BaseT/100BaseTX interface, and a 9-pin DB9 connector for the asynchronous serial debug port.

The MVME2300SC is equipped with two IEEE 1386.1 PCI Mezzanine Card (PMC) slots. The PMC slots are 64-bit capable and support both front and rear I/O. Pins 1-32 of PMC slot 1 connector J14 are routed to rows C and A of the 3-row P2 connector. Pins 1-32 of PMC slot 2 connector J24 are likewise routed to rows C and A of connector P2.

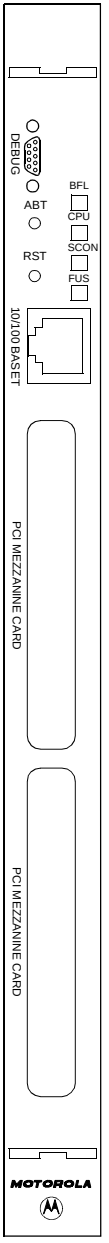
Additional PCI expansion is supported with a 114-pin Mictor connector (J6). This connection permits the stacking of one or two PMCspan dual-PMC carrier boards on the MVME2300SC for increased I/O capability. Each PMCspan board that you add requires an additional VME slot.

Pin assignments for the connectors on the MVME2300SC are listed in Chapter 5.

Status Indicators

The MVME2300SC front panel has four LED (light-emitting diode) status indicators:

- **BFL** (DS1, yellow). Board failure; lights when the BRDFAIL∗ signal line is active.
- **CPU** (DS2, green). CPU activity; lights when the DBB∗ (Data Bus Busy) signal line on the processor bus is active.
- **SCON** (DS3, green). System controller; lights when the MVME2300SC is functioning as VMEbus system controller.
- **FUS** (DS4, green). Fuses OK; indicates that +5Vdc and +12Vdc power is available on the board (–12Vdc is not fused). The fuses are resettable polyswitches.



10/100 BASET Port

The RJ45 port labeled **10/100 BASET** on the MVME2300SC front panel supplies the Ethernet LAN 10BaseT/100BaseTX interface, implemented with a DEC 21143 controller chip.

Note Ethernet AUI signals are routed to the P2 connector.

DEBUG Port

The DB9 port labeled **DEBUG** on the front panel of the MVME2300SC supplies the serial communications interface, implemented via a TL16C550 UART controller chip from Texas Instruments. It is asynchronous only. This serial port is configured for EIA-232-D DTE.

The serial I/O signals are also available via P2 and a transition module.

Pin Assignments

5

Connector Pin Assignments

This chapter summarizes the pin assignments for the following groups of interconnect signals on the MVME2300SC:

Connector		Location	Table
VMEbus connector		P1	Table 5-1
VMEbus connector, P2 I/O		P2	Table 5-2
Debug serial port, DB9		DEBUG (J2)	Table 5-3
Ethernet port, RJ45		10/100 BASET (J3)	Table 5-4
CPU debug connector		J1	Table 5-5
PCI expansion connector		J18	Table 5-6
PMC connectors, Slot 1	32-bit PCI	J11, J12	Table 5-7
	64-bit PCI extension and P2 I/O	J13, J14	Table 5-8
PMC connectors, Slot 2	32-bit PCI	J21, J22	Table 5-9
	64-bit PCI extension and P2 I/O	J23, J24	Table 5-10

The tables in this chapter furnish pin assignments only. For detailed descriptions of the interconnect signals, consult the support information for the MVME2300SC (available through your Motorola sales office).

VMEbus Connectors (P1, P2)

Two three-row 96-pin DIN type connectors, P1 and P2, supply the interface between the base board and the VMEbus. P1 provides power and VME signals for 24-bit addressing and 16-bit data. Its pin assignments are set by the IEEE P1014-1987 VMEbus Specification. P2 rows A and C provide power and interface signals for the transition module and SCSA backplane, if present. P2 Row C supplies the base board with power, with the upper 8 VMEbus address lines, and with an additional 16 VMEbus data lines. The pin assignments are listed in the following two tables.

Table 5-1. VMEbus Connector P1 Pin Assignments

	Row A	Row B	Row C	
1	VD0	VBBSY*	VD8	1
2	VD1	VBCLR*	VD9	2
3	VD2	VACFAIL*	VD10	3
4	VD3	VBGIN0*	VD11	4
5	VD4	VBGOUT0*	VD12	5
6	VD5	VBGIN1*	VD13	6
7	VD6	VBGOUT1*	VD14	7
8	VD7	VBGIN2*	VD15	8
9	GND	VBGOUT2*	GND	9
10	VSYSCLK	VBGIN3*	VSYSFAIL*	10
11	GND	VBGOUT3*	VBERR*	11
12	VDS1*	VBR0*	VSYSRESET*	12
13	VDS0*	VBR1*	VLWORD*	13
14	VWRITE*	VBR2*	VAM5	14
15	GND	VBR3*	VA23	15
16	VDTACK*	VAM0	VA22	16
17	GND	VAM1	VA21	17
18	VAS*	VAM2	VA20	18
19	GND	VAM3	VA19	19
20	VIACK*	GND	VA18	20
21	VIACKIN*	Not Used	VA17	21
22	VIACKOUT*	Not Used	VA16	22
23	VAM4	GND	VA15	23
24	VA7	VIRQ7*	VA14	24
25	VA6	VIRQ6*	VA13	25
26	VA5	VIRQ5*	VA12	26
27	VA4	VIRQ4*	VA11	27
28	VA3	VIRQ3*	VA10	28
29	VA2	VIRQ2*	VA9	29
30	VA1	VIRQ1*	VA8	30
31	–12V	Not Used	+12V	31
32	+5V	+5V	+5V	32

