MVME162P4 VME Embedded Controller

Installation and Use

V162PFXA/IH1

June 2000

Safety Summary

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

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

Ground the Instrument.

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

Do Not Operate in an Explosive Atmosphere.

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

Keep Away From Live Circuits Inside the Equipment.

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

Use Caution When Exposing or Handling a CRT.

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

Do Not Substitute Parts or Modify Equipment.

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

Observe Warnings in Manual.

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

⚠ **Warning**

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

Flammability

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

EMI Caution

⚠ **Caution**

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

Lithium Battery Caution

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

⚠ **Caution**

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

⚠ **Attention**

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

⚠ **Vorsicht**

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

CE Notice (European Community)

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

EN55022 “Limits and Methods of Measurement of Radio Interference Characteristics of Information Technology Equipment”; this product tested to Equipment Class B

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.

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

Contents

About This Manual

CHAPTER 1 Hardware Preparation and Installation

CHAPTER 2 Startup and Operation

CHAPTER 3 162Bug Firmware

CHAPTER 4 Functional Description

CHAPTER 5 Pin Assignments

APPENDIX A Specifications

APPENDIX B Troubleshooting

APPENDIX C Network Controller Data

APPENDIX D Disk/Tape Controller Data

APPENDIX E Related Documentation

List of Figures

List of Tables

About This Manual

MVME162P4 VME Embedded Controller Installation and Use provides instructions for hardware preparation and installation; a board-level hardware overview; and firmware-related general information and startup instructions for the MVME162P-2*xx* and -3*xx* series of embedded controllers, known collectively as the ‘‘MVME162P4’’ because they are equipped with the “Petra” chip and accommodate up to four IP modules.

The “Petra” chip that distinguishes MVME162P4 embedded controllers is an application-specific integrated circuit (ASIC) which combines the functions previously covered by the MC2 chip, the IP2 chip, and the MCECC chip in a single ASIC. As of the publication date, the information presented in this manual applies to the following MVME162P4 models:

Model Number	**Characteristics**
MVME162P-244L	25MHz 68LC040, 16MB SDRAM, 2 SIO, 4 DMA IP
MVME162P-244LE	25MHz 68LC040, 16MB SDRAM, 2 SIO, 4 DMA IP, Ethernet
MVME162P-244LSE	25MHz 68LC040, 16MB SDRAM, 2 SIO, 4 DMA IP, SCSI/Ethernet
MVME162P-344	32MHz 68040, 16MB SDRAM, 2 SIO, 4 DMA IP
MVME162P-344S	32MHz 68040, 16MB SDRAM, 2 SIO, 4 DMA IP, SCSI
MVME162P-344E	32MHz 68040, 16MB SDRAM, 2 SIO, 4 DMA IP, Ethernet
MVME162P-344SE	32MHz 68040, 16MB SDRAM, 2 SIO, 4 DMA IP, SCSI/Ethernet

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 Appendix E, *Related Documentation*.

Overview of Contents

Chapter 1, *Hardware Preparation and Installation*, provides unpacking instructions, hardware preparation guidelines, and installation instructions for the MVME162P4 VME Embedded Controller.

Chapter 2, *Startup and Operation*, provides information on powering up the MVME162P4 VME Embedded Controller after its installation in a system, and describes the functionality of the switches, status indicators, and I/O ports.

Chapter 3, *162Bug Firmware*, describes the basics of 162Bug 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 MVME162P4 VME embedded controller on a block diagram level.

Chapter 5, *Pin Assignments*, summarizes the pin assignments for the following groups of interconnect signals on the MVME162P4.

Appendix A, *Specifications*, lists the general specifications for the MVME162P4 VME embedded controller. The subsequent sections detail cooling requirements and EMC regulatory compliance.

Appendix B, *Troubleshooting*, includes simple troubleshooting steps in the event that you have difficulty with your MVME162P4 VME embedded controller.

Appendix C, *Network Controller Data*, describes the VMEbus Network Controller modules that are supported by the 162Bug firmware.

Appendix D, *Disk/Tape Controller Data*, describes the VMEbus disk/tape controller modules that are supported by the 162Bug firmware.

Appendix E, *Related Documentation*, provides all documentation related to the MVME162P4.

Comments and Suggestions

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

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

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

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

Terminology

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

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

An asterisk (∗) following a signal name for signals that are *level significant* denotes that the signal is *true* or valid when the signal is low. An asterisk (∗) following a signal name for signals that are *edge significant* denotes that the actions initiated by that signal occur on high to low transition.

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

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**.

Hardware Preparation and Installation

1

Introduction

This chapter provides unpacking instructions, hardware preparation guidelines, and installation instructions for the MVME162P4 VME Embedded Controller. Hardware preparation of the MVME712 series transition modules compatible with this board is described in separate manuals.

Getting Started

This section supplies an overview of startup procedures applicable to the MVME162P4. Equipment requirements, directions for unpacking, and ESD precautions that you should take complete the section.

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...**
Unpack the hardware.	*Guidelines for Unpacking* on page 1-2
Reconfigure jumpers or switches on the MVME162P4 board as necessary.	*Preparing the Board* on page 1-4.
Ensure that IP modules are properly installed on the MVME162P4 board.	*IP Installation on the MVME162P4* on page 1-21.
Install the MVME162P4 board in a chassis.	*MVME162P4 Installation* on page 1-22.
Connect a display terminal.	*Serial Connections* on page 1-26.

Table 1-1. Startup Overview (Continued)

What you need to do...	Refer to...
Connect any other equipment you will be using.	Chapter 5, *Pin Assignments*.
	For more information on optional devices and equipment, refer to the documentation provided with the equipment.
Power up the system.	*Applying Power* on page 2-2.
	Solving Startup Problems on page B-1.
Note that the firmware initializes and tests the board.	*Bringing up the Board* on page 2-4.
	You may also wish to obtain the 162Bug Firmware User's Manual, listed in Appendix E, *Related Documentation*.
Initialize the system clock.	*Debugger Commands* on page 3-6.
Examine and/or change environmental parameters.	*Modifying the Environment* on page 3-9.
Program the board as needed for your applications.	Programmer's Reference Guide, listed in Appendix E, *Related Documentation*.

Equipment Required

The following equipment is necessary to complete an MVME162P4 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 MVME162P4 board.

Use ESD Wrist Strap

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 MVME162P4 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 MVME162P4, you may need to reconfigure hardware to some extent before installing the module.

Most options on the MVME162P4 are under software control: By setting bits in control registers after installing the module in a system, you can modify its configuration. (The MVME162P4 registers are described in Chapter 3 under *ENV – Set Environment*, and/or in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide* as listed in *"Related Documentation"* in Appendix E.)

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 base board.

MVME162P4 Configuration

Figure 1-1 illustrates the placement of the jumper headers, connectors, configuration switches, and various other components on the MVME162P4. Manually configurable jumper headers and configuration switches on the MVME162P4 are listed in the following table.

Note Jumper J3 is not listed, as it is provided for manufacturing purposes only and cannot be configured.

Table 1-2. MVME162P4 Configuration Settings

Function	Factory Default
VME System Controller (J1) on page 1-7	2-3
IP Bus Clock (J14) on page 1-7	1-2
SIM Selection for Serial Port B (J15) on page 1-8	SIMM06
Serial Port 1/Console Clock (J16) on page 1-11	No jumpers

Table 1-2. MVME162P4 Configuration Settings (Continued)

Function	Factory Default
Serial Port 2 Clock (J17) on page 1-11	No jumpers
IP Bus Strobe (J19) on page 1-11	No jumper
SRAM Backup Power Source (J22) on page 1-12	1-3, 2-4
EPROM Size (J23) on page 1-13	2-3
Flash Write Protection (J24) on page 1-14	1-2
MC2 DRAM Size (S3) on page 1-14	Off-Off-Off
General-Purpose Readable Switch (S4 Pin 5) on page 1-16	Off-On7
IP DMA Snoop Control (S5 Pins 1/2) on page 1-17	On-On
IP Reset Mode (S5 Pin 3) on page 1-18	On
Flash Write Enable Mode (S5 Pin 4) on page 1-19	On
MCECC DRAM Size (S6) on page 1-19	On-Off-On

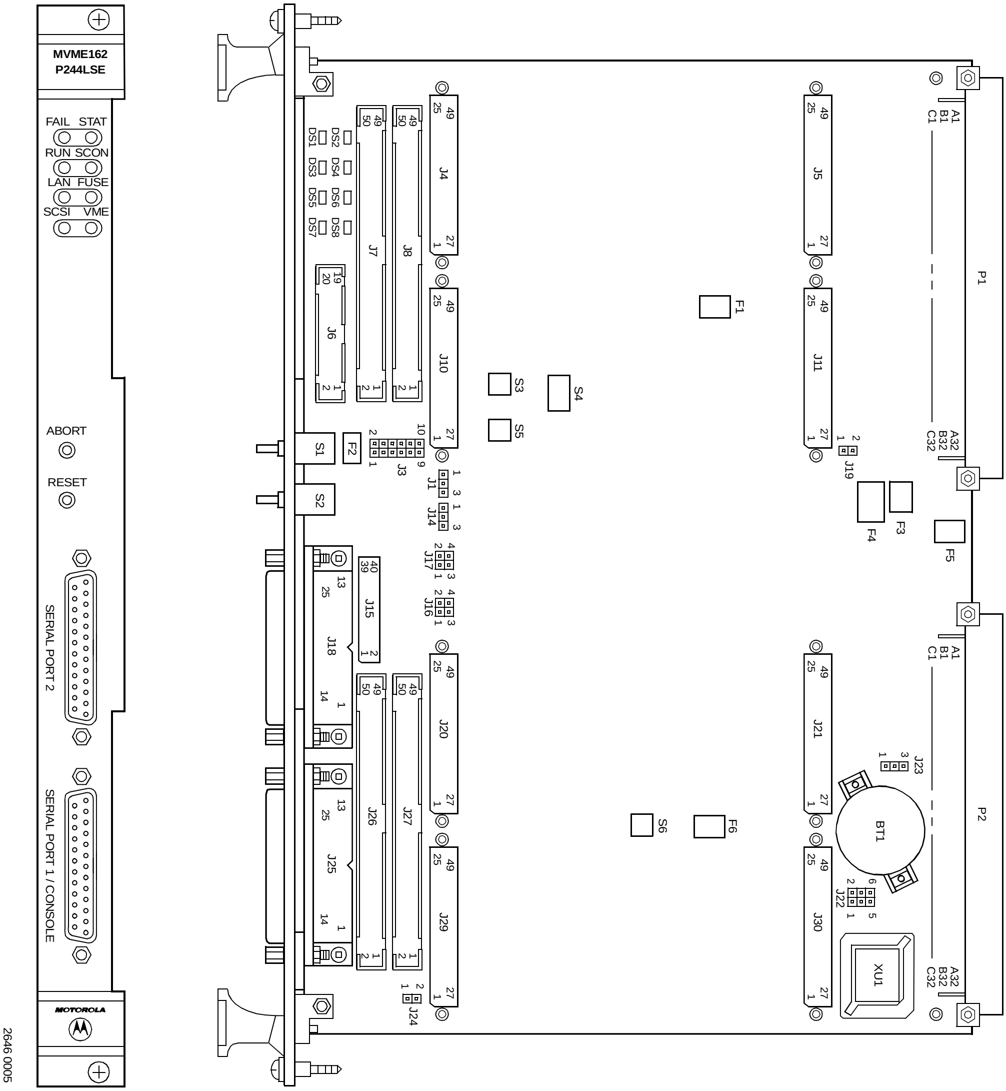


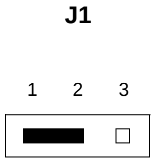
Figure 1-1. MVME162P4 Board Layout

VME System Controller (J1)

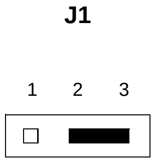
The MVME162P4 board is factory-configured in "automatic" system controller mode with a jumper across J1 pins 2-3. In this configuration, the MVME162P4 determines whether it is the system controller by its position on the bus. If the board is located in the first slot from the left, it configures itself as the system controller. When the board is operating as system controller, the **SCON** LED is turned on.

If you want the MVME162P4 to function as system controller in all cases, move the jumper to pins 1-2. If the MVME162P4 is not to be system controller under any circumstances, remove the jumper from J1.

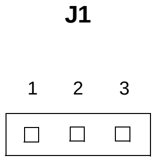
Note On MVME162P4 boards without the optional VMEbus interface (i.e., with no VMEchip2 ASIC), the jumper may be installed or removed with no effect on normal operation.



System Controller



Auto System Controller
(factory configuration)



Not System Controller

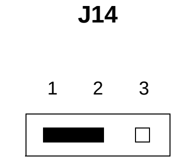
IP Bus Clock (J14)

Header J14 selects the speed of the IP bus clock. The IP bus clock speed may be 8MHz or it may be set synchronous to the processor bus clock (25MHz or 32MHz for the MC68040 and MC68LC040). The default factory configuration has a jumper installed on pins 1-2, denoting an 8MHz clock.

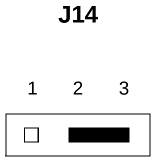
If the jumper is installed on J14 pins 2-3, the IP bus clock speed matches that of the processor bus clock (25/32MHz), allowing the IP module to pace the MPU. Whether the setting is 8MHz or the processor bus clock speed, all IP ports operate at the same speed.

Caution

The setting of the IP32 bit in the Control/Status registers (Petra IP2 sector, register at offset $1D, bit 0) must correspond to that of the jumper. The bit is cleared (0) for 8MHz, or set (1) to match the processor bus clock speed. If the jumper and the CSR bit are not configured the same, the board may not run properly.



Bus Clock = 8MHz
(Factory configuration)



Bus Clock = Proc. Bus Clock
(from MPU Bus Clock)

SIM Selection for Serial Port B (J15)

Port B of the MVME162P4 Z85230 serial communications controller is configurable via a serial interface module (SIM) that is installed at connector J15 on the board. Five serial interface modules are available:

- EIA-232-D (DCE and DTE)
- EIA-530 (DCE and DTE)
- EIA-485/EIA-422 (DCE or DTE)

You can change Port B from an EIA-232-D to an EIA-530 interface or to an EIA-485/EIA-422 interface (or vice-versa) by mounting the appropriate serial interface module. Port B is routed (via the SIM at J15) to the 25-pin DB25 front panel connector marked **SERIAL PORT 2**.

For the location of SIM connector J15 on the MVME162P4, refer to Figure 1-1. Figure 1-2 illustrates the secondary side (bottom) of a serial interface module, showing the J1 connector which plugs into SIM connector J15 on the MVME162P4. Figure 1-3 (sheets 3-6), Figure 1-4, Figure 1-5 (sheets 3 and 4), and Figure 1-6 illustrate the nine configurations available for Port B.

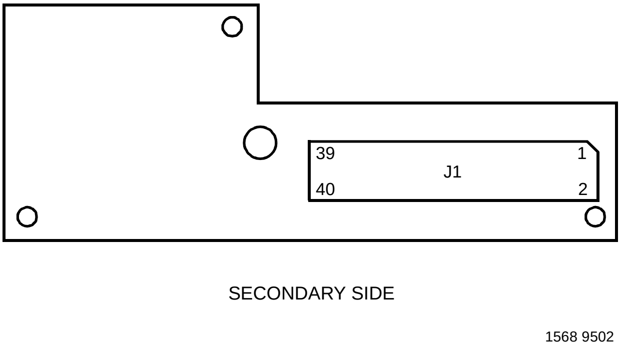


Figure 1-2. Serial Interface Module, Connector Side

For the part numbers of the serial interface modules, refer to Table 1-3. The part numbers are ordinarily printed on the primary side (top) of the SIMs, but may be found on the secondary side in some versions.

If you need to replace an existing serial interface module with a SIM of another type, go to *Removal of Existing SIM* below. If there is no SIM on the main board, skip to *Installation of New SIM*.

Table 1-3. Serial Interface Module Part Numbers

EIA Standard	Configuration	Part Number	Model Number
EIA-232-D	DTE	01-W3846B	SIMM05
	DCE	01-W3865B	SIMM06
EIA-530	DTE	01-W3868B	SIMM07
	DCE	01-W3867B	SIMM08
EIA-485	--	01-W3002F	SIMM09
or EIA-422	DTE or DCE		

Removal of Existing SIM

1. Each serial interface module is retained by two 4-40 x 3/16-in. Phillips-head screws in opposite corners. (Exception: SIMM09 is retained by one Phillips-head screw in the center of the module.) Remove the screw(s) and store them in a safe place for later use.

2. Grasp opposite sides of the SIM and gently lift straight up.

Caution

Avoid lifting the SIM by one side only, as the connector can be damaged on the SIM or the main board.

3. Place the SIM in a static-safe container for possible reuse.

Installation of New SIM

1. Observe the orientation of the connector keys on SIM connector J1 and MVME162P4 connector J15. Turn the SIM so that the keys line up and place it gently on connector J15, aligning the mounting hole(s) at the SIM corners (or center) with the matching standoff(s) on the MVME162P4.

2. Gently press the top of the SIM to seat it on the connector. If the SIM does not seat with gentle pressure, recheck the orientation. If the SIM connector is oriented incorrectly, the mounting hole(s) will not line up with the standoff(s).

Caution

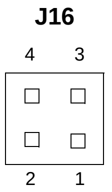
Do not attempt to force the SIM into place if it is oriented incorrectly.

3. Place the one or two 4-40 x 3/16” Phillips-head screw(s) that you previously removed (or that were supplied with the new SIM) into the one center or two opposite-corner mounting hole(s). Turn the screw(s) into the standoff(s) but do not overtighten.

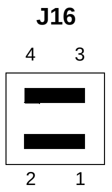
The signal relationships and signal connections in the various serial configurations available for ports A and B are illustrated in Figures 1-3 through 1-6.

Serial Port 1/Console Clock (J16)

The MVME162P4 is shipped from the factory with the **SERIAL PORT 1/CONSOLE** header configured for asynchronous communications (i.e., jumpers removed). To select synchronous communications for the **SERIAL PORT 1/CONSOLE** connection, install jumpers across pins 1-2 and pins 3-4.