Table 5-2. VMEbus Connector P2 Pin Assignment

	ROW A	ROW B	ROW C	
1	PMC1/2_2 (J14/J24-2)	+5V	PMC1/2_1 (J14/J24-1)	1
2	PMC1/2_4 (J14/J24-4)	GND	PMC1/2_3 (J14/J24-3)	2
3	PMC1/2_6 (J14/J24-6)	Not Used	PMC1/2_5 (J14/J24-5)	3
4	PMC1/2_8 (J14/J24-8)	VA24	PMC1/2_7 (J14/J24-7)	4
5	PMC1/2_10 (J14/J24-10)	VA25	PMC1/2_9 (J14/J24-9)	5
6	PMC1/2_12 (J14/J24-12)	VA26	PMC1/2_11 (J14/J24-11)	6
7	PMC1/2_14 (J14/J24-14)	VA27	PMC1/2_13 (J14/J24-13)	7
8	PMC1/2_16 (J14/J24-16)	VA28	PMC1/2_15 (J14/J24-15)	8
9	PMC1/2_18 (J14/J24-18)	VA29	PMC1/2_17 (J14/J24-17)	9
10	PMC1/2_20 (J14/J24-20)	VA30	PMC1/2_19 (J14/J24-19)	10
11	PMC1/2_22 (J14/J24-22)	VA31	PMC1/2_21 (J14/J24-21)	11
12	PMC1/2_24 (J14/J24-24)	GND	PMC1/2_23 (J14/J24-23)	12
13	PMC1/2_26 (J14/J24-26)	+5V	PMC1/2_25 (J14/J24-25)	13
14	PMC1/2_28 (J14/J24-28)	VD16	PMC1/2_27 (J14/J24-27)	14
15	PMC1/2_30 (J14/J24-30)	VD17	PMC1/2_29 (J14/J24-29)	15
16	PMC1/2_32 (J14/J24-32)	VD18	PMC1/2_31 (J14/J24-31)	16
17	Not Used	VD19	Not Used	17
18	Not Used	VD20	Not Used	18
19	+5VF	VD21	+12VF	19
20	GND	VD22	Not Used	20
21	Not Used	VD23	GND	21
22	Not Used	GND	AUI_RX+	22
23	Not Used	VD24	AUI_RX-	23
24	Not Used	VD25	AUI_TX+	24
25	RI_EXT	VD26	AUI_TX-	25
26	DTR_EXT	VD27	AUI_COL+	26
27	RTS_EXT	VD28	AUI_COL-	27
28	DSR_EXT	VD29	GND	28
29	DCD_EXT	VD30	Not Used	29
30	CTS_EXT	VD31	Not Used	30
31	RD_EXT	GND	Not Used	31
32	TD_EXT	+5V	Not Used	32

Serial Port Connector - DEBUG (J2)

A standard Micro D9 connector located on the front panel of the MVME2300SC provides the interface to the asynchronous serial debug port. The pin assignments for this connector are as follows:

Table 5-3. DEBUG (J2) Connector Pin Assignments

1	DCD_EXT
2	RD_EXT
3	TD_EXT
4	DTR_EXT
5	Ground
6	DSR_EXT
7	RTS_EXT
8	CTS_EXT
9	RT_EXT

Ethernet Connector - 10BASET (J3)

The 10BaseT/100BaseTx connector is an RJ45 connector located on the front panel of the MVME2300SC. The pin assignments for this connector are as follows:

Table 5-4. 10/100 BASET (J3) Connector Pin Assignments

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

CPU Debug Connector - J1

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

Table 5-5. CPU Debug (J1) Connector Pin Assignments

1	PA0		PA1	2
3	PA2		PA3	4
5	PA4		PA5	6
7	PA6		PA7	8
9	PA8		PA9	10
11	PA10		PA11	12
13	PA12		PA13	14
15	PA14		PA15	16
17	PA16		PA17	18
19	PA18	GND	PA19	20
21	PA20		PA21	22
23	PA22		PA23	24
25	PA24		PA25	26
27	PA26		PA27	28
29	PA28		PA29	30
31	PA30		PA31	32
33	APAR0		APAR1	34
35	APAR2		APAR3	36
37	APE*		RSRV*	38

Table 5-5. CPU Debug (J1) Connector Pin Assignments (Continued)

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

Table 5-5. CPU Debug (J1) Connector Pin Assignments (Continued)

77	PD38		PD39	78
79	PD40		PD41	80
81	PD42		PD43	82
83	PD44		PD45	84
85	PD46		PD47	86
87	PD48		PD49	88
89	PA50		PD51	90
91	PD52		PD53	92
93	PD54		PD55	94
95	PD56	GND	PD57	96
97	PD58		PD59	98
99	PD60		PD61	100
101	PD62		PD63	102
103	DPAR0		DPAR1	104
105	DPAR2		DPAR3	106
107	DPAR4		DPAR5	108
109	DPAR6		DPAR7	110
111				112
113	DPE*		DBDIS*	114

Table 5-5. CPU Debug (J1) Connector Pin Assignments (Continued)

115	TT0		TSIZ0	116
117	TT1		TSIZ1	118
119	TT2		TSIZ2	120
121	TT3		TC0	122
123	TT4		TC1	124
125	CI*		TC2	126
127	WT*		CSE0	128
129	GLOBAL*		CSE1	130
131	SHARED*		DBWO*	132
133	AACK*	+3.3V	TS*	134
135	ARTRY*		XATS*	136
137	DRTRY*		TBST*	138
139	TA*			140
141	TEA*			142
143			DBG*	144
145			DBB*	146
147			ABB*	148
149	TCLK_OUT		MPUBG0*	150
151			MPUBR0*	152

Table 5-5. CPU Debug (J1) Connector Pin Assignments (Continued)

153	MPUBR1*		IRQ0*	154
155	MPUBG1*		MCHK0*	156
157	IRQ1*		SMI*	158
159	MCHK1*		CKSTPI*	160
161	L2BR*		CKSTPO*	162
163	L2BG*		HALTED	164
165	CLAIM*		TLBISYNC*	166
167			TBEN0	168
169			Not Used	170
171		GND	DRVMOD0	172
173			DRVMOD1	174
175			NAPRUN	176
177	SRST1*		QREQ*	178
179	SRESET*		QACK*	180
181	HRESET*		CPUTDO	182
183	GND		CPUTDI	184
185	MPUCLK4		CPUTCK	186
187	MPUCLK4		CPUTMS	188
189	MPUCLK4		CPUTRST*	190

PCI Expansion Connector - J6

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

Table 5-6. J6 - PCI Expansion Connector (J6) Pin Assignments

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*		PCIXP*	16
17	PCIXGNT*		PCIXREQ*	18
19	+12V	GND	–12V	20
21	PERR*		SERR*	22
23	LOCK*		SDONE	24
25	DEVSEL*		SBO*	26
27	GND		GND	28
29	TRDY*		IRDY*	30
31	STOP*		FRAME*	32
33	GND		GND	34
35	ACK64*		Reserved	36
37	REQ64*		Reserved	38

Table 5-6. J6 - PCI Expansion Connector (J6) Pin Assignments (Continued)

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		AD6	52
53	AD9		AD8	54
55	AD11		AD10	56
57	AD13	+5V	AD12	58
59	AD15		AD14	60
61	AD17		AD16	62
63	AD19		AD18	64
65	AD21		AD20	66
67	AD23		AD22	68
69	AD25		AD24	70
71	AD27		AD26	72
73	AD29		AD28	74
75	AD31		AD30	76

Table 5-6. J6 - PCI Expansion Connector (J6) Pin Assignments (Continued)

77	PAR64		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	GND	AD44	96
97	AD47		AD46	98
99	AD49		AD48	100
101	AD51		AD50	102
103	AD53		AD52	104
105	AD55		AD54	106
107	AD57		AD56	108
109	AD59		AD58	110
111	AD61		AD60	112
113	AD63		AD62	114

PCI Mezzanine Card Connectors - J11 through J14

Four 64-pin SMT connectors, J11 through J14, supply 32/64-bit PCI interfaces and P2 I/O between the MVME2300SC board and an optional add-on PCI Mezzanine Card (PMC) in PMC Slot 1. The pin assignments for PMC Slot 1 are listed in the following two tables.

Table 5-7. J11 - J12 PMC1 Connector Pin Assignments

J11				J12			
1	TCK	–12V	2	1	+12V	TRST*	2
3	GND	INTA*	4	3	TMS	PMC1TDO	4
5	INTB*	INTC*	6	5	PMC2TDO	GND	6
7	PMC1P*	+5V	8	7	GND	Not Used	8
9	PMCINTD*	Not Used	10	9	Not Used	Not Used	10
11	GND	Not Used	12	11	Pull-up	+3.3V	12
13	PCICLK6	GND	14	13	PCIRST*	Pull-down	14
15	GND	PMC1GNT*	16	15	+3.3V	Pull-down	16
17	PMC1REQ*	+5V	18	17	Not Used	GND	18
19	+5V	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 (AD16)	AD23	26
27	AD22	AD21	28	27	+3.3V	AD20	28
29	AD19	+5V	30	29	AD18	GND	30
31	+5V	AD17	32	31	AD16	C/BE2*	32
33	FRAME*	GND	34	33	GND	Not Used	34
35	GND	IRDY*	36	35	TRDY*	+3.3V	36
37	DEVSEL*	+5V	38	37	GND	STOP*	38
39	GND	LOCK*	40	39	PERR*	GND	40
41	SDONE*	SBO*	42	41	+3.3V	SERR*	42
43	PAR	GND	44	43	C/BE1*	GND	44
45	+5V	AD15	46	45	AD14	AD13	46
47	AD12	AD11	48	47	GND	AD10	48
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	AD03	58	57	Not Used	Not Used	58
59	AD02	AD01	60	59	GND	Not Used	60
61	AD00	+5V	62	61	ACK64*	+3.3V	62
63	GND	REQ64*	64	63	GND	Not Used	64

Table 5-8. J13 - J14 PMC1 Connector Pin Assignments

J13				J14			
1	Reserved	GND	2	1	MC (P2-C1)	SD_15 (P2-A1)	2
3	GND	C/BE7*	4	3	SD_14 (P2-C2)	SD_13 (P2-A2)	4
5	C/BE6*	C/BE5*	6	5	SD_12 (P2-C3)	GND (P2-A3)	6
7	C/BE4*	GND	8	7	SD_11 (P2-C4)	SD_10 (P2-A4)	8
9	+5V	PAR64	10	9	SD _9 (P2-C5)	SD_8 (P2-A5)	10
11	AD63	AD62	12	11	SD_7 (P2-C6)	GND (P2-A6)	12
13	AD61	GND	14	13	SD_6 (P2-C7)	SD_5 (P2-A7)	14
15	GND	AD60	16	15	SD_4 (P2-C8)	SD_3 (P2-A8)	16
17	AD59	AD58	18	17	SD_2 (P2-C9)	SD_1 (P2-A9)	18
19	AD57	GND	20	19	GND (P2-C10)	SD_0 (P2-A10)	20
21	+5V	AD56	22	21	CLKFAIL (P2-C11)	FSYNC* (P2-A11)	22
23	AD55	AD54	24	23	SREF8K (P2-C12)	SCLK (P2-A12)	24
25	AD53	GND	26	25	GND (P2-C13)	SCLKx2* (P2-A13)	26
27	GND	AD52	28	27	SL_4* (P2-C14)	CLKFAILA (P2-A14)	28
29	AD51	AD50	30	29	SL_2* (P2-C15)	SL_3* (P2-A15)	30
31	AD49	GND	32	31	SL_0* (P2-C16)	SL_1* (P2-A16)	32
33	GND	AD48	34	33	Reserved	Reserved	34
35	AD47	AD46	36	35	Reserved	Reserved	36
37	AD45	GND	38	37	Reserved	Reserved	38
39	+5V	AD44	40	39	Reserved	Reserved	40
41	AD43	AD42	42	41	Reserved	Reserved	42
43	AD41	GND	44	43	Reserved	Reserved	44
45	GND	AD40	46	45	Reserved	Reserved	46
47	AD39	AD38	48	47	Reserved	Reserved	48
49	AD37	GND	50	49	Reserved	Reserved	50
51	GND	AD36	52	51	Reserved	Reserved	52
53	AD35	AD34	54	53	Reserved	Reserved	54
55	AD33	GND	56	55	Reserved	Reserved	56
57	+5V	AD32	58	57	Reserved	Reserved	58
59	Reserved	Reserved	60	59	Reserved	Reserved	60
61	Reserved	GND	62	61	Reserved	Reserved	62
63	GND	Reserved	64	63	Reserved	Reserved	64