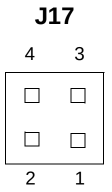
Internal Clock
(Factory configuration)



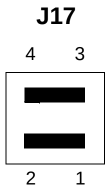
External Clock

Serial Port 2 Clock (J17)

The MVME162P4 is shipped from the factory with the **SERIAL PORT 2** header configured for asynchronous communications (that is, jumpers removed). To select synchronous communications for the **SERIAL PORT 2** connection, install jumpers across pins 1-2 and pins 3-4.



Internal Clock
(Factory configuration)



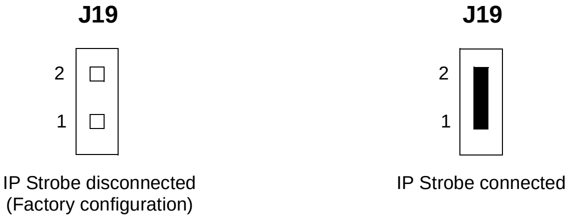
External Clock

IP Bus Strobe (J19)

Some IP bus implementations make use of the Strobe∗ signal (pin AA19 on the Petra ASIC) as an input to the IP modules from the Petra IP2 sector. Other IP interfaces require that the strobe be disconnected.

With a jumper installed between J19 pins 1-2, a programmable frequency source is connected to the Strobe∗ signal on the IP bus (for details, refer to the Petra/IP2 chip programming model in the *Programmer's Reference Guide*).

If the jumper is removed from J19, the strobe line is available for a sideband type of messaging between IP modules. The Strobe∗ signal is not connected to any active devices on the board, but it may be connected to a pull-up resistor.



IP Strobe disconnected (Factory configuration)

IP Strobe connected

SRAM Backup Power Source (J22)

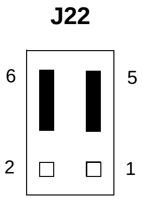
Header J22 determines the source for onboard static RAM backup power on the MVME162P4. In the factory configuration, VMEbus +5V standby voltage serves as primary and secondary power source (the onboard battery is disconnected).

The backup power configurations available for onboard SRAM through header J22 are illustrated in the following diagram.

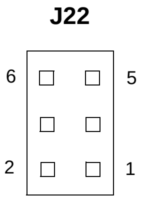
Note For MVME162P4s without the optional VMEbus interface (that is, without the VMEchip2 ASIC), you must select the onboard battery as the backup power source.

Caution

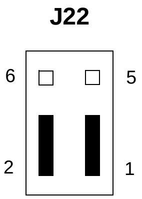
Removing all jumpers may temporarily disable the SRAM. Do not remove all jumpers from J22, except for storage.



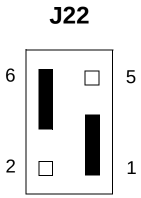
Primary Source Onboard Battery
Secondary Source Onboard Battery



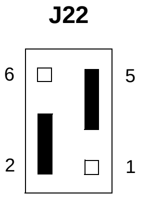
Backup Power Disabled
(For storage only)



Primary Source VMEbus +5V STBY
Secondary Source VMEbus +5V STBY
(Factory configuration)



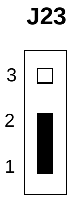
Primary Source VMEbus +5V STBY
Secondary Source Onboard Battery



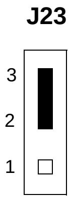
Primary Source Onboard Battery
Secondary Source VMEbus +5V STBY

EPROM Size (J23)

Header J23 selects the EPROM size. The MVME162P4 is factory-configured for a 4Mbit EPROM.



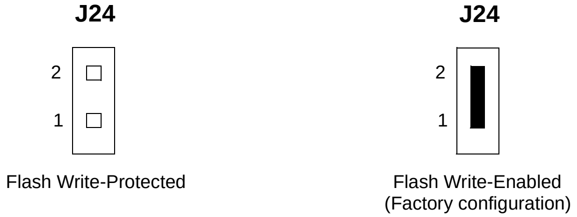
8Mbit EPROM



4Mbit EPROM
(Factory configuration)

Flash Write Protection (J24)

When the Flash write-enable jumper is installed, Flash memory can be written to via the normal software routines. When the jumper is removed (factory configuration), Flash memory is not writable.



MC2 DRAM Size (S3)

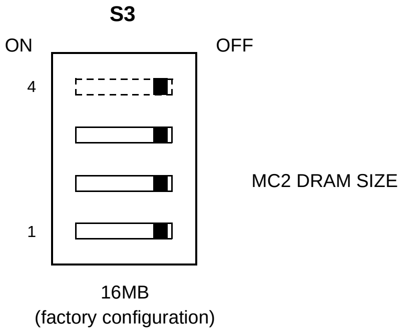
MVME1x2P4 boards use SDRAM (Synchronous DRAM) in place of DRAM. The MVME162P4's 16/32MB shared SDRAM is configurable to emulate either of the following memory models:

- ❑ 1MB, 4MB, 8MB, or 16MB shared parity-protected DRAM
- ❑ 4MB, 8MB, 16MB, or 32MB ECC-protected DRAM

The two memory controllers modeled in the Petra ASIC duplicate the functionality of the "parity memory controller" found in MC2 ASICs as well as that of the "single-bit error correcting/double-bit error detecting" memory controller found in MCECC ASICs. Board firmware will initialize the memory controller as appropriate.

If the Petra ASIC is supporting MVME1x2P4 functionality, firmware will enable the parity (MC2) memory controller model. If the Petra ASIC is supporting MVME1x2P2 functionality, firmware will enable either the parity or the MCECC memory controller model, depending on the board configuration. Board configuration is a function of switch settings and resistor population options.

S3 comes into play in the MC2 memory controller model. S3 is a four-segment slide switch whose lower three segments establish the size of the parity DRAM (segment 4 is not used.) Refer to the illustration and table below for specifics.



2734 0004

Table 1-4. MC2 DRAM Size Settings

S3 Segment 1	S3 Segment 2	S3 Segment 3	MC2 DRAM Size
ON	ON	ON	1MB
ON	ON	OFF	Not Defined
OFF	ON	ON	4MB
OFF	ON	OFF	8MB
OFF	OFF	ON	Disabled
OFF	OFF	OFF	16MB

Notes As shown in the preceding table, the Petra/MC2 interface supports parity DRAM emulations up to 16MB. For sizes beyond 16MB, it is necesary to use the MCECC memory model.

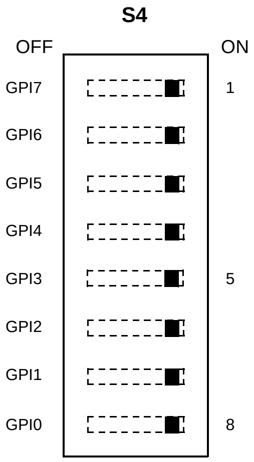
For access to the MCECC registers, you must first disable the MC2 interface by setting S3 to 001 (Off/Off/On).

General-Purpose Readable Switch (S4 Pin 5)

Switch S4 is similar in function to the general-purpose readable jumper headers found on earlier MVME162/172 series boards. S4 provides eight software-readable switch segments. These switches can be read as bits in a register (at address $FFF4202C) in the MC2 General-Purpose Inputs register in the Petra ASIC (refer to the *Programmer's Reference Guide* for details). Bit GPI7 is associated with switch segment 1; bit GPI0 is associated with switch segment 8. The bit values are read as a **0** when the switch is on, and as a **1** when the switch is off. The MVME162P4 is shipped from the factory with S4 set to all **0**s (all switches set to **ON**) except for GPI3, as diagrammed below.

If the MVME162Bug firmware is installed, three bits are user-definable (i.e., GPI2, GPI1, GPI0). If the MVME162Bug firmware is not installed, seven jumpers are user-definable (i.e., GPI7, GPI6, GPI5, GPI4, GPI2, GPI1, GPI0).

Note Switch segment 5 (GPI3) is reserved to select either the Flash memory map (switch set to **ON**) or the EPROM memory map (switch set to **OFF**). GPI3 is not user-definable.



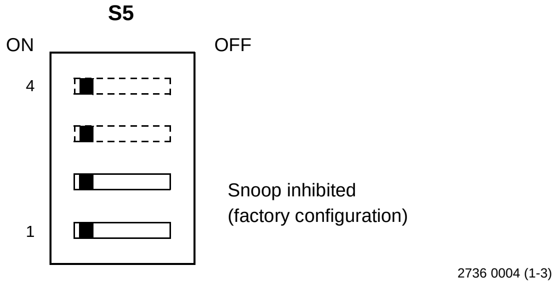
	162 BUG Installed (default)	User Code Installed
GPI7	USER-DEFINABLE	USER-DEFINABLE
GPI6	USER-DEFINABLE	USER-DEFINABLE
GPI5	USER-DEFINABLE	USER-DEFINABLE
GPI4	REFER TO DEBUG MANUAL	REFER TO DEBUG MANUAL
GPI3	ON=FLASH; OFF=EPROM	ON=FLASH; OFF=EPROM
GPI2	REFER TO DEBUG MANUAL	REFER TO DEBUG MANUAL
GPI1	REFER TO DEBUG MANUAL	REFER TO DEBUG MANUAL
GPI0	REFER TO DEBUG MANUAL	REFER TO DEBUG MANUAL

EPROM Selected
(factory configuration)

2735 0004

IP DMA Snoop Control (S5 Pins 1/2)

Segments 1 and 2 of switch S5 define the state of the snoop control bus when an IP DMA controller is local bus master. As shown in Table 1-5, S5 segment 1 controls Snoop Control signal 1 on the MC680x0 processor. S5 segment 2 controls Snoop Control signal 0. Setting a segment to **ON** produces a logical 0; setting it to **OFF** produces a logical 1.



S5 varies in function according to the type of processor installed. For MVME162P4 boards with an MC68040 processor, setting segments 1 and 2 of switch S5 to **OFF** or leaving both segments set to **ON** (the factory configuration) inhibits snooping. Enabling snooping requires one of two possible **ON**/**OFF** combinations, according to the operation desired. MVME172P4 boards with an MC68060 processor have different snoop functionality.

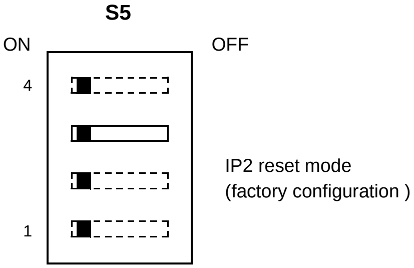
The following table lists the snoop operations represented by the settings of S5 with both types of processor. For further details, refer to the MC68040 or MC68060 microprocessor user's manuals listed in the *Related Documentation* appendix.

Table 1-5. Switch S5 Snoop Control Encoding

S5-1 (SC1)	S5-2 (SC0)	Requested Snoop Operation	
		MC68040	**MC68060**
On	On	Snoop disabled	Snoop disabled
On	Off	Source dirty, sink byte/word/longword	Snoop enabled
Off	On	Source dirty, invalidate line	Snoop disabled
Off	Off	Snoop disabled (Reserved)	Snoop enabled

IP Reset Mode (S5 Pin 3)

Segment 3 of switch S5 defines the IP controller model (IP1 or IP2) to be emulated when the board comes up. With S5 segment 3 set to **ON** (the factory configuration), the board initializes in IP2 mode. With S5 segment 3 set to **OFF**, the board initializes in IP1 mode.



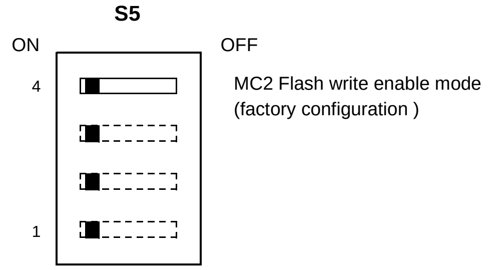
2736 0004 (2-3)

In IP2 mode, IP resets occur only in response to a direct software write or to a power-up reset; the IP reset control bit is not self-clearing.

In IP1 mode, the IP reset control bit clears itself after after a 1msec interval. IP resets may occur in response to a software write, a power-up reset, or a local bus reset. For details, refer to the *Programmer's Reference Guide* listed in "*Related Documentation*" in Appendix E.

Flash Write Enable Mode (S5 Pin 4)

Segment 4 of switch S5 defines the Flash memory controller model (MC1 or MC2) to be emulated when enabling or disabling Flash memory accesses on the MVME162P4 board. With S5 segment 4 set to **ON** (the factory configuration), the board initializes in MC2 mode. With S5 segment 4 set to **OFF**, the board initializes in MC1 mode.



2736 0004 (3-3)

In MC2 mode, writes to Flash memory are enabled or inhibited by a control bit at memory location $FFF42042. With the control bit set to **1**, Flash memory is write-enabled.

In MC1 mode, writes to Flash memory are enabled by a memory access to any location in the range $FFFCC000-$FFFCFFF. Writes to Flash memory are disabled by a memory access to any location in the range $FFFC8000-$FFFCBFFF. For details, refer to the *Programmer's Reference Guide* listed in Appendix E, *Related Documentation*.

MCECC DRAM Size (S6)

MVME1x2P4 boards use SDRAM (Synchronous DRAM) in place of DRAM. The MVME162P4's 16/32MB shared SDRAM is configurable to emulate either of the following memory models:

- ❑ 1MB, 4MB, 8MB, or 16MB shared parity-protected DRAM
- ❑ 4MB, 8MB, 16MB, or 32MB ECC-protected DRAM

The two memory controllers modeled in the Petra ASIC duplicate the functionality of the “parity memory controller” found in MC2 ASICs as well as that of the “single-bit error correcting/double-bit error detecting” memory controller found in MCECC ASICs. Board firmware will initialize the memory controller as appropriate.

If the Petra ASIC is supporting MVME1x2P4 functionality, firmware will enable the parity (MC2) memory controller model. If the Petra ASIC is supporting MVME1x2P2 functionality, firmware will enable either the parity or the MCECC memory controller model, depending on the board configuration. Board configuration is a function of switch settings and resistor population options.

S6 comes into play in the MCECC memory controller model. S6 is a four-segment slide switch whose lower three segments establish the size of the ECC DRAM (segment 4 is not used.) Refer to the illustration and table below for specifics.

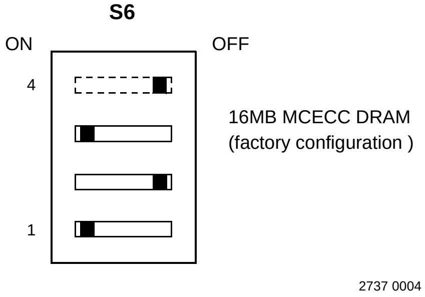


Table 1-6. MCECC DRAM Size Settings

S6 Segment 1	S6 Segment 2	S6 Segment 3	MCECC DRAM Size
ON	ON	ON	4MB
ON	ON	OFF	8MB
ON	OFF	ON	16MB
ON	OFF	OFF	32MB
OFF	ON	ON	64MB

Note For the MCECC memory model to be enabled, the MC2 emulation must be disabled. You disable the MC2 memory model by setting the MC2 DRAM size select switch (S3) to 110 (Off/Off/On).

The factory default setting for S6 is 32MB (On/Off/Off).

Installation Instructions

This section covers:

- ❑ Installation of IndustryPacks (IPs) on the MVME162P4
- ❑ Installation of the MVME162P4 in a VME chassis
- ❑ System considerations relevant to the installation. Ensure that an EPROM device is installed as needed. Before installing IndustryPacks, ensure that the serial ports and all header jumpers and configuration switches are set as appropriate.

IP Installation on the MVME162P4

The MVME162P4 accommodates up to four IndustryPack (IP) modules. Install the IP modules on the MVME162P4 as follows:

1. Each IP module has two 50-pin connectors that plug into two corresponding 50-pin connectors on the MVME162P4: J4/J5, J10/J11, J20/J21, J29/J30. See Figure 2-1 for the MVME162P4 connector locations.
   – Orient the IP module(s) so that the tapered connector shells mate properly. Plug IP_a into connectors J4 and J5; plug IP_b into J10 and J11. Plug IP_c into J20 and J21; plug IP_d into J29 and J30. If a double-sized IP is used, plug IP_ab into J4, J5, J10, and J11; plug IP_cd into J20, J21, J29, and J30.
2. Four additional 50-pin connectors (J7, J8, J26, and J27) are provided behind the MVME162P4 front panel for external cabling connections to the IP modules. There is a one-to-one correspondence between the signals on the cabling connectors and

the signals on the associated IP connectors (i.e., J8 has the same IP_a signals as J4; J7 has the same IP_b signals as J10; J27 has the same IP_c signals as J20; and J26 has the same IP_d signals as J29).

- Connect user-supplied 50-pin cables to J7, J8, J26, and J27 as needed. (Because of the varying requirements for each different kind of IP, Motorola does not supply these cables.)
- Bring the IP cables out the narrow slots in the MVME162P4 front panel and attach them to the appropriate external equipment, depending on the nature of the particular IP(s).

MVME162P4 Installation

With EPROM, SIM, and IP modules installed and headers or switches properly configured, proceed as follows to install the MVME162P4 in a VME chassis:

1. Turn all equipment power OFF and disconnect the power cable from the AC power source.

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 the chassis cover 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 MVME162P4.
   - If you intend to use the MVME162P4 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 MVME162P4 as system controller, it can occupy any unused double-height card slot.

4. Slide the MVME162P4 into the selected card slot. Be sure the module is seated properly in the P1 and P2 connectors on the backplane. Do not damage or bend connector pins.

5. Secure the MVME162P4 in the chassis with the screws provided, making good contact with the transverse mounting rails to minimize RF emissions.

6. Install the MVME712 series transition module in the front or the rear of the VME chassis. (To install an MVME712M, which has a double-wide front panel, you may need to shift other modules in the chassis.)

7. On the chassis backplane, remove the INTERRUPT ACKNOWLEDGE (IACK) and BUS GRANT (BG) jumpers from the header for the card slot occupied by the MVME162P4.

Note Some VME backplanes (for example, those used in Motorola "Modular Chassis" systems) have an autojumpering feature for automatic propagation of the IACK and BG signals. Step 7 does not apply to such backplane designs.

8. Connect the P2 Adapter Board or LCP2 Adapter Board and cable(s) to MVME162P4 backplane connector P2. This provides a connection point for terminals or other peripherals at the EIA-232-D serial ports, SCSI ports, and LAN Ethernet port. For information on installing the P2 or LCP2 Adapter Board and the MVME712 series transition module(s), refer to the corresponding user's manuals (the *Programmer's Reference Guide* provides some connection diagrams).