PCI Mezzanine Card Connectors - J21 through J24

Four 64-pin SMT connectors, J21 through J24, supply 32/64-bit PCI interfaces and P2 I/O between the MVME2300SC board and an optional add-on PCI Mezzanine Card (PMC) in PMC Slot 2. The pin assignments for PMC Slot 2 are listed in the following two tables.

Table 5-9. J21 and J22 PMC2 Connector Pin Assignments

J21				J22			
1	TCK	–12V	2	1	+12V	TRST#	2
3	GND	INTA#	4	3	TMS	PMC2TDO	4
5	INTB#	INTC#	6	5	PCIXTDO	GND	6
7	PMC2P#	+5V	8	7	GND	Not Used	8
9	PMCINTC#	Not Used	10	9	Not Used	Not Used	10
11	GND	Not Used	12	11	Pull-up	+3.3V	12
13	PCICLK7	GND	14	13	PCIRST#	Pull-down	14
15	GND	PMC2GNT#	16	15	+3.3V	Pull-down	16
17	PMC2REQ#	+5V	18	17	Not Used	GND	18
19	+5V	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 (AD17)	AD23	26
27	AD22	AD21	28	27	+3.3V	AD20	28
29	AD19	+5V	30	29	AD18	GND	30
31	+5V	AD17	32	31	AD16	C/BE2#	32
33	FRAME#	GND	34	33	GND	Not Used	34
35	GND	IRDY#	36	35	TRDY#	+3.3V	36
37	DEVSEL#	+5V	38	37	GND	STOP#	38
39	GND	LOCK#	40	39	PERR#	GND	40
41	SDONE#	SBO#	42	41	+3.3V	SERR#	42
43	PAR	GND	44	43	C/BE1#	GND	44
45	+5V	AD15	46	45	AD14	AD13	46
47	AD12	AD11	48	47	GND	AD10	48
49	AD09	+5V (Vio)	50	49	AD08	+3.3V	50
51	GND	C/BE0#	52	51	AD07	Not Used	52
53	AD06	AD05	54	53	+3.3V	Not Used	54
55	AD04	GND	56	55	Not Used	GND	56
57	+5V	AD03	58	57	Not Used	Not Used	58
59	AD02	AD01	60	59	GND	Not Used	60
61	AD00	+5V (Vio)	62	61	ACK64#	+3.3V	62
63	GND	REQ64#	64	63	GND	Not Used	64

Table 5-10. J23 and J24 PMC2 Connector Pin Assignments

	J23				J24		
1	Reserved	GND	2	1	MC (P2-C1)	SD_15 (P2-A1)	**2**
3	GND	C/BE7#	4	3	SD_14 (P2-C2)	SD_13 (P2-A2)	**4**
5	C/BE6#	C/BE5#	6	5	SD_12 (P2-C3)	GND (P2-A3)	**6**
7	C/BE4#	GND	8	7	SD_11 (P2-C4)	SD_10 (P2-A4)	**8**
9	+5V	PAR64	10	9	SD _9 (P2-C5)	SD_8 (P2-A5)	**10**
11	AD63	AD62	12	11	SD_7 (P2-C6)	GND (P2-A6)	**12**
13	AD61	GND	14	13	SD_6 (P2-C7)	SD_5 (P2-A7)	**14**
15	GND	AD60	16	15	SD_4 (P2-C8)	SD_3 (P2-A8)	**16**
17	AD59	AD58	18	17	SD_2 (P2-C9)	SD_1 (P2-A9)	**18**
19	AD57	GND	20	19	GND (P2-C10)	SD_0 (P2-A10)	**20**
21	+5V	AD56	22	21	CLKFAIL (P2-C11)	FSYNC* (P2-A11)	**22**
23	AD55	AD54	24	23	SREF8K (P2-C12)	SCLK (P2-A12)	**24**
25	AD53	GND	26	25	GND (P2-C13)	SCLKx2* (P2-A13)	**26**
27	GND	AD52	28	27	SL_4* (P2-C14)	CLKFAILA (P2-A14)	**28**
29	AD51	AD50	30	29	SL_2* (P2-C15)	SL_3* (P2-A15)	**30**
31	AD49	GND	32	31	SL_0* (P2-C16)	SL_1* (P2-A16)	**32**
33	GND	AD48	34	33	Reserved	Reserved	**34**
35	AD47	AD46	36	35	Reserved	Reserved	**36**
37	AD45	GND	38	37	Reserved	Reserved	**38**
39	+5V	AD44	40	39	Reserved	Reserved	**40**
41	AD43	AD42	42	41	Reserved	Reserved	**42**
43	AD41	GND	44	43	Reserved	Reserved	**44**
45	GND	AD40	46	45	Reserved	Reserved	**46**
47	AD39	AD38	48	47	Reserved	Reserved	**48**
49	AD37	GND	50	49	Reserved	Reserved	**50**
51	GND	AD36	52	51	Reserved	Reserved	**52**
53	AD35	AD34	54	53	Reserved	Reserved	**54**
55	AD33	GND	56	55	Reserved	Reserved	**56**
57	+5V	AD32	58	57	Reserved	Reserved	**58**
59	Reserved	Reserved	60	59	Reserved	Reserved	**60**
61	Reserved	GND	62	61	Reserved	Reserved	**62**
63	GND	Reserved	64	63	Reserved	Reserved	**64**

Board Specifications

The following table lists the general specifications for the MVME2300SC VME processor module. Subsequent sections detail cooling requirements and EMC regulatory compliance.

A complete functional description of the MVME2300SC boards appears in Chapter 4. Specifications for the optional PMCs can be found in the documentation for those modules.

Table A-1. MVME2300SC Specifications

Characteristics		Specifications
MPU	MPC604, 300 MHz	SPECint95: 10.8 @ 50ns EDO
		16KB/16KB I/D on-chip cache
Memory	DRAM	32MB or 64MB, ECC-protected
	Flash	1MB via two 32-pin PLCC sockets
		4MB via surface mount
TOD clock device	M48T559	8KB NVRAM
Timers	One watchdog timer; time-out generates reset	
	Four real-time 16-bit programmable timers	
Power requirements, with no PMCs installed (See Note)	+12Vdc, 0mA –12Vdc, 0mA (typical)	+5Vdc (±5%), 4.5A typical, 5.5A maximum with MP604
Operating temperature	0°C to 55°C entry air with forced-air cooling (refer to *Cooling Requirements* section)	
Storage temperature	–40°C to +85° C	
Relative humidity	10% to 80%	
Vibration (operating)	2 Gs RMS, 20Hz-2000Hz random	
Altitude (operating)	5000 meters (16,405 feet)	

Table A-1. MVME2300SC Specifications (Continued)

Characteristics		Specifications
Physical dimensions (base board only)	Height	Double-high VME board, 9.2 in. (233 mm)
	Front panel width	0.8 in. (19.8 mm)
	Front panel height	10.3 in. (261.7 mm)
	Depth	6.3 in. (160 mm)
PCI Mezzanine Card (PMC) slots	Address/Data	A32/D32/D64, PMC PN1-4 connectors
	Bus Clock	33MHz
	Signaling	5V
	Power	7.5 watts maximum per slot (see Note)
	Module types	Basic, single-wide (74.0 mm x 149.0 mm)
		Basic, double-wide, (149.0 mm x 149.0 mm)
	PMC I/O	Front panel and/or VMEbus P2 I/O; P2 pinout accommodates access to SCSA backplane bus
PCI expansion connector	Address/Data	A32/D32/D46, 114-pin connector
	PCI bus clock	33 MHz
	Signaling	5V
Peripheral Component Interconnect (PCI)	PCI bridge	
	PCIbus, 32-/64-bit, 33MHz	
VMEbus (IEEE STD 1014)	DTB master	A16-A32; D08-D64, BLT
	DTB slave	A24-A32; D08-D64, BLT, UAT
	Arbiter	Round Robin or Priority
	Interrupt handler	IRQ 1-7
	Interrupt controller	Any one of seven
	System controller	Via jumper or auto detect
	Location monitor	Two LMA32
Ethernet interface	DEC 21143 controller with PCI local bus DMA	
	Front panel I/O through RJ45 connector; VMEbus P2 I/O for AUI	

Table A-1. MVME2300SC Specifications (Continued)

Characteristics	Specifications
Asynchronous serial debug port	TL16C550 UART
	Front panel I/O through RJ45 connector; VMEbus P2 I/O
Front panel switches and status indicators	Reset and Abort switches
	Four LEDs: BFL, CPU, SCON, FUS

Note The power requirement listed for the MVME2300SC 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 MVME2300SC: +3.3V, +5V, +12V, and –12V.

Cooling Requirements

The Motorola MVME2300SC 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 VME system 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 that component vendors' specifications are not exceeded.

While the exact amount of airflow required for cooling depends on the ambient air temperature and the type, number, and location of boards and other heat sources, adequate cooling can usually be achieved with 10 CFM and 490 LFM flowing over the module. Less airflow is required to cool the module in environments having lower maximum ambients. Under more favorable thermal conditions, it may be possible to operate the module reliably at higher than 55° C with increased airflow. It is important to note that there are several factors, in addition to the rated CFM of the air mover, which determine the actual volume and speed of air flowing over a module.

EMC Regulatory Compliance

The MVME2300SC 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 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 MVME2300SC 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.

Troubleshooting B

Solving Startup Problems

In the event of difficulty with your MVME2300SC VME processor module, 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. (The board was tested under these conditions before it left the factory.) The self-tests may not run in all user-customized environments.

Table B-1. Troubleshooting MVME2300SC Modules

Condition	Possible Problem	Try This:
I. Nothing works, no display on the terminal.	A. If the CPU LED is not lit, the board may not be getting correct power.	1. Make sure the system is plugged in. 2. Check that the board is securely installed in its backplane or chassis. 3. Check that all necessary cables are connected to the board, per this manual. 4. Check for compliance with Installation Considerations, per this manual. 5. Review the Installation and Startup procedures, per this manual. They include a step-by-step 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 cabled or configured incorrectly.	Configure and cable the system console terminal as described in this manual.

Table B-1. Troubleshooting MVME2300SC Modules (Continued)

Condition	Possible Problem	Try This:
II. There is a display on the terminal, but input from the keyboard (and/or mouse, if present) has no effect.	A. The keyboard or mouse may be connected incorrectly.	Recheck the keyboard and/or mouse connections and power.
	B. Board jumpers may be configured incorrectly.	Check the board jumpers per this manual.
	C. You may have invoked flow control by pressing a HOLD or PAUSE key, or by typing**:** **<CTRL>-S**	Press the HOLD or PAUSE key again. If this does not free up the keyboard, type in: **<CTRL>-Q**
III. Debug prompt `PPC1-Bug>` does not appear at 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 installedl. 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 `PPC1-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.
	B. There may be some fault in the board hardware.	**Caution** Performing the next step (**env;d**) will change some parameters that may affect your system's operation.