Note If you intend to use the MVME162P4 with Port B in an EIA-530 configuration or an EIA-485/EIA-422 configuration, do not install the P2 or LCP2 Adapter Board and the MVME712 series transition module. They are incompatible with the EIA-530 interface and the EIA-485/EIA-422 interface.

9. Connect the appropriate cable(s) to the panel connectors for the serial ports, SCSI port, and LAN Ethernet port.
   – Note that some cables are not provided with the MVME712 series transition module and must be made or purchased by the user. (Motorola recommends shielded cable for all peripheral connections to minimize radiation.)
10. Connect the peripheral(s) to the cable(s).
11. Install any other required VMEmodules in the system.
12. Replace the chassis cover.
13. Connect the power cable to the AC power source and turn the equipment power ON.

System Considerations

The MVME162P4 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 MVME162P4 may not operate properly without its main board connected to VMEbus backplane connectors P1 and P2.

Whether the MVME162P4 operates as a VMEbus master or 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 the *VMEchip2* chapter of the *Programmer's Reference Guide*. D8 and/or D16 devices in the system must be handled by the MC680x0/MC68LC0x0 software. For specifics, refer to the memory maps in the *Programmer's Reference Guide*, listed in Appendix E, *Related Documentation*.

The MVME162P4 contains shared onboard DRAM whose base address is software-selectable. Both the onboard processor and offboard VMEbus devices see this local DRAM at base physical address $00000000, as programmed by the MVME162Bug firmware. This may be changed via software to any other base address. Refer to the *Programmer's Reference Guide* for more information.

If the MVME162P4 tries to access offboard resources in a nonexistent location and is not system controller, and if the system does not have a global bus timeout, the MVME162P4 waits forever for the VMEbus cycle to complete. This will cause the system to lock up. There is only one situation in which the system might lack this global bus timeout: when the MVME162P4 is not the system controller and there is no global bus timeout elsewhere in the system.

Multiple MVME162P4s may be installed in a single VME chassis. In general, hardware multiprocessor features are supported.

Note If you are installing multiple MVME162P4s in an MVME945 chassis, do not install an MVME162P4 in slot 12. The height of the IP modules may cause clearance difficulties in that slot position.

Other MPUs on the VMEbus can interrupt, disable, communicate with, and determine the operational status of the processor(s). One register of the GCSR (global control/status register) set in the VMEchip2 ASIC includes four bits that function as location monitors to allow one MVME162P4 processor to broadcast a signal to any other MVME162P4 processors. All eight registers of the GCSR set are accessible from any local processor as well as from the VMEbus.

The following circuits are protected by solid-state fuses that open during overload conditions and reset themselves once the overload is removed:

- ❏ IndustryPack +5V (F1, F6)
- ❏ Remote reset connector +5V (F2)
- ❏ LAN AUI +12V (F3)
- ❏ IndustryPack ±12V (F4, F5)

The **FUSE** LED illuminates to indicate that +12 Vdc is available. When an MVME712M module is used, the yellow DS1 LED on the MVME712M illuminates when LAN power is available, signifying that the fuse is good. If the Ethernet transceiver fails to operate, check fuse F3.

The MVME162P4 provides SCSI terminator power through a 1A fuse (F1) located on the P2 Adapter Board or LCP2 Adapter Board. If the fuse is blown, the SCSI device(s) may function erratically or not at all. When the P2 Adapter Board is used with an MVME712M and the SCSI bus is connected to the MVME712M, the green DS2 LED on the MVME712M front panel illuminates when SCSI terminator power is available. If the green DS2 LED flickers during SCSI bus operation, check P2 Adapter Board fuse F1.

If a solid-state fuse opens, you will need to remove power for several minutes to let the fuse reset to a closed or shorted condition.

Serial Connections

The MVME162P4 uses a Zilog Z85230 serial port controller to implement the two serial communications interfaces. Each interface supports:

- ❑ CTS, DCD, RTS, and DTR control signals
- ❑ TXD and RXD transmit/receive data signals
- ❑ TXC and RXC synchronous clock signals

The Z85230 supports synchronous (SDLC/HDLC) and asynchronous protocols. The MVME162P4 hardware supports asynchronous serial baud rates of 110b/s to 38.4Kb/s.

For additional information on the MVME162P4 serial communications interface, refer to the *Z85230 Serial Communications Controller Product Brief* listed under *Manufacturer’s Documents* in the *Related Documentation* appendix. For additional information on the EIA-232-D interface, refer to the *EIA-232-D Standard*.

The following figures illustrate the signal relationships and signal connections in the various serial configurations available for ports A and B.

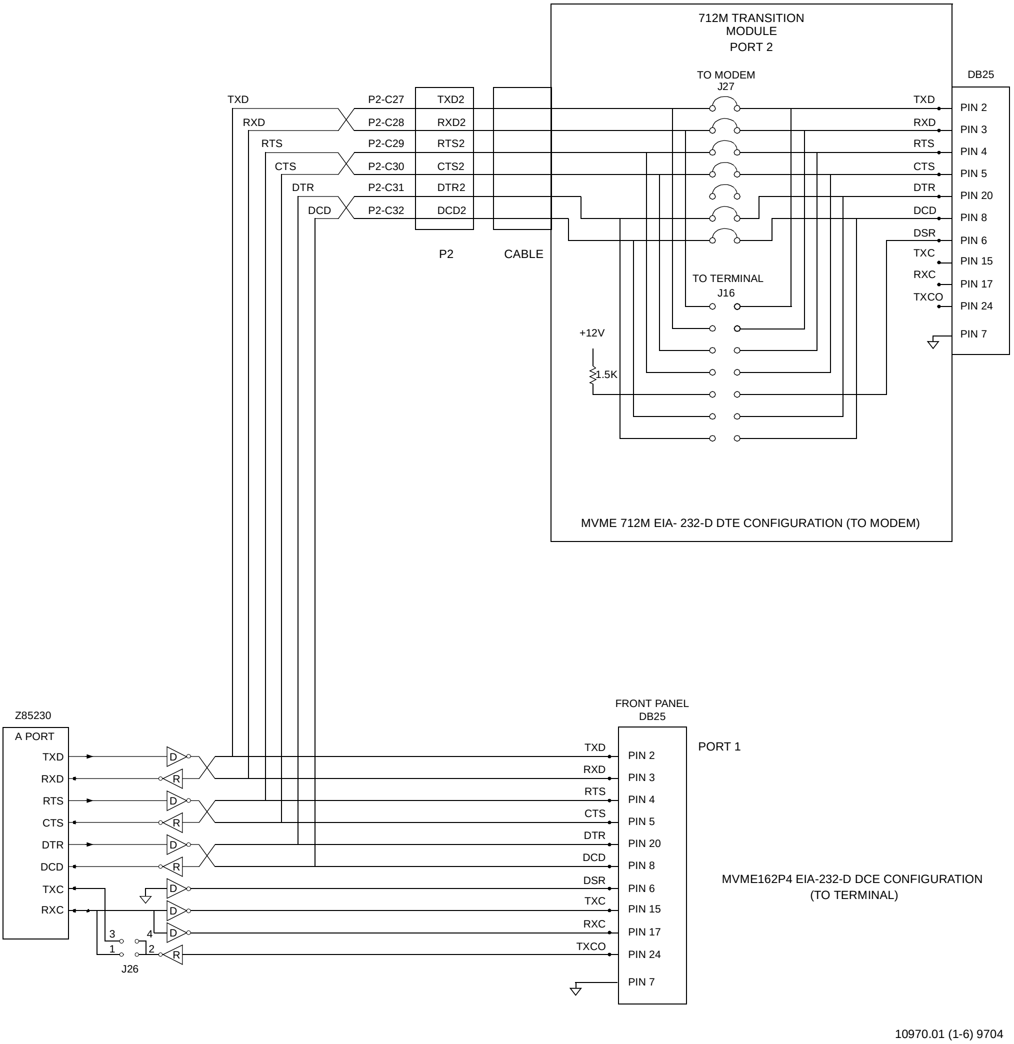


Figure 1-3. EIA-232-D Connections to MVME712M (Sheet 1 of 6)

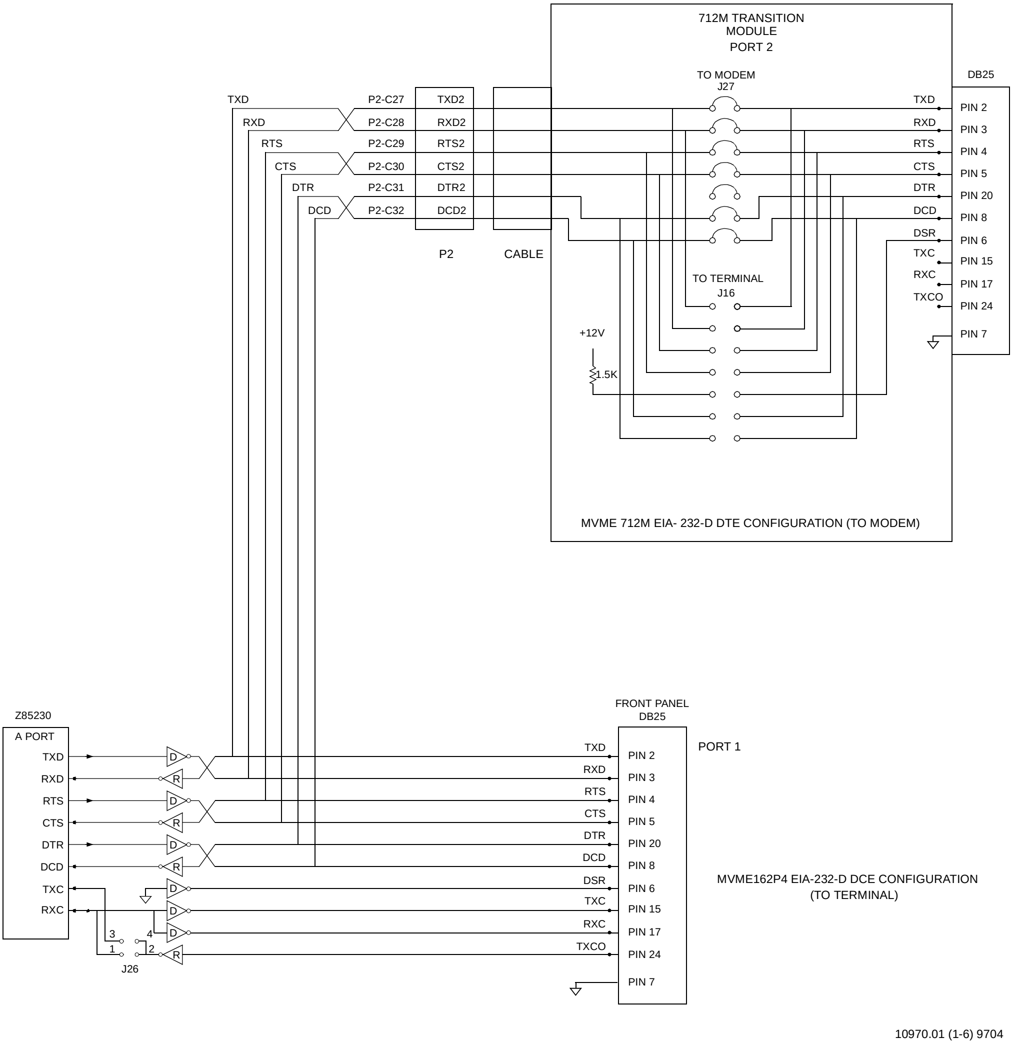


Figure 1-3. EIA-232-D Connections to MVME712M (Sheet 2 of 6)

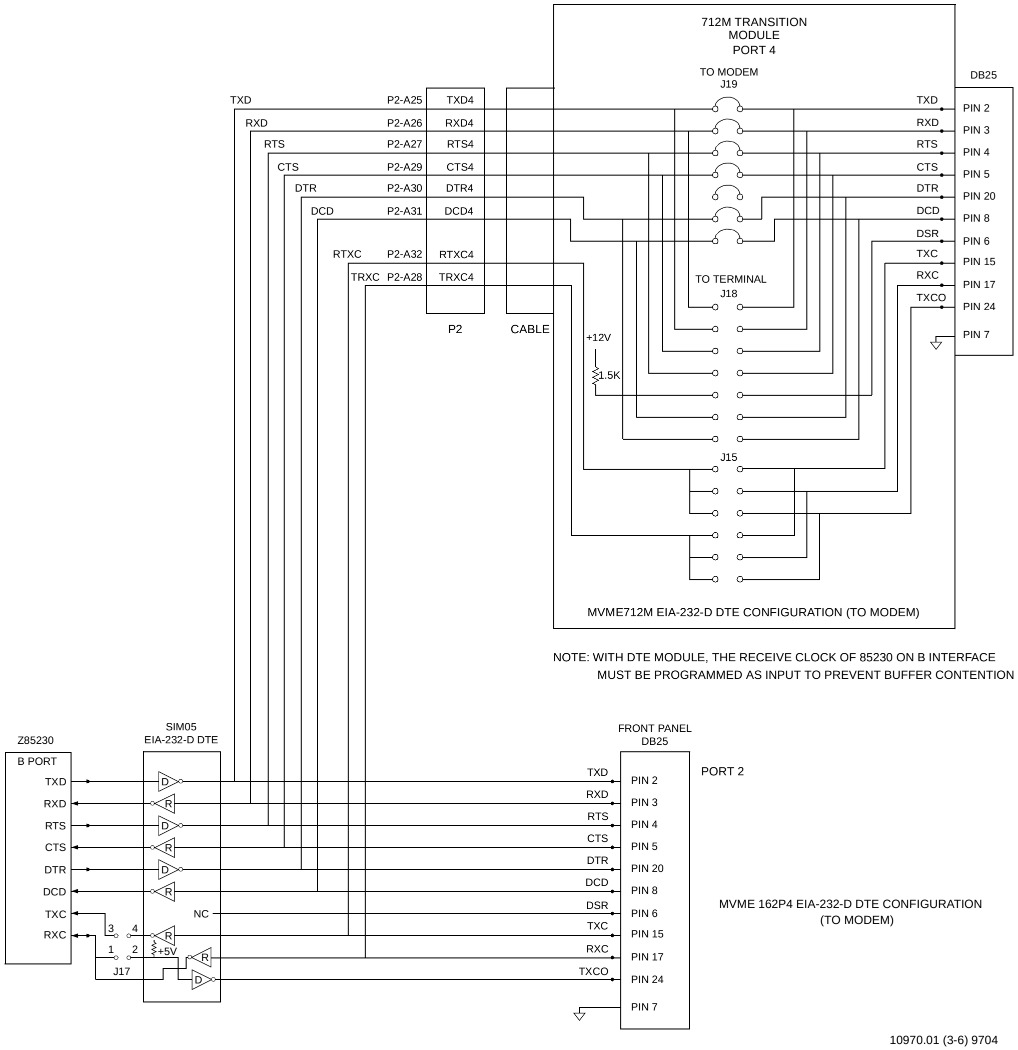


Figure 1-3. EIA-232-D Connections to MVME712M (Sheet 3 of 6)

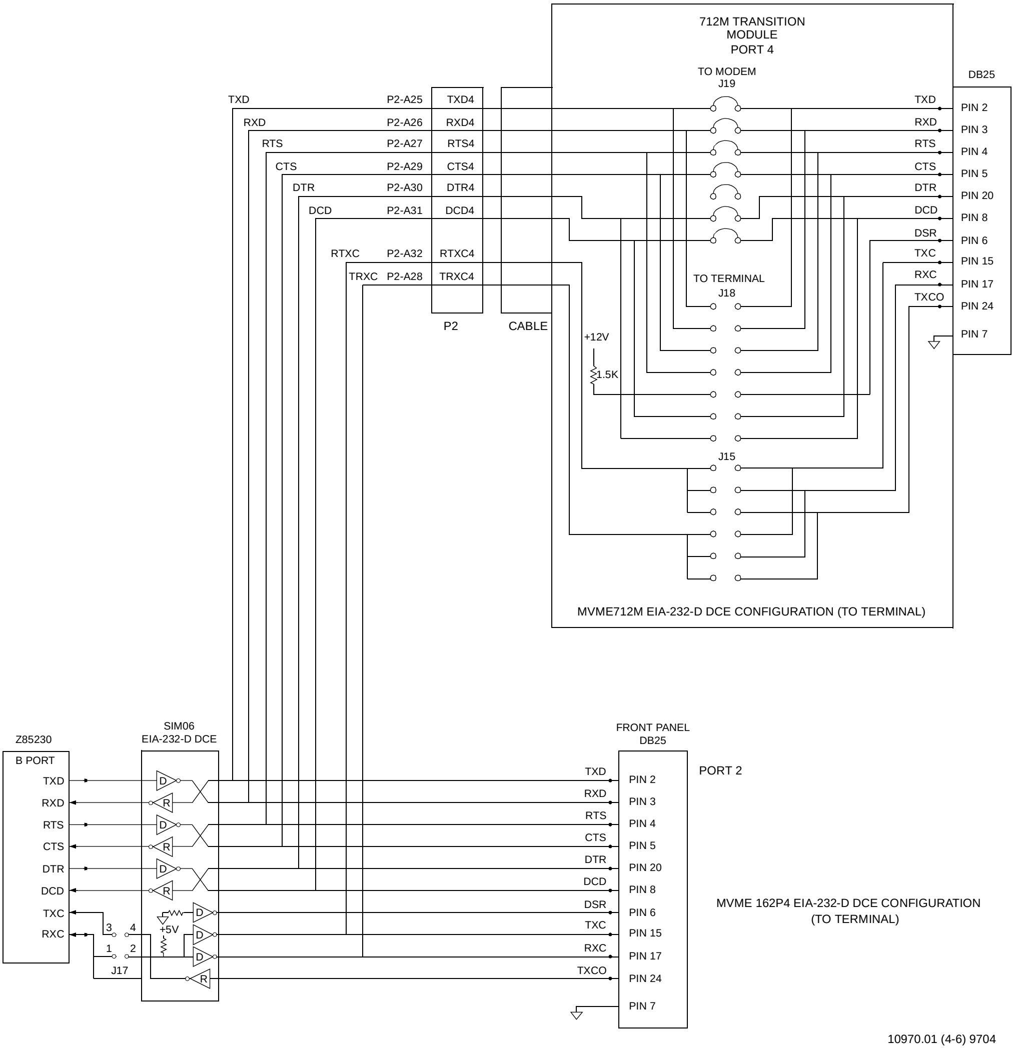
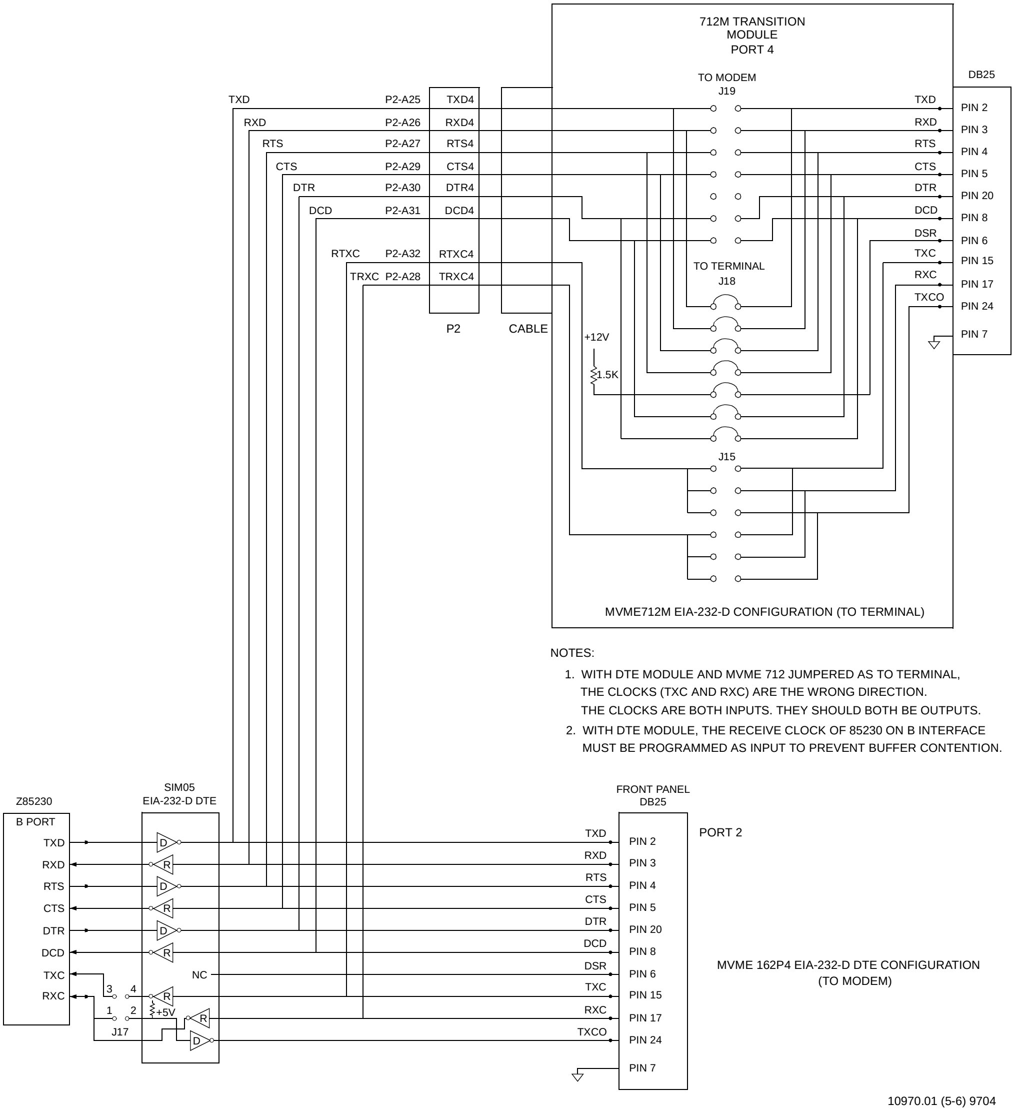


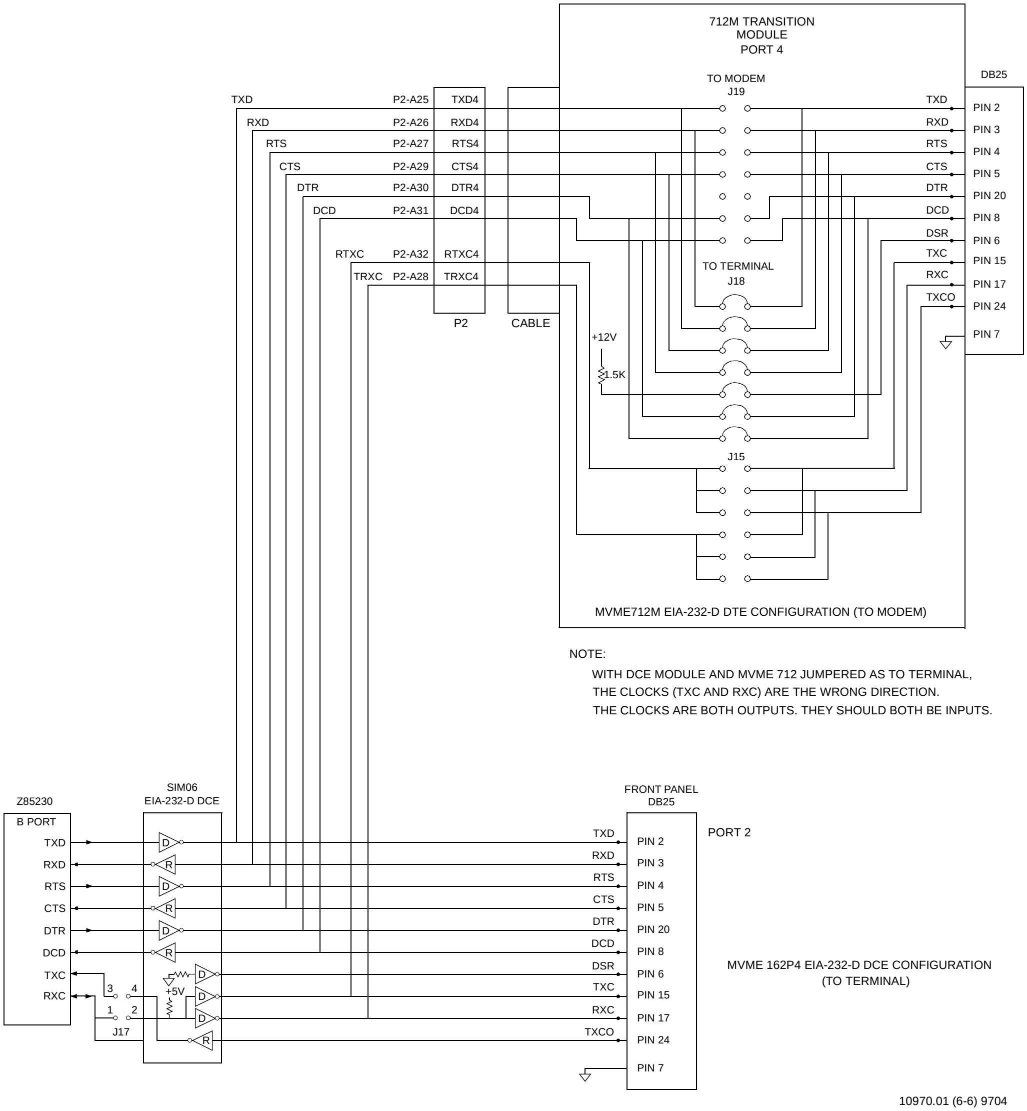
Figure 1-3. EIA-232-D Connections to MVME712M (Sheet 4 of 6)



NOTES:

1. WITH DTE MODULE AND MVME 712 JUMPERED AS TO TERMINAL, THE CLOCKS (TXC AND RXC) ARE THE WRONG DIRECTION. THE CLOCKS ARE BOTH INPUTS. THEY SHOULD BOTH BE OUTPUTS.
2. WITH DTE MODULE, THE RECEIVE CLOCK OF 85230 ON B INTERFACE MUST BE PROGRAMMED AS INPUT TO PREVENT BUFFER CONTENTION.

Figure 1-3. EIA-232-D Connections to MVME712M (Sheet 5 of 6)



NOTE:

WITH DCE MODULE AND MVME 712 JUMPERED AS TO TERMINAL,
THE CLOCKS (TXC AND RXC) ARE THE WRONG DIRECTION.
THE CLOCKS ARE BOTH OUTPUTS. THEY SHOULD BOTH BE INPUTS.

Figure 1-3. EIA-232-D Connections to MVME712M (Sheet 6 of 6)

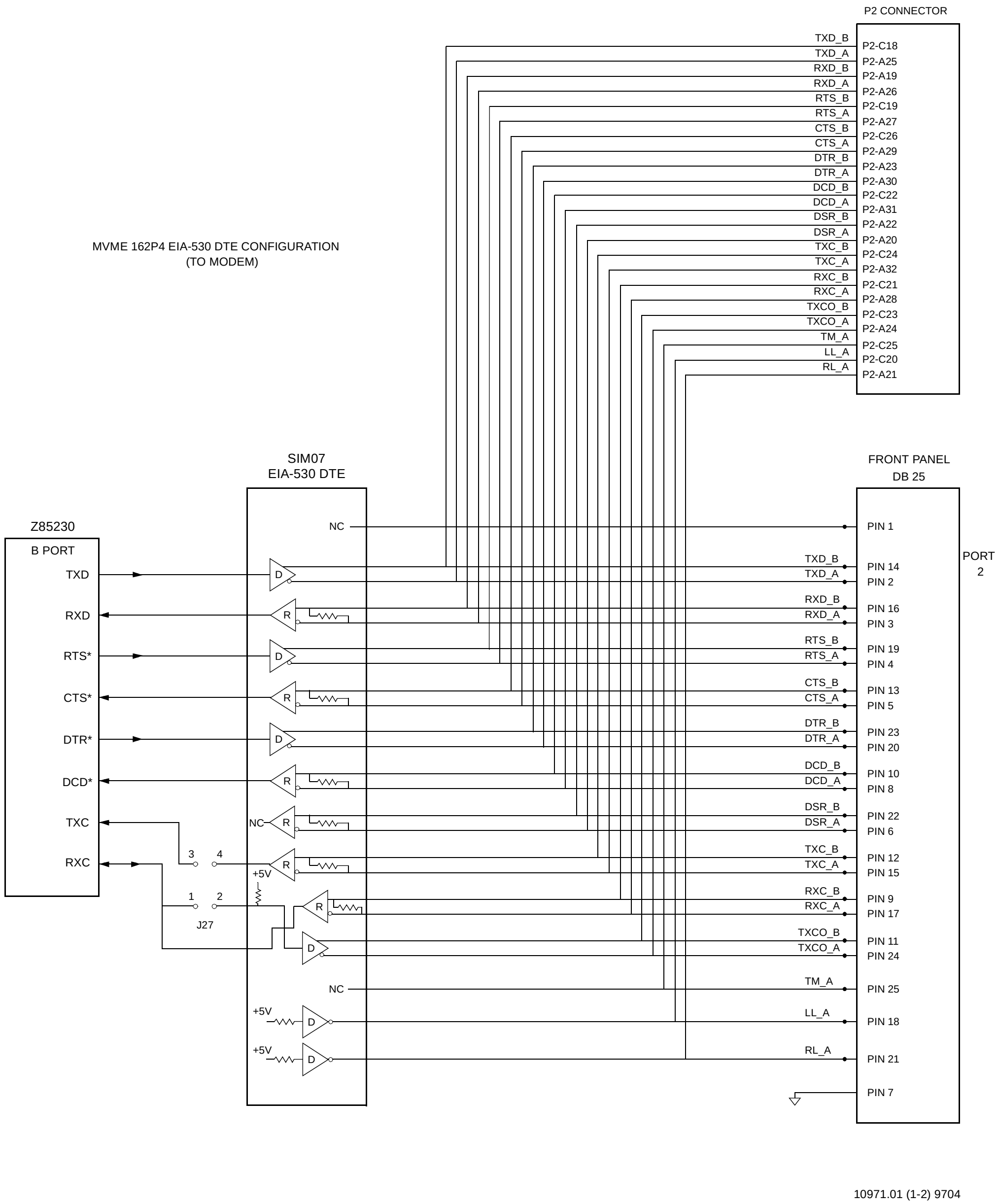


Figure 1-4. EIA-530 Connections (Sheet 1 of 2)

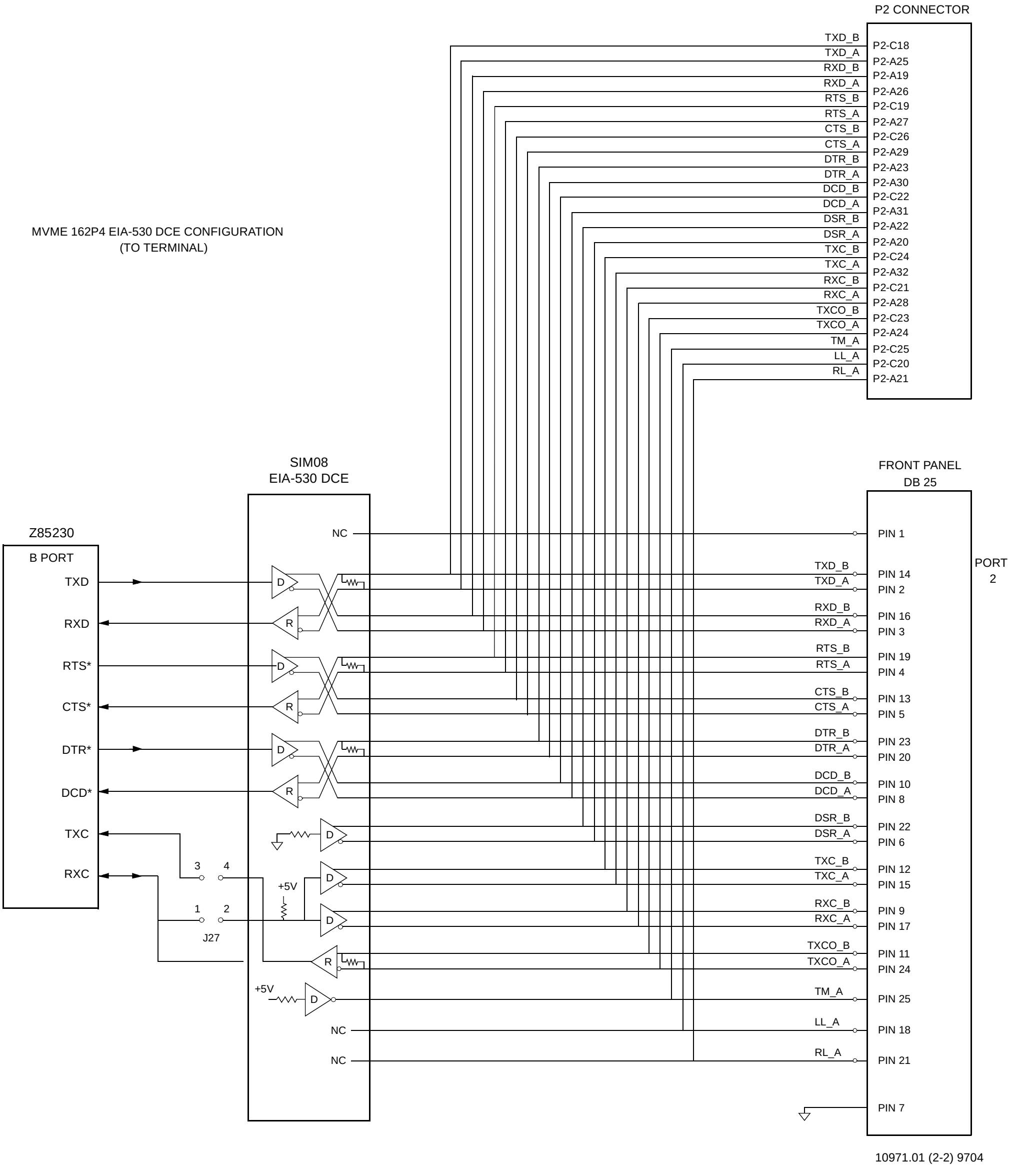
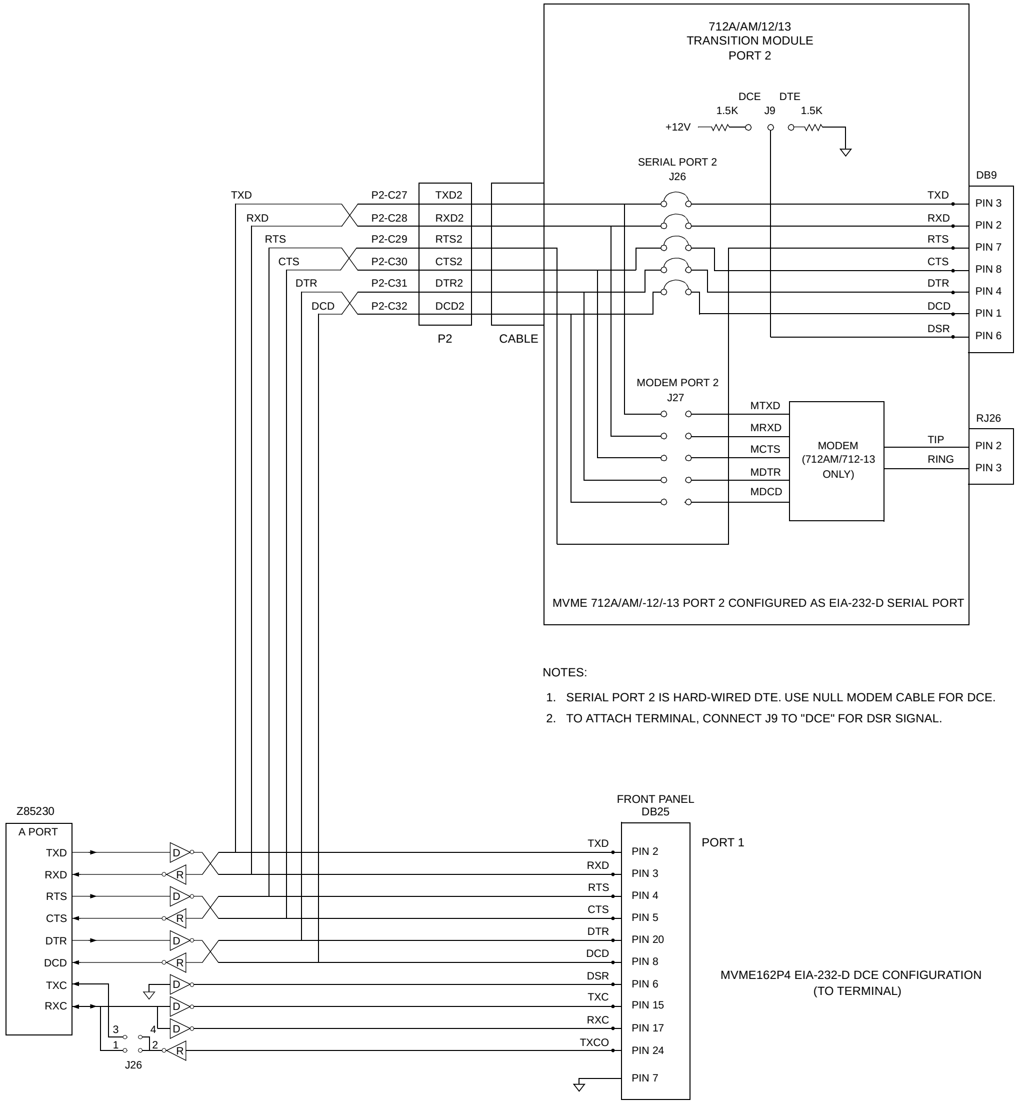