(continues>)

Table B-1. Troubleshooting MVME2300SC Modules (Continued)

Condition	Possible Problem	Try This:
IV. *Continued*		2. At the command line prompt, type in: **env;d <CR>** This sets up the default parameters for the debugger environment. 3. When prompted to Update Non-Volatile RAM, type in: **y <CR>** 4. When prompted to Reset Local System, type in: **y <CR>** 5. After clock speed is displayed, immediately (within five seconds) press the Return key: **<CR>** or **BREAK** to exit to the System Menu. Then enter a 3 for "Go to System Debugger" and Return: **3 <CR>** Now the prompt should be: `PPC1-Diag>` 6. You may need to use the **cnfg** command (see your board Debugger Manual) to change clock speed and/or Ethernet Address, and then later return to: **env <CR>** and step 3. 7. Run the 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 B-1. Troubleshooting MVME2300SC Modules (Continued)

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

Related Documentation C

Motorola Computer Group Documents

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

- Contacting your local Motorola sales office
- Visiting MCG's World Wide Web literature site,

Document Title	Publication Number
MVME2300SC VME Processor Module Installation and Use	V2300SCA/IH
MVME2300-Series VME Processor Module Installation and Use	V2300A/IH
MVME2300-Series VME Processor Module Programmer's Reference Guide	V2300A/PG
PPCBug Firmware Package User's Manual (Parts 1 and 2)	PPCBUGA1/UM PPCBUGA2/UM
PPCBug Diagnostics Manual	PPCDIAA/UM
PMCspan PMC Adapter Carrier Module Installation and Use	PMCSPANA/IH

To locate and view the most up-to-date product information in PDF or HTML format, visit .

Manufacturers' Documents

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

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

Document Title and Source	Publication Number
PowerPC 604™ RISC Microprocessor Technical Summary Literature Distribution Center for Motorola Telephone: 1-800- 441-2447 FAX: (602) 994-6430 or (303) 675-2150	MPC604E/D
PowerPC 604™ RISC Microprocessor User's Manual Literature Distribution Center for Motorola Telephone: 1-800- 441-2447 FAX: (602) 994-6430 or (303) 675-2150 OR IBM Microelectronics PowerPC604e User Manual	MPC604EUM/AD G522-0330-00
PowerPC™ Microprocessor Family: The Programming Environment for 32-Bit Microprocessors Literature Distribution Center for Motorola Telephone: 1-800- 441-2447 FAX: (602) 994-6430 or (303) 675-2150	MPCFPE/AD

Document Title and Source	Publication Number
OR IBM Microelectronics Programming Environment Manual Web Site:	G522-0290-01
PC16550 UART National Semiconductor Corporation Customer Support Center (or nearest Sales Office) 2900 Semiconductor Drive P.O. Box 58090 Santa Clara, California 95052-8090 Telephone: 408-721-5000 Telephone: 1-800-272-9959	PC16550DV
21140 Fast Etherworks PCI 10-Flash-100 Ethernet Adapter Compaq Computer Corporation Telephone: 1-800.at.compaq	EK-DE500-OM

C

Document Title and Source	Publication Number
W83C553 Enhanced System I/O Controller with PCI Arbiter (PIB) Winbond Electronics Corporation Winbond Systems Laboratory 2727 North First Street San Jose, CA 95134 Telephone: (408) 943-6666 FAX:(408) 544-1798	W83C553F
M48T559 CMOS 8K x 8 TIMEKEEPER™ SRAM Data Sheet STMicroelectronics 1000 East Bell Road Phoenix, Arizona 85022 Telephone: (602) 485-6100 Fax: 602 485-6102	M48T559
Universe User Manual Tundra Semiconductor Corporation 603 March Road Kanata, ON K2K 2M5, Canada Telephone: 1-800-267-7231 OR 101-1265 Montecito Ave. Mountain View, California 94043, USA Telephone: (650) 960-0282 FAX: (650) 960-0321	Universe Part Number 9000000.MD303.01

Related Specifications

For additional information, refer to the following table for related specifications. For your convenience, a source for the listed document is also provided. It is important to note that in many cases, the information is preliminary and the revision levels of the documents are subject to change without notice.

Document Title and Source	Publication Number
Versatile Backplane Bus: VMEbus Institute of Electrical and Electronics Engineers, Inc. Web Site: http://standards.ieee.org/catalog/ OR Microprocessor System Bus for 1 to 4 Byte Data Bureau Central de la Commission Electrotechnique Internationale 3, rue de Varembé Geneva, Switzerland	ANSI/IEEE Standard 1014-1987 IEC 821 BUS
IEEE - Common Mezzanine Card Specification (CMC) Institute of Electrical and Electronics Engineers, Inc. Web Site: http://standards.ieee.org/catalog/	P1386 Draft 2.0
IEEE - PCI Mezzanine Card Specification (PMC) Institute of Electrical and Electronics Engineers, Inc. Web Site: http://standards.ieee.org/catalog/	P1386.1 Draft 2.0
Bidirectional Parallel Port Interface Specification Institute of Electrical and Electronics Engineers, Inc. Web Site: http://standards.ieee.org/catalog/	IEEE Standard 1284

C

Document Title and Source	Publication Number
Peripheral Component Interconnect (PCI) Local Bus Specification, Revision 2.0 PCI Special Interest Group	PCI Local Bus Specification
PowerPC Reference Platform (PRP) Specification, Third Edition, Version 1.0, Volumes I and II International Business Machines Corporation	MPR-PPC-RPU-02
PowerPC Microprocessor Common Hardware Reference Platform: A System Architecture (CHRP), Version 1.0 Literature Distribution Center for Motorola Telephone: 1-800- 441-2447 FAX: (602) 994-6430 or (303) 675-2150 OR Morgan Kaufmann Publishers, Inc. Telephone: (415) 392-2665 Telephone: 1-800-745-7323	
Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange Electronic Industries Alliance (for publications)	TIA/EIA-232 Standard

Glossary

10Base-5 An Ethernet implementation in which the physical medium is a doubly shielded, 50-ohm coaxial cable capable of carrying data at 10 Mbps for a length of 500 meters (also referred to as thicknet). Also known as thick Ethernet.