Figure 1-4. EIA-530 Connections (Sheet 2 of 2)

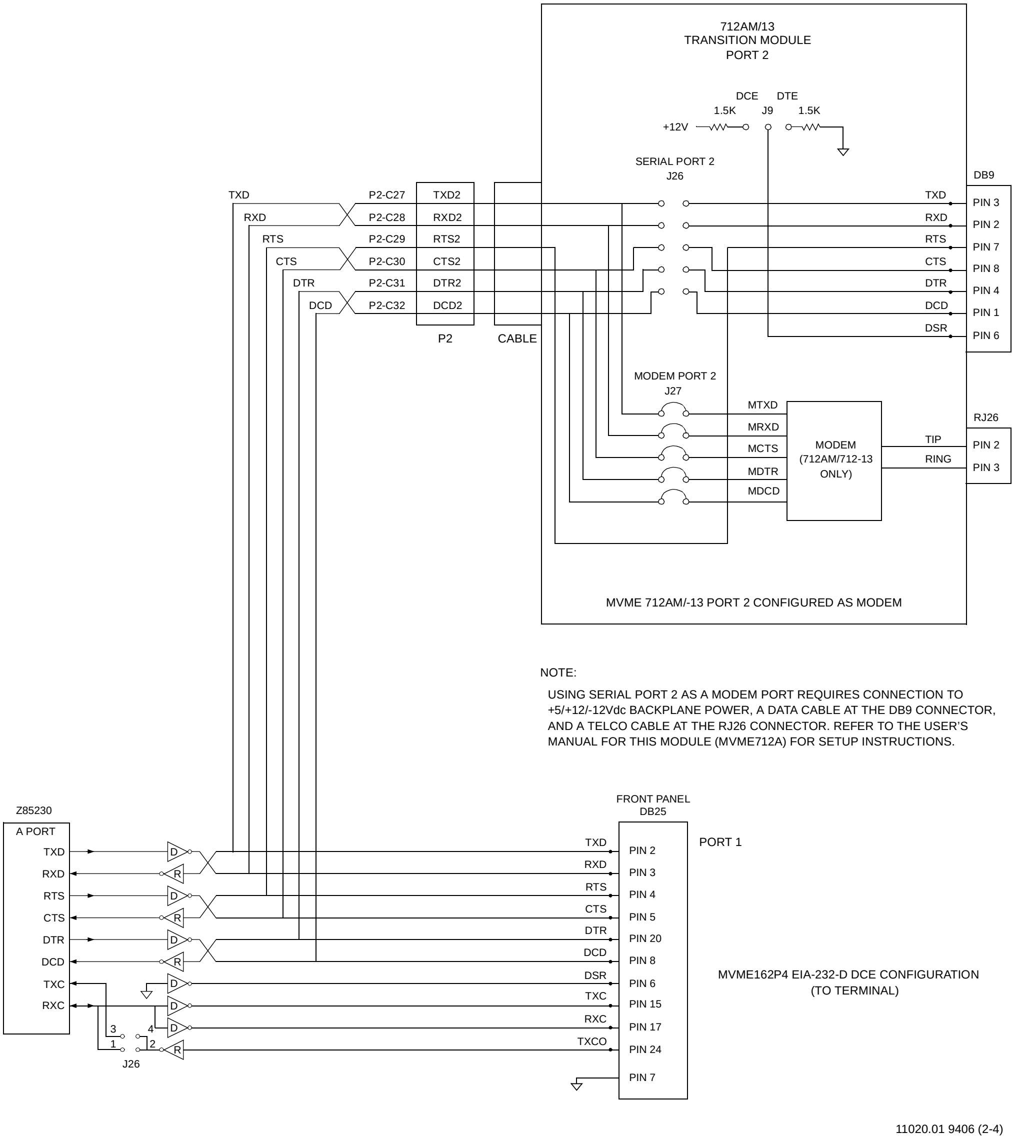


NOTES:

1. SERIAL PORT 2 IS HARD-WIRED DTE. USE NULL MODEM CABLE FOR DCE.
2. TO ATTACH TERMINAL, CONNECT J9 TO "DCE" FOR DSR SIGNAL.

11020.01 9406 (1-4)

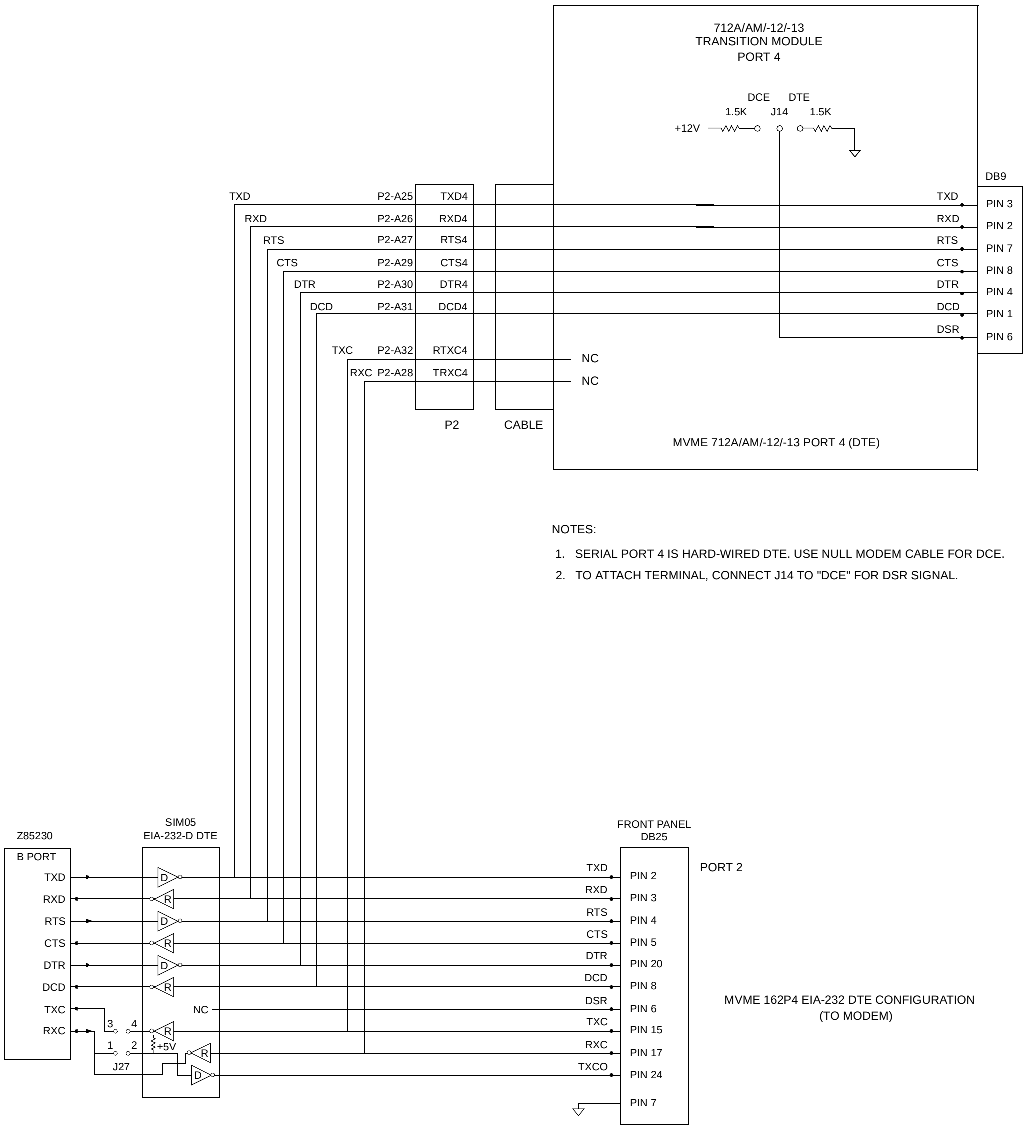
Figure 1-5. EIA-232-D Connections to MVME712A/AM/-12/-13 (Sheet 1 of 4)



NOTE:

USING SERIAL PORT 2 AS A MODEM PORT REQUIRES CONNECTION TO +5/+12/-12Vdc BACKPLANE POWER, A DATA CABLE AT THE DB9 CONNECTOR, AND A TELCO CABLE AT THE RJ26 CONNECTOR. REFER TO THE USER'S MANUAL FOR THIS MODULE (MVME712A) FOR SETUP INSTRUCTIONS.

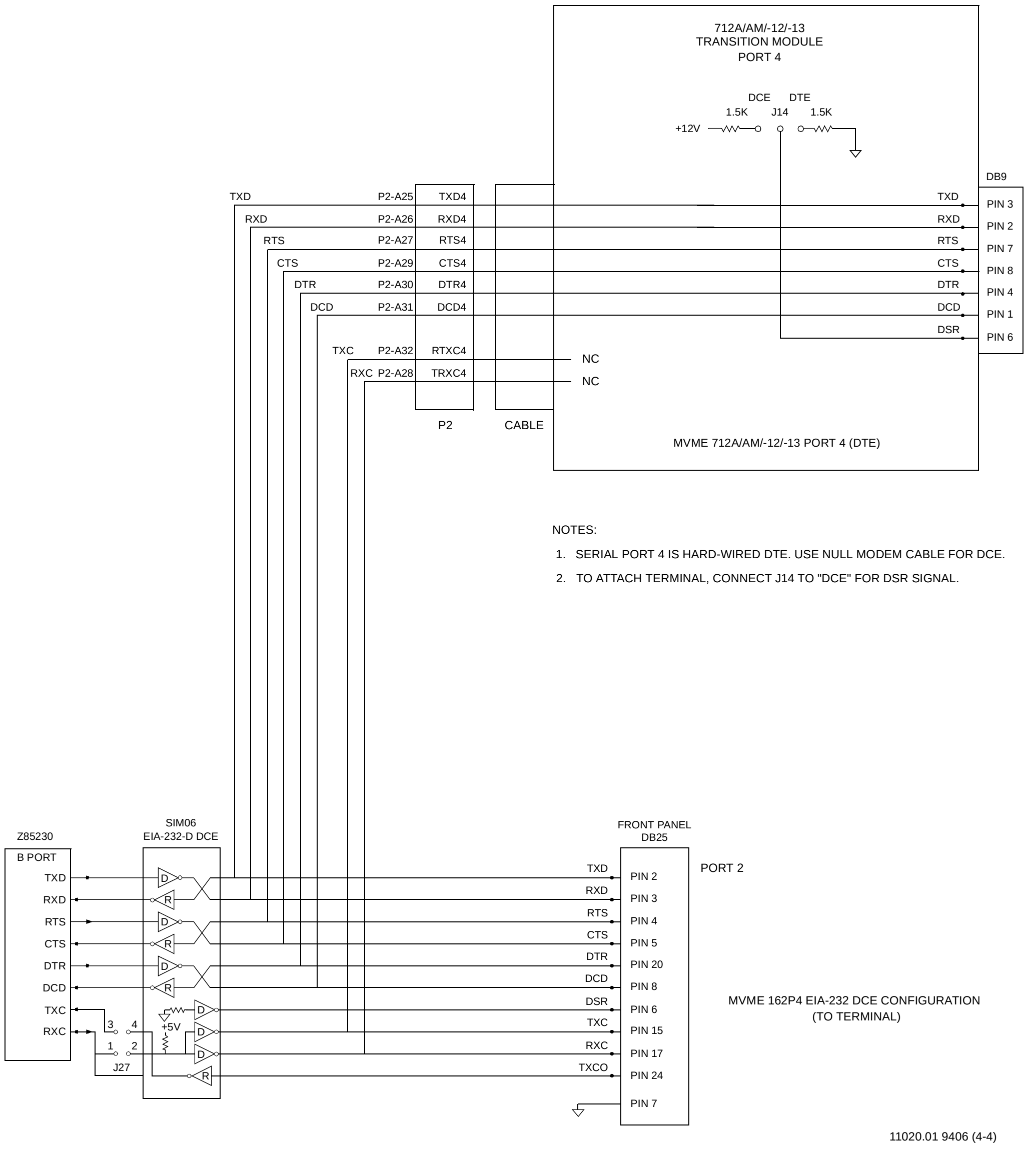
Figure 1-5. EIA-232-D Connections to MVME712A/AM/-12/-13 (Sheet 2 of 4)



NOTES:

1. SERIAL PORT 4 IS HARD-WIRED DTE. USE NULL MODEM CABLE FOR DCE.
2. TO ATTACH TERMINAL, CONNECT J14 TO "DCE" FOR DSR SIGNAL.

Figure 1-5. EIA-232-D Connections to MVME712A/AM/-12/-13 (Sheet 3 of 4)



NOTES:

1. SERIAL PORT 4 IS HARD-WIRED DTE. USE NULL MODEM CABLE FOR DCE.
2. TO ATTACH TERMINAL, CONNECT J14 TO "DCE" FOR DSR SIGNAL.

Figure 1-5. EIA-232-D Connections to MVME712A/AM/-12/-13 (Sheet 4 of 4)

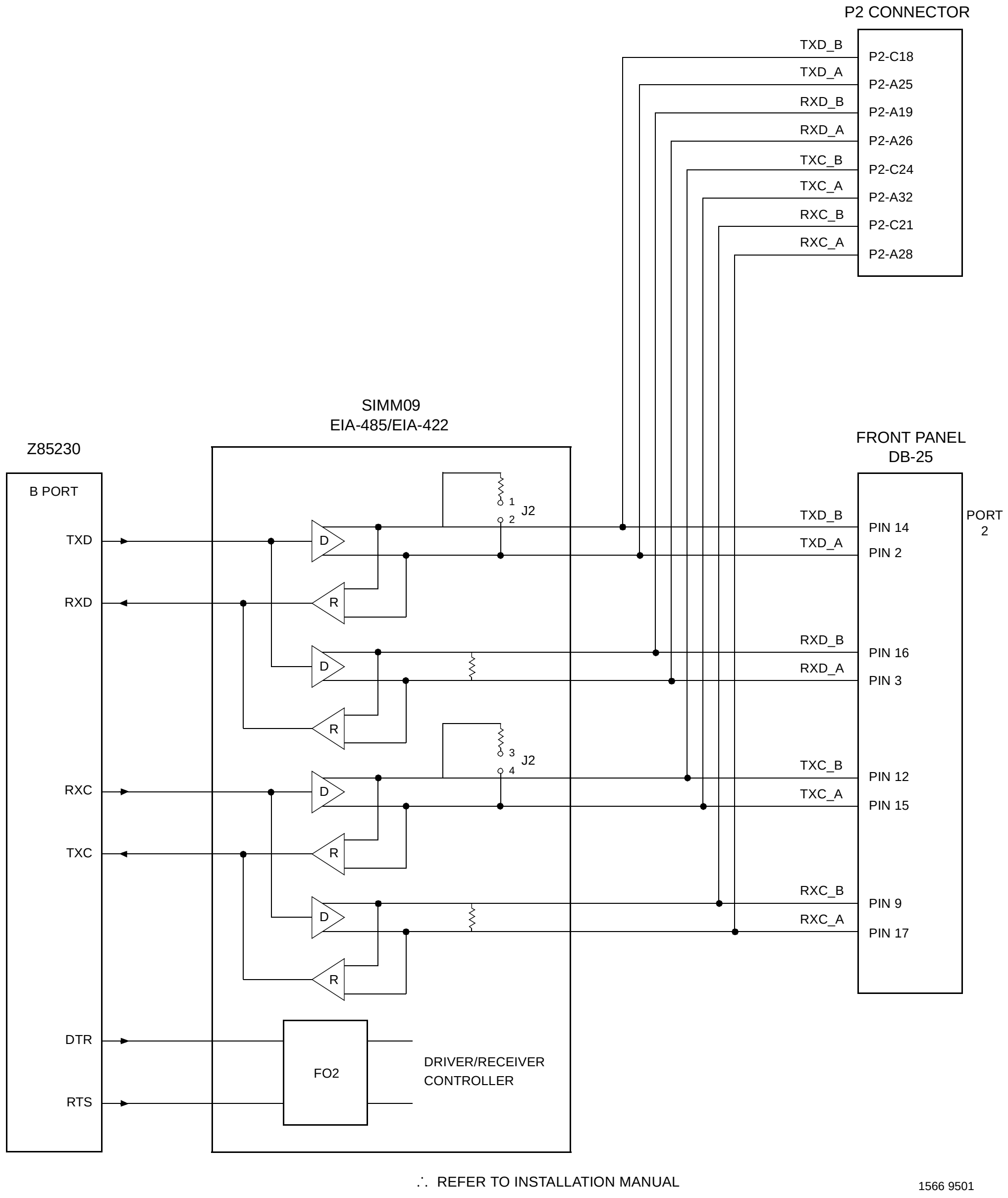


Figure 1-6. EIA-485/EIA-422 Connections

Startup and Operation 2

Introduction

This chapter provides information on powering up the MVME162P4 VME Embedded Controller after its installation in a system, and describes the functionality of the switches, status indicators, and I/O ports.

For programming information, consult the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide*, *Appendix E, Related Documentation*.

Front Panel Switches and Indicators

There are two switches (**ABORT** and **RESET**) and eight LEDs (**FAIL**, **STAT**, **RUN**, **SCON**, **LAN**, **FUSE**, **SCSI**, and **VME**) located on the MVME162P4 front panel.

Table 2-1. MVME162P4 Front Panel Controls

Control/Indicator	Function
Abort Switch (**ABORT**)	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 MVME162P4 Flash memory. The interrupter connected to the Abort switch is an edge-sensitive circuit, filtered to remove switch bounce.
Reset Switch (**RESET**)	Resets all onboard devices. Also drives a SYSRESET∗ signal if the MVME162P4 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 Local Control/Status Register (LCSR) in the VMEchip2 ASIC. For further details, refer to Chapter 4, *Functional Description*.
FAIL LED (DS1, red)	Board failure. Lights if a fault occurs on the MVME162P4 board.
STAT LED (DS2, amber)	CPU status. Lights if the processor enters a halt condition.
RUN LED (DS3, green)	CPU activity. Indicates that one of the local bus masters is executing a local bus cycle.

Table 2-1. MVME162P4 Front Panel Controls (Continued)

Control/Indicator	Function
SCON LED (DS4, green)	System controller. Lights when the VMEchip2 ASIC is functioning as VMEbus system controller.
LAN LED (DS5, green)	LAN activity. Lights when the LAN controller is functioning as local bus master.
FUSE LED (DS6, green)	Fuse OK. Indicates that +5Vdc, +12Vdc, and –12Vdc power is available to the LAN and SCSI interfaces and IP connectors.
SCSI LED (DS7, green)	SCSI activity. Lights when the SCSI controller is functioning as local bus master.
VME LED (DS8, green)	VME activity. Lights when the board is using the VMEbus or being accessed from the VMEbus.

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 MVME162P4’s MPU, hardware, and firmware along with the rest of the system.

The Flash-resident firmware initializes the devices on the MVME162P4 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 MVME162Bug 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 MVME162P4 hardware. The figure below charts the flow of the basic initialization sequence that takes place during system startup.

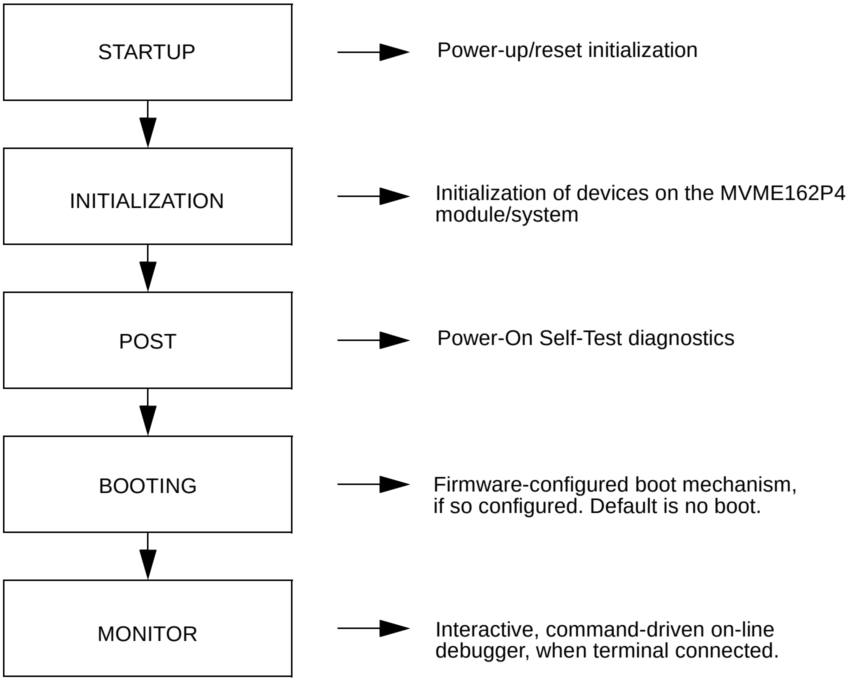

STARTUP
Power-up/reset initialization
INITIALIZATION
Initialization of devices on the MVME162P4 module/system
POST
Power-On Self-Test diagnostics
BOOTING
Firmware-configured boot mechanism, if so configured. Default is no boot.
MONITOR
Interactive, command-driven on-line debugger, when terminal connected.

Figure 2-1. MVME162P4/Firmware System Startup

Pre-Startup Checklist

Before you power up the MVME162P4 system, be sure that the following conditions exist:

1. Jumpers and/or configuration switches on the MVME162P4 VME Embedded Controller and associated equipment are set as required for your particular application.

2. The MVME162P4 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 MVME162P4 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 MVME162P4 comes with MVME162Bug 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 *MVME162P4 Configuration on page 1-4* and verify that jumpers and switches are configured as necessary for your particular application.

1. Configuration switch S4 on the MVME162P4 contains eight segments, which all affect the operation of the firmware. They are read as a register (at location $FFF4202C) in the Petra MC2 sector. (The *MVME1x2P4 VME Embedded Controller Programmer’s*

Reference Guide has additional information on the Petra MC2 emulation.) The bit values are read as a **0** when the corresponding switch segment is set to **ON**, or as a **1** when that segment is set to **OFF**.

The default configuration for S4 has S4 set to all **0**s (all switch segments set to **ON**) except for GPI3 (segment 5). The 162Bug firmware reserves/defines the four lower order bits (GPI0 to GPI3). Table 2-2 describes the bit assignments on S4.

2. Configure header J1 as appropriate for the desired system controller functionality (always system controller, never system controller, or self-regulating) on the MVME162P4.

3. Header J14 configures the IP bus clock for either 8MHz or the processor bus clock speed (25MHz or 32MHz for the MC68040 and MC68LC040). The factory configuration has a jumper installed on J14 pins 1-2, denoting an 8MHz clock. Verify that this setting is appropriate for your application.

Table 2-2. Software-Readable Switches

Bit No.	S4 Segment	Function
GPI0	8	When set to 1 (high), instructs the debugger to use local static RAM for its work page (variables, stack, vector tables, etc.).
GPI1	7	When set to 1 (high), instructs the debugger to use the default setup/operation parameters in ROM instead of the user setup/operation parameters in NVRAM. The effect is the same as pressing the **RESET** and **ABORT** switches simultaneously. This feature can be helpful in the event the user setup is corrupted or does not meet a sanity check. Refer to the **ENV** command description for the Flash/ROM defaults.
GPI2	6	Reserved for future use.
GPI3	5	When set to 0 (low), informs the debugger that it is executing out of Flash memory. When set to 1 (high), it informs the debugger that it is executing out of the PROM.

Table 2-2. Software-Readable Switches (Continued)

Bit No.	S4 Segment	Function
GPI4	4	Open to your application.
GPI5	3	Open to your application.
GPI6	2	Open to your application.
GPI7	1	Open to your application.

4. You may configure Port B of the Z85230 serial communications controller with a serial interface module (SIM), which you install at connector J15 on the MVME162P4 board. Five serial interface modules are available:

   – EIA-232-D DTE (SIMM05)

   – EIA-232-D DCE (SIMM06)

   – EIA-530 DTE (SIMM07)

   – EIA-530 DCE (SIMM08)

   – EIA-485, or EIA-422 DTE or DCE (all with SIMM09)

   For information on removing and/or installing a SIM, refer to Chapter 1.

5. Headers J16 and J17 configure serial ports 1 and 2 to drive or receive clock signals provided by the TXC and RXC signal lines. The MVME162P4 is factory-configured for asynchronous communication: it comes with no jumpers on J16 or J17. Refer to the instructions in Chapter 1 if your application requires configuring ports 1 and 2 for synchronous communication.

6. Header J19 enables or disables the IP bus strobe function on the MVME162P4. The factory configuration puts a jumper on J19 pins 1-2 to connect the Strobe∗ signal to the Petra/IP2 chip. Verify that the strobe line should be connected in your application.

7. The jumpers on header J22 establish the SRAM backup power source on the MVME162P4. The factory configuration uses VMEbus +5V standby voltage as the primary and secondary power source (the onboard battery is disconnected). Verify that this configuration is appropriate for your application.

8. The EPROM size select header, J23, should be jumpered between pins 1-2. This sets it up for a 4Mbit x 8 EPROM density, the factory default.

9. Header J24 defines the state of Flash memory write protection. The factory configuration has the jumper removed to inhibit writes to Flash. Verify that this setting is appropriate for your application.

10. Verify that the settings of configuration switches S3 (MC2 DRAM size), S5 (IP DMA snoop control, IP Reset mode, and Flash Write Enable mode), and S6 (MCECC DRAM size) are appropriate for your memory controller emulation.

11. Refer to the setup procedure for your particular chassis or system for details concerning the installation of the MVME162P4.

12. Connect the terminal to be used as the 162Bug system console to the default EIA-232-D port at Serial Port 1 on the front panel of the MVME162P4, *or* Serial Port 2 on the MVME712*x* series transition module. (For other connection options, refer to *Serial Connections* on page 1-26.) Set the terminal 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 162Bug 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.

13. If you have equipment (such as a host computer system and/or a serial printer) to connect to the other EIA-232-D port connectors (marked **SERIAL PORT** on the MVME712*x* transition module), connect the appropriate cables and configure the port(s) as detailed

in Step 12 above. After power-up, you can reconfigure the port(s) by programming the MVME162P4 Z85230 Serial Communications Controller (SCC) or by using the 162Bug **PF** command.

14. Power up the system. 162Bug executes some self-checks and displays the debugger prompt `162-Bug>` if the firmware is in Board mode.

    However, if the **ENV** command has placed 162Bug in System mode, the system performs a self-test and tries to autoboot. Refer to the **ENV** and **MENU** commands (Table 3-2).

    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.

15. Before using the MVME162P4 after the initial installation, set the date and time using the following command line structure:

    `162-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., September 7, 2000:

    `162-Bug>` **SET 0907001037**

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

Note If you wish to execute the debugger out of Flash and Flash does not contain 162Bug, you may copy the EPROM version of 162Bug to Flash memory. To copy the EPROM version of 162Bug to Flash memory, first set switch S4 segment 5 to **ON**, make sure that 162Bug is in Bug mode, and copy the EPROM contents to Flash memory with the **PFLASH** command as follows:

`162-Bug>` **PFLASH FF800000:80000 FFA00000**

Then slide switch S4 segment 5 back to **OFF**. (162Bug always executes from memory location FF800000; the setting of S4 determines whether that location is in EPROM or Flash.)

Autoboot

Autoboot is a software routine included in the 162Bug 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. Controllers, devices, and their LUNs are listed in Appendix D.

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.

Caution

Although you can use streaming tape to autoboot, the same power supply must be connected to the tape drive, the controller, and the MVME162P4. 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 MVME162P4 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

As shipped from the factory, 162Bug occupies the first quarter of Flash memory. This leaves the remainder of the Flash memory and the EPROM socket (XU1) available for your use.

Note You may wish to contact your Motorola sales office for assistance in using these resources.

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 MVME162P4 Flash/PROM 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.

For complete details on using the ROMboot function, refer to the *Debugging Package for Motorola 68K CISC CPUs User's Manual*, listed in Appendix E, *Related Documentation*.

Network Boot

Network Auto Boot is a software routine in the 162Bug Flash/EPROM 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. (Refer to Appendix C for default LUNs.)

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 MVME162P4 initiates a system reset. You can also initiate a reset by pressing and quickly releasing the **RESET** switch on the MVME162P4 front panel, or reset the board in software.

For details on resetting the MVME162P4 board through software, refer to the *MVME1x2P4 Embedded Controller Programmer's Reference Guide*.

Both “cold” and “warm” reset modes are available. By default, 162Bug is in “cold” mode. During *cold* resets, a total system initialization takes place, as if the MVME162P4 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 162Bug variables and tables are preserved, as well as the target state registers and breakpoints.

Note that when the MVME162P4 comes up in a cold reset, 162Bug runs in Board mode. Using the Environment (**ENV**) or **MENU** commands can make 162Bug 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 162Bug environment is ever lost (vector table is destroyed, stack corrupted, etc.).

Abort

Aborts are invoked by pressing and releasing the **ABORT** switch on the MVME162P4 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 162Bug package includes a set of hardware diagnostics for testing and troubleshooting the MVME162P4. 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 `162-Diag>` appears. Refer to the *Debugging Package for Motorola 68K CISC CPUs User's 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.

162Bug Firmware

3

Introduction

The 162Bug firmware is the layer of software just above the hardware. The firmware supplies the appropriate initialization for devices on the MVME162P4 board upon power-up or reset.

This chapter describes the basics of 162Bug 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 162Bug commands appears at the end of the chapter.

For complete user information about 162Bug, refer to the *Debugging Package for Motorola 68K CISC CPUs User's Manual* and to the *MVME162Bug Diagnostics User's Manual*, listed in Appendix E, *Related Documentation*.

162Bug Overview

The firmware for the M68000-based (68K) series of board and system level products has a common genealogy, deriving from the Bug firmware currently used on all Motorola M68000-based CPUs. The M68000 firmware version implemented on the MVME162P4 MC68040- or MC68LC040-based embedded controller is known as MVME162Bug, or 162Bug. It includes diagnostics for testing and configuring IndustryPack modules.

162Bug is a powerful evaluation and debugging tool for systems built around MVME162P4 CISC-based microcomputers. Facilities are available for loading and executing user programs under complete operator control for system evaluation. The 162Bug firmware provides a high degree of functionality, user friendliness, portability, and ease of maintenance.

162Bug 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

In addition, the TRAP #15 system calls make various 162Bug routines that handle I/O, data conversion, and string functions available to user programs.

162Bug consists of three parts:

- ❑ A command-driven user-interactive *software debugger*, described in this chapter. It is referred to here as “the debugger” or “162Bug”.
- ❑ A command-driven *diagnostic package* for the MVME162P4 hardware, referred to here as “the diagnostics”.
- ❑ A *user interface* or *debug/diagnostics monitor* that accepts commands from the system console terminal.

When using 162Bug, you operate out of either the *debugger directory* or the *diagnostic directory*.

- ❑ If you are in the debugger directory, the debugger prompt `162-Bug>` is displayed and you have all of the debugger commands at your disposal.
- ❑ If you are in the diagnostic directory, the diagnostic prompt `162-Diag>` is displayed and you have all of the diagnostic commands at your disposal as well as all of the debugger commands.

Because 162Bug is command-driven, it performs its various operations in response to user commands entered at the keyboard. When you enter a command, 162Bug 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 162Bug, depending on the outcome of the user program.

If you have used one or more of Motorola's other debugging packages, you will find the CISC 162Bug very similar. Some effort has also been made to improve the consistency of interactive commands. For example, delimiters between commands and arguments may be commas or spaces interchangeably.

162Bug Implementation

MVME162Bug is written largely in the "C" programming language, providing benefits of portability and maintainability. Where necessary, assembler has been used in the form of separately compiled modules containing only assembler code. No mixed-language modules are used.

Physically, 162Bug is contained in a 28F016SA Flash memory chip, providing 512KB (128K longwords) of storage. Optionally, the 162Bug firmware can be loaded and executed in a single a single 27C040 DIP EPROM installed in socket XU1. The executable code is checksummed at every power-on or reset firmware entry, and the result (which includes a precalculated checksum contained in the memory devices) is tested for an expected zero. Users are cautioned against modification of the memory devices unless precautions for re-checksumming are taken.

Note MVME162P4 boards ordered without the VMEbus interface are shipped with Flash memory blank (the factory uses the VMEbus to program the Flash memory with debugger code. To use the 162Bug package, be sure that switch S4 segment 5 is configured to select the EPROM memory map.

If you subsequently wish to run the debugger from Flash memory, you must first initialize Flash memory with the **PFLASH** command, then reconfigure S4. Refer to Step 15 (Note) under *Bringing up the Board* on page 2-4 for further details.

Memory Requirements

The program portion of 162Bug is approximately 512KB of code, consisting of download, debugger, and diagnostic packages contained entirely in Flash memory or in EPROM.

The 162Bug firmware executes from address $FF800000 whether in Flash or EPROM. If you set switch S4 segment 5 to **ON**, the address spaces of the Flash and EPROM are swapped. For MVME162P-2*xx* and -3*xx* series boards (MVME162P4), the factory ship configuration *except* in the no-VMEbus case has switch S4 segment 5 set to **OFF** (162Bug operating out of Flash).

The 162Bug initial stack completely changes 8KB of SRAM memory at addresses $FFE0C000 through $FFE0DFFF, at power-up or reset.

Table 3-1. Memory Offsets with 162Bug

Type of Memory Present	Default DRAM Base Address	Default SRAM Base Address
4/8/16/32MB shared DRAM (SDRAM). Appears as parity memory at 1/8/16MB, ECC at 32MB.	$00000000	$FFE00000 (onboard SRAM)

The shared DRAM can be modeled as ECC or parity type, as indicated above.

The 162Bug firmware requires 2KB of NVRAM for storage of board configuration, communication, and booting parameters. This storage area begins at $FFFC16F8 and ends at $FFFC1EF7.

162Bug requires a minimum of 64KB of contiguous read/write memory to operate. The **ENV** command controls where this block of memory is located. Regardless of where the onboard RAM is located, the first 64KB is used for 162Bug stack and static variable space and the rest is reserved as user space. Whenever the MVME162P4 is reset, the target PC is initialized to the address corresponding to the beginning of the user space, and the target stack pointers are initialized to addresses within the user space, with the target Interrupt Stack Pointer (ISP) set to the top of the user space.

Using 162Bug

162Bug is command-driven; it performs its various operations in response to commands that you enter at the keyboard. When the `162-Bug>` prompt appears on the terminal screen, the debugger is ready to accept debugger commands. When the `162-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 the carriage return is entered. This allows you to correct entry errors, if necessary, with the control characters described in the *Debugging Package for Motorola 68K CISC CPUs User's Manual*, Chapter, listed in Appendix E, *Related Documentation*.

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 *Debugging Package for Motorola 68K CISC CPUs User's Manual*, Chapter 5, listed in Appendix E, *Related Documentation*).