10Base-2 An Ethernet implementation in which the physical medium is a single-shielded, 50-ohm RG58A/U coaxial cable capable of carrying data at 10 Mbps for a length of 185 meters (also referred to as AUI or thinnet). Also known as thin Ethernet.

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.

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; a 7-bit code used to encode alphanumeric information. In the IBM-compatible world, this is expanded to eight 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.

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

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

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

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

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.

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

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

DIMM **D**ual **I**nline **M**emory **M**odule.

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.

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

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.
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.
ESD	**E**lectro-**S**tatic **D**ischarge/Damage
Ethernet	A local area network standard that uses radio frequency signals carried by coaxial cables.
Falcon	The DRAM controller chip developed by Motorola for the MVME2600 and MVME3600 series of boards. It is intended to be used in sets of two to provide the necessary interface between the Power PC60*x* bus and the 144-bit ECC DRAM (system memory array) and/or ROM/Flash.
fast Ethernet	See 100Base-TX.
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.
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).
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.
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

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
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.
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
nonvolatile memory	A memory in which the data content is maintained whether the power supply is connected or not.
NVRAM	**N**on-**V**olatile **R**andom **A**ccess **M**emory
OEM	**O**riginal **E**quipment **M**anufacturer
OS	**O**perating **S**ystem. The software that manages the computer resources, accesses files, and dispatches programs.
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.

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

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.

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

ROM — **R**ead-**O**nly **M**emory

RTC — **R**eal-**T**ime **C**lock

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

SCSA — **S**ignal **C**omputing **S**ystem **A**rchitecture. A hardware model for computer telephony servers. A key SCSA element is a TDM (time division multiplexed) telephony bus for voice and video signals, known as the SCbus™ in VME implementations of this architecture.

SCSI — **S**mall **C**omputer **S**ystems **I**nterface. An industry-standard high-speed interface primarily used for secondary storage. SCSI-1 provides up to 5 Mbps data transfer.

SCSI-2 (Fast/Wide) — An improvement over plain SCSI; and includes command queuing. Fast SCSI provides 10 Mbps data transfer on an 8-bit bus. Wide SCSI provides up to 40 Mbps data transfer on a 16- or 32-bit bus.

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

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

TDM — **T**ime **D**ivision **M**ultiplexing. A multiplexing scheme in which individual I/O ports or channels share slices of time on an aggregate channel, Receivers and transmitters are synchronized. The SCbus™ is a TDM implementation that provides up to 2048 time slots, the equivalent of 1024 voice conversations at 64 Kbps.

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 which provides the complete interface between the PCI bus and the VMEbus.
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.
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).
volatile memory	A memory in which the data content is lost when the power supply is disconnected.

Index

Numerics

10/100 BASET port 4-20
16/32-bit timers 4-18

A

abort process 2-11
address pipelining (PowerPC bus) 4-8, 4-9
altitude (operating) A-1
ambient air temperature (effect on cooling) A-4
arguments, firmware command 3-4
asynchronous serial debug port 4-15
Autoboot function, enabling 3-15
autoboot process 2-7
autojumpering (VME backplane) 1-19

B

backplane
  connectors, P1 and P2 1-21
  jumpers 1-19
battery 4-17
baud rate (serial port) 1-20
baud rate, default 2-6
BBRAM (battery-backed-up RAM) and clock 3-9
BG (bus grant) signal 1-19
bits per character (serial port) 1-20
block diagram
  board 4-6
  memory 4-7
board
  architecture 4-1
  configuration 1-5
  dimensions A-2
  features 4-1
  layout, MVME2300SC 1-7
  placement 1-19
  preparation 1-1
booting the system 2-7, 2-8, 2-9
BREAK key 2-11
break process 2-11
bus grant (BG) signal 1-19
buses, standard 4-3, 4-4

C

cables, I/O ports A-4
chassis grounding A-4
command identifier, firmware 3-4
command line input, firmware 3-3
command mode, debugger 3-3
commands, firmware
  CNFG (Configure Board Information Block) 3-10
  ENV (Set Bug/Operating System Environment) 3-11
conductive chassis rails (EMC compliance) A-4
configurable items, MVME2300SC base board 1-6
Configure Board Information Block (CNFG) firmware command 3-10

configuring
hardware 2-5
PMCs 1-10, 2-5
PPCBug parameters 3-12
VMEbus interface 3-21
connector pin assignments 5-1
console terminal, preparing 1-20
cooling requirements A-3
counters 4-17

D

date and time, setting 2-7
debug firmware, PPCBug 3-1
DEBUG port 1-20
debugger
directory 3-8
firmware (PPCBug) 3-9, 4-10
memory requirements 3-3
prompt 3-2
DECchip 21143 LAN controller 4-11
default baud rate 2-6
diagnostic facilities 2-11
diagnostics
directory 3-8
hardware 3-8
mode (debugger) 3-3
prompt 3-2
test groups 3-8
dimensions of base board A-2
directories, debugger and diagnostic 3-8
directories, switching 2-11
documentation, related C-1
DRAM
base address 1-21
latency 4-7
memory 4-5
parity function, enabling 3-20
quantity available A-1
speed setting 3-19

E

EMC regulatory compliance A-4
environmental parameters 3-9
ESD (electrostatic discharge), precautions against 1-4
Ethernet
interface 4-11
station address 4-11

F

features
hardware 4-1
Universe ASIC 4-14
firmware
directories 3-3
initialization 2-2
Flash memory 4-10
bank selection (J8) 1-8
latency 4-10
forced air cooling A-3
front panel
controls 2-1
switches and indicators 2-1
functional description 4-1

G

general description, MVME2300SC 4-3
global bus timeout 1-21
grounding A-4
grounding strap, use of 1-4

H

handshaking, hardware 2-6
hardware
diagnostics 3-8
features 4-1
initialization 2-3
preparation 1-1
HE (Help) command (debugger) 3-8
humidity, relative A-1