A debugger command is made up of the following parts:

- ❑ The command name, either uppercase or lowercase (for example, **MD** or **md**).
- ❑ A port number (if the command is set up to work with more than one port).
- ❑ 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 162Bug debugger commands are summarized in the following table. The commands are described in detail in the *Debugging Package for Motorola 68K CISC CPUs User's Manual*.

Table 3-2. Debugger Commands

Command	Description
AB	Automatic Bootstrap Operating System
NOAB	No Autoboot
AS	One Line Assembler
BC	Block of Memory Compare
BF	Block of Memory Fill
BH	Bootstrap Operating System and Halt
BI	Block of Memory Initialize
BM	Block of Memory Move
BO	Bootstrap Operating System
BR	Breakpoint Insert
NOBR	Breakpoint Delete
BS	Block of Memory Search
BV	Block of Memory Verify
CM	Concurrent Mode
NOCM	No Concurrent Mode
CNFG	Configure Board Information Block
CS	Checksum
DC	Data Conversion
DMA	DMA Block of Memory Move
DS	One Line Disassembler

Table 3-2. Debugger Commands (Continued)

Command	Description
DU	Dump S-records
ECHO	Echo String
ENV	Set Environment to Bug/Operating System
GD	Go Direct (Ignore Breakpoints)
GN	Go to Next Instruction
GO	Go Execute User Program
GT	Go to Temporary Breakpoint
HE	Help
IOC	I/O Control for Disk
IOI	I/O Inquiry
IOP	I/O Physical (Direct Disk Access)
IOT	I/O "Teach" for Configuring Disk Controller
IRQM	Interrupt Request Mask
LO	Load S-records from Host
MA	Macro Define/Display
NOMA	Macro Delete
MAE	Macro Edit
MAL	Enable Macro Expansion Listing
NOMAL	Disable Macro Expansion Listing
MAW	Save Macros
MAR	Load Macros
MD	Memory Display
MENU	Menu
MM	Memory Modify
MMD	Memory Map Diagnostic
MS	Memory Set
MW	Memory Write
NAB	Automatic Network Boot Operating System
NBH	Network Boot Operating System and Halt

Table 3-2. Debugger Commands (Continued)

Command	Description
NBO	Network Boot Operating System
NIOC	Network I/O Control
NIOP	Network I/O Physical
NIOT	Network I/O Teach
NPING	Network Ping
OF	Offset Registers Display/Modify
PA	Printer Attach
NOPA	Printer Detach
PF	Port Format
NOPF	Port Detach
PFLASH	Program FLASH Memory
PS	Put RTC Into Power Save Mode for Storage
RB	ROMboot Enable
NORB	ROMboot Disable
RD	Register Display
REMOTE	Connect the Remote Modem to CSO
RESET	Cold/Warm Reset
RL	Read Loop
RM	Register Modify
RS	Register Set
SD	Switch Directories
SET	Set Time and Date
SYM	Symbol Table Attach
NOSYM	Symbol Table Detach
SYMS	Symbol Table Display/Search
T	Trace
TA	Terminal Attach
TC	Trace on Change of Control Flow
TIME	Display Time and Date

Table 3-2. Debugger Commands (Continued)

Command	Description
TM	Transparent Mode
TT	Trace to Temporary Breakpoint
VE	Verify S-Records Against Memory
VER	Display Revision/Version
WL	Write Loop

Modifying the Environment

You can use the factory-installed debug monitor, 162Bug, to modify certain parameters contained in the MVME162P4'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 162Bug command **CNFG** to change those parameters.
- Use the 162Bug command **ENV** to change configurable 162Bug parameters in NVRAM.

The **CNFG** and **ENV** commands are both described in the *Debugging Package for Motorola 68K CISC CPUs User's Manual,* listed in Appendix E, *Related Documentation*. 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 162Bug 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

MVME162P4 board. (Note that although no memory mezzanine is present on MVME1x2P4 series boards, the on-board memory is modeled as such for backward compatibility.)

The board structure for the MVME162P4 is as follows:

```
162-Bug>cnfg
Board (PWA) Serial Number = "     "
Board Identifier = "     "
Artwork (PWA) Identifier = "     "
MPU Clock Speed = "     "
Ethernet Address = 08003E200000
Local SCSI Identifier = "     "
Parity Memory Mezzanine Artwork (PWA) Identifier = "     "
Parity Memory Mezzanine (PWA) Serial Number = "     "
Static Memory Mezzanine Artwork (PWA) Identifier = "     "
Static Memory Mezzanine (PWA) Serial Number = "     "
ECC Memory Mezzanine #1 Artwork (PWA) Identifier = "     "
ECC Memory Mezzanine #1 (PWA) Serial Number = "     "
ECC Memory Mezzanine #2 Artwork (PWA) Identifier = "     "
ECC Memory Mezzanine #2 (PWA) Serial Number = "     "
Serial Port 2 Personality Artwork (PWA) Identifier = "     "
Serial Port 2 Personality Module (PWA) Serial Number = "     "
IndustryPack A Board Identifier = "     "
IndustryPack A (PWA) Serial Number = "     "
IndustryPack A Artwork (PWA) Identifier = "     "
IndustryPack B Board Identifier = "     "
IndustryPack B (PWA) Serial Number = "     "
IndustryPack B Artwork (PWA) Identifier = "     "
IndustryPack C Board Identifier = "     "
IndustryPack C (PWA) Serial Number = "     "
IndustryPack C Artwork (PWA) Identifier = "     "
IndustryPack D Board Identifier = "     "
IndustryPack D (PWA) Serial Number = "     "
IndustryPack D Artwork (PWA) Identifier = "     "
162-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 *MVME162P VME Embedded Controller Programmer's Reference Guide* (listed in Appendix E, *Related Documentation*) for the actual location and other information about the Board Information Block. Refer to the *Debugging Package for Motorola 68K CISC CPUs User's Manual* for a **CNFG** description and examples.

ENV - Set Environment

Use the **ENV** command to view and/or configure interactively all 162Bug operational parameters that are kept in Non-Volatile RAM (NVRAM).

Refer to the *Debugging Package for Motorola 68K CISC CPUs User's Manual* for a description of the use of **ENV**. Additional information on registers in the MVME162P4 that affect these parameters appears in your *MVME1x2P4 VME Embedded Controller 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 MVME162P4, you may wish to use **env;d** <CR> to restore the factory defaults as a troubleshooting aid (see Appendix B).

Configuring the 162Bug Parameters

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

Table 3-3. ENV Command Parameters

ENV Parameter and Options	Default	Meaning of Default
Bug or System environment [B/S]	B	Bug mode
Field Service Menu Enable [Y/N]	N	Do not display field service menu.
Remote Start Method Switch [G/M/B/N]	B	Use both methods [Global Control and Status Register (GCSR) in the VMEchip2, and Multiprocessor Control Register (MPCR) in shared RAM] to pass and execute cross-loaded programs.
Probe System for Supported I/O Controllers [Y/N]	Y	Accesses will be made to the appropriate system buses (e.g., VMEbus, local MPU bus) to determine presence of supported controllers.
Negate VMEbus SYSFAIL∗ Always [Y/N]	N	Negate VMEbus SYSFAIL∗ after successful completion or entrance into the bug command monitor.
Local SCSI Bus Reset on Debugger Startup [Y/N]	N	No local SCSI bus reset on debugger startup.
Local SCSI Bus Negotiations Type [A/S/N]	A	Asynchronous negotiations.
Industry Pack Reset on Debugger Startup [Y/N]	Y	IP modules are reset on debugger startup.
Ignore CFGA Block on a Hard Disk Boot [Y/N]	Y	Configuration Area (CFGA) Block contents are disregarded at boot (hard disk only).
Auto Boot Enable [Y/N]	N	Auto Boot function is disabled.
Auto Boot at power-up only [Y/N]	Y	Auto Boot is attempted at power-up reset only.
Auto Boot Controller LUN	00	Specifies LUN of disk/tape controller module currently supported by the Bug. Default is $0.
Auto Boot Device LUN	00	Specifies LUN of disk/tape device currently supported by the Bug. Default is $0.

Table 3-3. ENV Command Parameters (Continued)

ENV Parameter and Options	Default	Meaning of Default
Auto Boot Abort Delay	15	The time in seconds that the Auto Boot sequence will delay before starting the boot. The delay gives you the option of stopping the boot by use of the Break key. The time span is 0-255 seconds.
Auto Boot Default String [Y(NULL String)/(String)]		You may specify a string (filename) to pass on to the code being booted. Maximum length is 16 characters. Default is the null string.
ROM Boot Enable [Y/N]	N	ROMboot function is disabled.
ROM Boot at power-up only [Y/N]	Y	ROMboot is attempted at power-up only.
ROM Boot Enable search of VMEbus [Y/N]	N	VMEbus address space will not be accessed by ROMboot.
ROM Boot Abort Delay	00	The time in seconds that the ROMboot sequence will delay before starting the boot. The delay gives you the option of stopping the boot by use of the Break key. The time span is 0-255 seconds.
ROM Boot Direct Starting Address	FF800000	First location tested when the Bug searches for a ROMboot module.
ROM Boot Direct Ending Address	FFDFFFFC	Last location tested when the Bug searches for a ROMboot module.
Network Auto Boot Enable [Y/N]	N	Network Auto Boot function is disabled.
Network Auto Boot at power-up only [Y/N]	Y	Network Auto Boot is attempted at power-up reset only.
Network Auto Boot Controller LUN	00	Specifies LUN of a disk/tape controller module currently supported by the Bug. Default is $0.
Network Auto Boot Device LUN	00	Specifies LUN of a disk/tape device currently supported by the Bug. Default is $0.

Table 3-3. ENV Command Parameters (Continued)

ENV Parameter and Options	Default	Meaning of Default
Network Auto Boot Abort Delay	5	The time in seconds that the Network Boot sequence will delay before starting the boot. The delay gives you the option of stopping the boot by use of the Break key. The time span is 0-255 seconds.
Network Autoboot Configuration Parameters Pointer (NVRAM)	00000000	The address where the network interface configuration parameters are to be saved in NVRAM; these are the parameters necessary to perform an unattended network boot.
Memory Search Starting Address	00000000	Where the Bug begins to search for a work page (a 64KB block of memory) to use for vector table, stack, and variables. This must be a multiple of the debugger work page, modulo $10000 (64KB). In a multi-controller environment, each MVME162P4 board could be set to start its work page at a unique address to allow multiple debuggers to operate simultaneously.
Memory Search Ending Address	00100000	Top limit of the Bug's search for a work page. If no 64KB contiguous block of memory is found in the range specified by Memory Search Starting Address and Memory Search Ending Address parameters, the bug will place its work page in the onboard static RAM on the MVME162P4. Default Memory Search Ending Address is the calculated size of local memory.
Memory Search Increment Size	00010000	Multi-CPU feature used to offset the location of the Bug work page. This must be a multiple of the debugger work page, modulo $10000 (64KB). Typically, Memory Search Increment Size is the product of CPU number and size of the Bug work page. Example: first CPU $0 (0 x $10000), second CPU $10000 (1 x $10000), etc.
Memory Search Delay Enable [Y/N]	N	No delay before the Bug begins its search for a work page.

Table 3-3. ENV Command Parameters (Continued)

ENV Parameter and Options	Default	Meaning of Default
Memory Search Delay Address	FFFFD20F	Default address is $FFFFD20F. This is the MVME162P4 GCSR GPCSR0 as accessed through VMEbus A16 space; it assumes the MVME162P4 GRPAD (group address) and BDAD (board address within group) switches are set to "on". This byte-wide value is initialized to $FF by MVME162P4 hardware after a System or Power-On reset. In a multi-162P4 environment, where the work pages of several Bugs reside in the memory of the primary (first) MVME162P4, the non-primary CPUs will wait for the data at the Memory Search Delay Address to be set to $00, $01, or $02 (refer to the *Memory Requirements* section in Chapter 3 for the definition of these values) before attempting to locate their work page in the memory of the primary CPU.
Memory Size Enable [Y/N]	Y	Memory is sized for Self-Test diagnostics.
Memory Size Starting Address	00000000	Default Starting Address is $0.
Memory Size Ending Address	00100000	Default Ending Address is the calculated size of local memory.
Note Memory Configuration Defaults. The default configuration for Dynamic RAM mezzanine boards will position the mezzanine with the largest memory size to start at the address selected with the **ENV** parameter "Base Address of Dynamic Memory". The Base Address parameter defaults to 0. The smaller sized mezzanine will follow immediately above the larger in the memory map. If mezzanines of the same size and type are present, the first (closest to the board) is mapped to the selected base address. If mezzanines of the same size but different type (parity and ECC) are present, the parity type will be mapped to the selected base address and the ECC type mezzanine will follow. The SRAM does not default to a location in the memory map that is contiguous with Dynamic RAM.		
Base Address of Dynamic Memory	00000000	Beginning address of Dynamic Memory (Parity and/or ECC type memory). Must be a multiple of the Dynamic Memory board size, starting with 0. Default is $0.

Table 3-3. ENV Command Parameters (Continued)

<table>
<tr><th>ENV Parameter and Options</th><th>Default</th><th>Meaning of Default</th></tr>
<tr><td>Size of Parity Memory</td><td>00100000</td><td>The size of the Parity type dynamic RAM mezzanine, if any. The default is the calculated size of the Dynamic memory mezzanine board.</td></tr>
<tr><td>Size of ECC Memory Board 0</td><td>00000000</td><td>The size of the first ECC type memory mezzanine. The default is the calculated size of the memory mezzanine.</td></tr>
<tr><td>Size of ECC Memory Board 1</td><td>00000000</td><td>The size of the second ECC type memory mezzanine. The default is the calculated size of the memory mezzanine.</td></tr>
<tr><td>Base Address of Static Memory</td><td>FFE00000</td><td>The beginning address of SRAM. The default is FFE00000 for the onboard 128KB SRAM, or E1000000 for the 2MB SRAM mezzanine. If only 2MB SRAM is present, it defaults to address 00000000.</td></tr>
<tr><td>Size of Static Memory</td><td>00080000</td><td>The size of the SRAM type memory present. The default is the calculated size of the onboard SRAM or an SRAM type mezzanine.</td></tr>
<tr><td colspan="3"><b>ENV</b> asks the following series of questions to set up the VMEbus interface for the MVME162 series modules. You should have a working knowledge of the VMEchip2 as given in the <i>MVME1x2P4 VME Embedded Controller Programmer’s Reference Guide</i> in order to perform this configuration. Also included in this series are questions for setting ROM and Flash access time.
The slave address decoders are used to allow another VMEbus master to access a local resource of the MVME162P4. There are two slave address decoders set. They are set up as follows:</td></tr>
<tr><td>Slave Enable #1 [Y/N]</td><td>Y</td><td>Yes, set up and enable Slave Address Decoder #1.</td></tr>
<tr><td>Slave Starting Address #1</td><td>00000000</td><td>Base address of the local resource that is accessible by the VMEbus. Default is the base of local memory, $0.</td></tr>
<tr><td>Slave Ending Address #1</td><td>000FFFFF</td><td>Ending address of the local resource that is accessible by the VMEbus. Default is the end of calculated memory.</td></tr>
</table>

Table 3-3. ENV Command Parameters (Continued)

ENV Parameter and Options	Default	Meaning of Default
Slave Address Translation Address #1	00000000	This register allows the VMEbus address and the local address to differ. The value in this register is the base address of the local resource that is associated with the starting and ending address selection from the previous questions. Default is 0.
Slave Address Translation Select #1	00000000	This register defines which bits of the address are significant. A logical "1" indicates significant address bits, logical "0" is non-significant. Default is 0.
Slave Control #1	03FF	Defines the access restriction for the address space defined with this slave address decoder. Default is $03FF.
Slave Enable #2 [Y/N]	N	Do not set up and enable Slave Address Decoder #2.
Slave Starting Address #2	00000000	Base address of the local resource that is accessible by the VMEbus. Default is 0.
Slave Ending Address #2	00000000	Ending address of the local resource that is accessible by the VMEbus. Default is 0.
Slave Address Translation Address #2	00000000	Works the same as Slave Address Translation Address #1. Default is 0.
Slave Address Translation Select #2	00000000	Works the same as Slave Address Translation Select #1. Default is 0.
Slave Control #2	0000	Defines the access restriction for the address space defined with this slave address decoder. Default is $0000.
Master Enable #1 [Y/N]	Y	Yes, set up and enable Master Address Decoder #1.
Master Starting Address #1	02000000	Base address of the VMEbus resource that is accessible from the local bus. Default is the end of calculated local memory (unless memory is less than 16MB; then this register is always set to 01000000).

Table 3-3. ENV Command Parameters (Continued)

ENV Parameter and Options	Default	Meaning of Default
Master Ending Address #1	EFFFFFFF	Ending address of the VMEbus resource that is accessible from the local bus. Default is the end of calculated memory.
Master Control #1	0D	Defines the access characteristics for the address space defined with this master address decoder. Default is $0D.
Master Enable #2 [Y/N]	N	Do not set up and enable Master Address Decoder #2.
Master Starting Address #2	00000000	Base address of the VMEbus resource that is accessible from the local bus. Default is $00000000.
Master Ending Address #2	00000000	Ending address of the VMEbus resource that is accessible from the local bus. Default is $00000000.
Master Control #2	00	Defines the access characteristics for the address space defined with this master address decoder. Default is $00.
Master Enable #3 [Y/N]	Depends on calculated size of local RAM	Yes, set up and enable Master Address Decoder #3. This is the default if the board contains less than 16MB of calculated RAM. Do not set up and enable the Master Address Decoder #3. This is the default for boards containing at least 16MB of calculated RAM.
Master Starting Address #3	00000000	Base address of the VMEbus resource that is accessible from the local bus. If enabled, the value is calculated as one more than the calculated size of memory. If not enabled, the default is $00000000.
Master Ending Address #3	00000000	Ending address of the VMEbus resource that is accessible from the local bus. If enabled, the default is $00FFFFFF, otherwise $00000000.

Table 3-3. ENV Command Parameters (Continued)

ENV Parameter and Options	Default	Meaning of Default
Master Control #3	00	Defines the access characteristics for the address space defined with this master address decoder. If enabled, the default is $3D, otherwise $00.
Master Enable #4 [Y/N]	N	Do not set up and enable Master Address Decoder #4.
Master Starting Address #4	00000000	Base address of the VMEbus resource that is accessible from the local bus. Default is $0.
Master Ending Address #4	00000000	Ending address of the VMEbus resource that is accessible from the local bus. Default is $0.
Master Address Translation Address #4	00000000	This register allows the VMEbus address and the local address to differ. The value in this register is the base address of the VMEbus resource that is associated with the starting and ending address selection from the previous questions. Default is 0.
Master Address Translation Select #4	00000000	This register defines which bits of the address are significant. A logical "1" indicates significant address bits, logical "0" is non-significant. Default is 0.
Master Control #4	00	Defines the access characteristics for the address space defined with this master address decoder. Default is $00.
Short I/O (VMEbus A16) Enable [Y/N]	Y	Yes, Enable the Short I/O Address Decoder.
Short I/O (VMEbus A16) Control	01	Defines the access characteristics for the address space defined with the Short I/O address decoder. Default is $01.
F-Page (VMEbus A24) Enable [Y/N]	Y	Yes, Enable the F-Page Address Decoder.
F-Page (VMEbus A24) Control	02	Defines the access characteristics for the address space defined with the F-Page address decoder. Default is $02.

Table 3-3. ENV Command Parameters (Continued)

ENV Parameter and Options	Default	Meaning of Default
ROM Access Time Code	04	Defines the ROM access time. The default is $04, which sets an access time of five clock cycles of the local bus.
Flash Access Time Code	03	Defines the Flash access time. The default is $03, which sets an access time of four clock cycles of the local bus.
MCC Vector Base VMEC2 Vector Base #1 VMEC2 Vector Base #2	05 06 07	Base interrupt vector for the component specified. Default: MC2chip = $05, VMEchip2 Vector 1 = $06, VMEchip2 Vector 2 = $07.
VMEC2 GCSR Group Base Address	D2	Specifies group address ($FFFF*XX*00) in Short I/O for this board. Default = $D2.
VMEC2 GCSR Board Base Address	00	Specifies base address ($FFFFD2*XX*) in Short I/O for this board. Default = $00.
VMEbus Global Time Out Code	01	Controls VMEbus timeout when the MVME162P4 is system controller. Default $01 = 64 µs.
Local Bus Time Out Code	02	Controls local bus timeout. Default $02 = 256 µs.
VMEbus Access Time Out Code	02	Controls the local-bus-to-VMEbus access timeout. Default $02 = 32 ms.

Configuring the IndustryPacks

ENV asks the following series of questions to set up IndustryPack modules (IPs) on MVME162P4s.

The *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide* describes the base addresses and the IP register settings. Refer to that manual for information on setting base addresses and register bits.