I

IACK (interrupt acknowledge) signal 1-19
indicators (front panel) 2-1
initial conditions 2-4
installation considerations 1-21
installing
  multiple MVME2300SC boards 1-21
  MVME2300SC in chassis 1-18
  PMC carrier board 1-17
  PMC modules 1-11, 1-12
  PMCspan-002 1-15
  primary PMCspan module 1-13
  secondary PMCspan module 1-16
interconnect signals 5-1
interface
  Ethernet 4-11
  PCI bus 4-12
  VMEbus 4-14
interrupt acknowledge signal (IACK) 1-19
interrupt controller (MPIC) 4-17
interval timers 4-18
ISA bus 4-3, 4-4, 4-16

J

jumper headers 1-6
  J10 (software-readable jumpers) 1-9, 2-5
  J8 (Flash memory bank selection) 1-8, 2-5
  J9 (system controller selection) 1-9, 2-5
jumpers, backplane 1-19

L

L2 cache parity function, enabling 3-20
latency
  DRAM 4-7
  Flash memory 4-10
LED/serial startup diagnostic codes
  in hardware initialization 3-20
LEDs
  MVME2300SC front panel 2-1, 4-20
lowercase characters, commands and 3-9

M

manufacturers' documents C-2
memory
  block diagram 4-7
  quantity available A-1
  requirements, debugger 3-3
  sizing function, enabling 3-18
MPIC (MultiProcessor Interrupt Controller) 4-17
MPU specifications A-1
MVME2300SC
  board layout 1-7
  cooling requirements A-3
  EMC regulatory compliance A-4
  features 4-1
  installing 1-18
  regulatory compliance A-4
  specifications A-1

N

Network Auto Boot function, enabling 3-17
network boot process 2-9
non-volatile RAM (NVRAM) 3-9, 3-11
NVRAM (BBRAM) configuration area 4-11

O

operating parameters 3-9
operating temperature A-1
option fields, in firmware command 3-4

P

P1 and P2 connectors 4-19, 5-1
parity setting (serial port) 1-20
PCI bus 4-3, 4-5, 4-12, 4-15
PCI expansion 4-12, 4-14
PCI Host Bridge (PHB) 4-17
PCI/ISA Bridge (PIB)
  controller functions 4-15
Peripheral Component Interconnect (PCI) A-2
PHB (PCI Host Bridge) 4-17
PIB controller 4-15

pin assignments, connector 5-1
PMC slot 1
connector pin assignments, J11 and J12 5-13
connector pin assignments, J13 and J14 5-14
PMC slot 2
connector pin assignments, J21 and J22 5-16
connector pin assignments, J23 and J24 5-17
PMCs
carrier board installation 1-17
configuring 1-10, 2-5
installing 1-11
placement on MVME2300SC 1-12
power requirements A-3
PMCspan expansion modules
preparing 1-10
port
asynchronous serial 4-15
debug 4-15
Ethernet 4-20
power requirements, MVME2300SC 1-21, 4-4, A-1, A-3
powering up the board 2-1
PowerPC-bus-to-DRAM access timing 4-7
PPCBug
firmware 3-1, 3-9, 4-10
implementation 3-2
overview 3-1
prompt 3-2
preparing
MVME2300SC board 1-1
PMCs 1-10
PMCspan expansion modules 1-10
system console terminal 1-20
processor bus 4-3, 4-4
prompt, debugger 3-2, 3-8
prompt, diagnostics 3-2

R

Raven MPU/PCI bus bridge controller ASIC 4-4, 4-15
readable jumpers (J10) 1-9
real-time clock/NVRAM/timer function 4-16
regulatory compliance A-4
regulatory guidelines A-4
related documentation C-1
related specifications C-5
relative humidity A-1
remote control/status connector 4-18
reset vector header (J8) 1-8
resetting the system 2-1, 2-9, 2-10, 3-9
restart mode 3-9
RF emissions, minimizing A-4
ROMboot function, enabling 3-16
ROMboot process 2-8
ROMFAL (ROM first access length) parameter 3-19
ROMNAL (ROM next access length) parameter 3-19

S

safety procedures 1-5
SCSA bus 1-21, 4-2, 4-3, 4-4
SCSI bus 3-13
SD command (debugger) 2-11, 3-8
serial
communication parameters 2-4
port, MVME2300SC 1-20, 4-20
Set Bug/Operating System Environment (ENV) firmware command 3-11
setting date and time 2-7
setting up the terminal 1-20
shielded cables A-4
single-width PMC module installation 1-12
size of base board A-2
software-readable jumpers (J10) 1-9
speaker output 4-18
specifications, related C-5
startup overview 1-1
startup problems, solving B-1

INDEX

status indicators (front panel) 4-20
stop bits per character (serial port) 1-20
storage temperature A-1
switches (front panel) 2-1
switching directories 2-11
SYSFAIL* signal 3-13
system
  console setup 2-6
  console, connecting 1-20
  controller function 2-5
  reset 2-10
  startup 2-3
system controller
  selection header (J9) 1-9
  using board as 1-19
System Fail (SYSFAIL*) signal 2-8

T

temperature
  operating A-1
  storage A-1
terminal
  configuration 2-4
  input/output control 3-3
  setup 1-20
testing the hardware 3-8
timeout, global 1-21
timers, programmable 4-17
  16/32-bit 4-18
  interval 4-18
  via Universe chip A-1
timing, PowerPC bus to DRAM 4-7
TL16C550 UART 4-20
troubleshooting procedures 3-8, B-1
types of reset 2-10

U

Universe VMEbus interface ASIC 4-14
uppercase characters, commands and 3-9

V

vibration (operating) A-1
VME processor module
  board layout 1-7
VMEbus 4-3, 4-4
  address/data configurations 1-21
  connectors 5-1
  system controller selection (J9) 1-9
  Universe ASIC and 4-14
VMEbus interface A-2
  configuring 3-21

W

Winbond PCI/ISA bus bridge controller 4-15

X

XON/XOFF handshaking 2-6

INDEX

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

Gold-standard solutions

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

We buy equipment

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

Learn more!

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

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

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

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

ARTISAN® TECHNOLOGY GROUP