```
IP A Base Address= 00000000?
IP B Base Address= 00000000?
IP C Base Address= 00000000?
IP D Base Address= 00000000?
```

Base address for mapping IP modules. Only the upper 16 bits are significant.

```
IP D/C/B/A Memory Size          = 00000000?
```

Define the memory size requirements for the IP modules:

Bits	IP	Register Address
31-24	D	FFFBC00F
23-16	C	FFFBC00E
15-08	B	FFFBC00D
07-00	A	FFFBC00C

```
IP D/C/B/A General Control      = 00000000?
```

Define the general control requirements for the IP modules:

Bits	IP	Register Address
31-24	D	FFFBC01B
23-16	C	FFFBC01A
15-08	B	FFFBC019
07-00	A	FFFBC018

```
IP D/C/B/A Interrupt 0 Control = 00000000?
```

Define the interrupt control requirements for the IP modules, channel 0:

Bits	IP	Register Address
31-24	D	FFFBC016
23-16	C	FFFBC014
15-08	B	FFFBC012
07-00	A	FFFBC010

```
IP D/C/B/A Interrupt 1 Control = 00000000?
```

Define the interrupt control requirements for the IP modules, channel 1:

Bits	IP	Register Address
31-24	D	FFFBC017
23-16	C	FFFBC015
15-08	B	FFFBC013
07-00	A	FFFBC011

Caution

If you have specified environmental parameters that will cause an overlap condition, a warning message will appear before the environmental parameters are saved in NVRAM. The important information about each configurable element in the memory map is displayed, showing where any overlap conditions exist. This allows you to quickly identify and correct an undesirable configuration before it is saved.

If an undesirable configuration already exists, you may wish to restore the factory defaults with **env;d** <CR>.

ENV warning example:

```
WARNING: Memory MAP Overlap Condition Exists

S-Address  E-Address  Enable   Overlap  M-Type  Memory-MAP-Name

$00000000  $FFFFFFFF  Yes      Yes      Master  Local Memory (DRAM)

$FFE00000  $FFE7FFFF  Yes      Yes      Master  Static RAM

$01000000  $EFFFFFFF  Yes      Yes      Master  VMEbus Master #1

$00000000  $00000000  No       No       Master  VMEbus Master #2

$00000000  $00FFFFFF  Yes      Yes      Master  VMEbus Master #3

$00000000  $00000000  No       No       Master  VMEbus Master #4

$F0000000  $FF7FFFFF  Yes      Yes      Master  VMEbus F Pages (A24/A32)

$FFFF0000  $FFFFFFFF  Yes      Yes      Master  VMEbus Short I/O (A16)

$FF800000  $FFBFFFFF  Yes      Yes      Master  Flash/PROM

$FFF00000  $FFFEFFFF  Yes      Yes      Master  Local I/O
```

```
$00000000  $00000000   No       No        Master   Industry Pack A
$00000000  $00000000   No       No        Master   Industry Pack B
$00000000  $00000000   No       No        Master   Industry Pack C
$00000000  $00000000   No       No        Master   Industry Pack D
$00000000  $00000000   No       No        Slave    VMEbus Slave #1
$00000000  $00000000   No       No        Slave    VMEbus Slave #2
```

Functional Description 4

Introduction

This chapter describes the MVME162P4 VME embedded controller on a block diagram level. The *Summary of Features* provides an overview of the MVME162P4, 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 MVME162P4 blocks, including programmable registers in the ASICs and peripheral chips, can be found in the *MVME1x2P4 Embedded Controller Programmer's Reference Guide* (part number V1x2PFXA/PG). Refer to that manual for a functional description of the MVME162P4 in greater depth.

Summary of Features

The following table summarizes the features of the MVME162P4 VME embedded controller.

Table 4-1. MVME162P4 Features

Feature	Description
Microprocessor	MVME162P4: 25MHz or 32MHZ MC68040 or MC68LC040 processor
Form factor	6U VMEbus
Memory	16/32MB synchronous DRAM (SDRAM), configurable to emulate 1/4/8/16MB parity-protected DRAM or 4/8/16/32MB ECC-protected DRAM
	512KB SRAM with battery backup
Flash memory	MVME162P4: One Intel 28F016SA 1MB or 2MB 8-bit Flash device
EPROM	One 32-pin JEDEC standard PLCC EPROM socket with 512Kb x8 density
Real-time clock	8KB NVRAM with RTC, battery backup, and watchdog function (SGS-Thomson M48T58)

Table 4-1. MVME162P4 Features (Continued)

Feature	Description
Switches	**RESET** and **ABORT** switches on front panel
Status LEDs	Eight: Board Fail (**FAIL**), CPU Status (**STAT**), CPU Activity (**RUN**), System Controller (**SCON**), LAN Activity (**LAN**), Fuse Status (**FUSE**), SCSI Activity (**SCSI**), VME Activity (**VME**)
Timers	Four 32-bit tick timers and watchdog timer in Petra ASIC
	Two 32-bit tick timers and watchdog timer in VMEchip2 ASIC
Interrupts	Eight software interrupts (on versions with VMEchip2 ASIC)
VME I/O	VMEbus P2 connector
Serial I/O	Two EIA-232-D, EIA-530, EIA-422, or EIA-425 configurable serial ports via front panel and transition module
Ethernet I/O	Optional Ethernet transceiver interface via DB15 connector on transition module
IP interface	Four IndustryPack interface channels via 3M connectors behind front panel
SCSI I/O	Optional SCSI interface with DMA via P2 or LCP2 adapter board
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)

Processor and Memory

The MVME162P4 is based on the MC68040/MC68LC040 microprocessor. The boards are built with 16MB or 32MB shared DRAM (SDRAM). Various versions of the MVME162P4 may have the SDRAM configured to model 1MB, 4MB, 8MB, or 16MB of parity-protected DRAM or 4MB, 8MB, 16MB, or 32MB of ECC-protected DRAM.

All boards are available with 512KB of SRAM (with battery backup); time-of-day clock (with battery backup); an optional Ethernet transceiver interface; two serial ports with EIA-232-D or EIA-530 or EIA-485/-422 interface; six tick timers with watchdog timer(s); an EPROM socket; 1MB or 2MB Flash memory (one Flash device); four IndustryPack (IP) interfaces with DMA; optional SCSI bus interface with DMA; and an optional VMEbus interface (local bus to VMEbus/VMEbus to local bus, with A16/A24/A32, D8/D16/D32 bus widths and a VMEbus system controller).

I/O Implementation

Input/Output (I/O) signals on the MVME162P4 are routed to the VMEbus P2 connector. The main board is connected through a P2 adapter board and cables to the transition boards. The MVME162P4 supports the MVME712-12, MVME712-13, MVME712M, MVME712A, MVME712AM, and MVME712B series of transition boards (all referred to in this manual as MVME712*x*, unless separately specified).

The MVME712*x* transition boards provide configuration headers, serial port drivers, and industry-standard connectors for various I/O devices. Although the MVME712*x* series transition boards were originally designed to support MVME167 boards, they lend themselves readily to the MVME162P4 application as long as you keep a few special considerations in mind (refer to the section on the Serial Communications Interface, later in this chapter, for details).

The I/O connection for the serial ports on the MVME162P4 is also implemented with two DB25 connectors on the front panel. In addition, the panel has cutouts for routing of flat cables to the optional IndustryPack modules.

ASICs

The following ASICs are used on the MVME162P4:

- **VMEchip2 ASIC** (VMEbus interface). Provides two tick timers, a watchdog timer, programmable map decoders for the master and slave interfaces, and a VMEbus to/from local bus DMA controller as well as a VMEbus to/from local bus non-DMA programmed access interface, a VMEbus interrupter, a VMEbus system controller, a VMEbus interrupt handler, and a VMEbus requester.

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.

❑ **Petra ASIC**. Combines the functions previously covered by the MC2 chip, the MCECC chip, and the IP2 chip in a single ASIC.

  – **MC2 function**. Provides a parity DRAM emulation. Also supplies four tick timers and interfaces to the LAN chip, SCSI chip, serial port chip, BBRAM, EPROM/Flash, and SRAM.

  – **MCECC function**. Provides an ECC DRAM emulation.

  – **IP2 function**. Provides control and status information for up to four single-wide or two double-wide **IndustryPack modules** that can be plugged into the MVME162P4 main board.

Block Diagram

The block diagram in Figure 4-1 on page 4-5 illustrates the MVME162P4's overall architecture.

Functional Description

This section contains a functional description of the major blocks on the MVME162P4.

Data Bus Structure

The local bus on the MVME162P4 is a 32-bit synchronous bus that is based on the MC68040 bus, and which supports burst transfers and snooping. The various local bus master and slave devices use the local bus to communicate. The local bus is arbitrated by priority type; the priority of the local bus masters from highest to lowest is: 82596CA LAN, 53C710 SCSI, VMEbus, and MPU. As a general rule, any master can access any

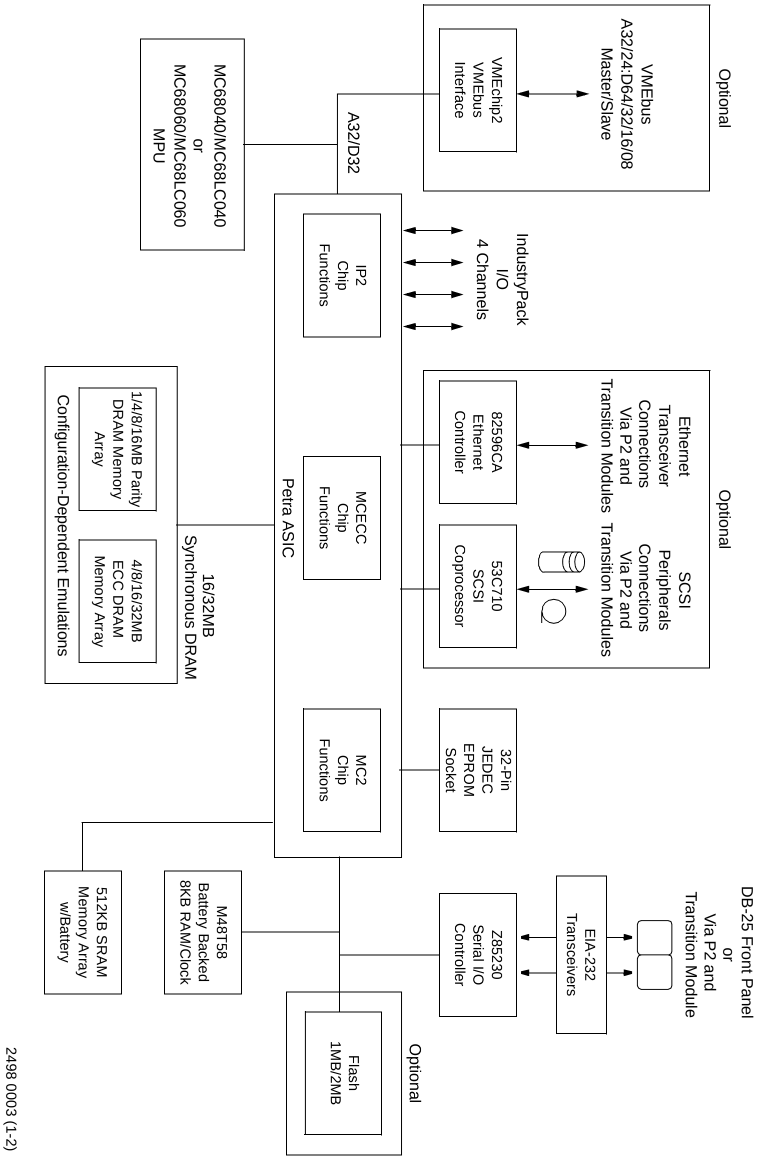


Figure 4-1. MVME162P4 Block Diagram

slave; not all combinations pass the common sense test, however. Refer to the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide* and to the user's guide for each device to determine its port size, data bus connection, and any restrictions that apply when accessing the device.

Microprocessor

MVME162P4 models may be ordered with an MC68040 or MC68LC040 microprocessor.

The MC68040 has on-chip instruction and data caches and a floating-point processor. (A floating-point *co*processor is the major difference between the MC68040 and MC68LC040.) Refer to the MC68040 user's manual for more information.

MC68*xx*040 Cache

The MVME162P4 local bus masters (VMEchip2, processor, 53C710 SCSI controller, and 82596CA Ethernet controller) have programmable control of the snoop/caching mode. The IP DMA local bus master's snoop control function is governed by the settings of switch S5 segments 1 and 2 (refer to *IP DMA Snoop Control (S5 Pins 1/2)* on page 1-17). S5 determines the value of the snoop control signal for all IP DMA transfers. This includes the IP DMA which executes when the DMA control registers are updated while the IP DMA is operating in command chaining mode.

The MVME162P4 local bus slaves that support the snoop/caching mode are defined in the "Local Bus Memory Map" in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide*.

Note As outlined in Table 1-5, the snoop capabilities of the MC68xx040 processor differ from those of the MC68xx060 used on MVME172P4 series boards. Application software must take these differences into account.

No-VMEbus-Interface Option

In support of possible future configurations in which the MVME162P4 might be offered as an embedded controller without the VMEbus interface, certain logic in the VMEchip2 has been duplicated in the Petra chip. (For the location of the overlapping logic, refer to Chapter 1 in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide*.) As long as the VMEchip2 ASIC is present, the redundant logic is inhibited in the Petra chip. The enabling signals for these functions are controlled by software and Petra chip hardware initialization.

Memory Options

The following memory options are available on the different versions of MVME162P4 boards.

DRAM

MVME162P4 boards are built with 16MB or 32MB shared DRAM (SDRAM). Depending on build options chosen at the time of manufacture, various versions of the MVME162P4 have the SDRAM configured to model 1MB, 4MB, 8MB, or 16MB of parity-protected DRAM or 4MB, 8MB, 16MB, or 32MB of ECC-protected DRAM.

The SDRAM memory array itself is always a single-bit error correcting and multi-bit error detection memory, irrespective of which interface model you use to access the SDRAM. When the MC2 (parity) memory controller interface is used to access the SDRAM, single-bit errors are undetectable to users and multi-bit errors are defined to be parity errors.

Firmware will initialize the memory controller to maintain backward compatibility with MVME162FX or -LX products. If the Petra ASIC is supporting MVME162FX functionality, the parity memory controller model will be enabled by default. If the Petra ASIC is supporting MVME162LX functionality, firmware will enable either the parity or the ECC memory controller model, depending on board configuration. (The board configuration is a function of switch settings and resistor population options.)

User code can modify Petra register settings to operate in either mode. User code can also modify map decoder/switch settings to enable the maximum amount of memory available. The minimum SDRAM configuration is 16MB.

For specifics on SDRAM performance and for detailed programming information, refer to the chapters on MC2 and MCECC memory controller emulations in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide*.

SRAM

The MVME162P4 implementation includes a 512KB SRAM (static RAM) option. SRAM architecture is single non-interleaved. SRAM performance is described in the section on the SRAM memory interface in the chapter on the MC2 memory controller emulation in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide,* listed in Appendix E, *Related Documentation*. An onboard battery supplies VCC to the SRAM when main power is removed. The worst-case elapsed time for battery protection is 200 days. The SRAM arrays are not parity protected.

The battery backup function for the onboard SRAM is provided by is a coin-type Panasonic CR2032 device (or equivalent) that supports primary and secondary power sources. In the event of a main board power failure, the EM1275 checks power sources and switches to the source with the higher voltage.

If the voltage of the backup source is lower than two volts, the EM1275 blocks the second memory cycle; this allows software to provide an early warning to avoid data loss. Because the second access may be blocked during a power failure, software should do at least two accesses before relying on the data.

The MVME162P4 provides jumpers (on J22) that allow either power source of the backup battery to be connected to the VMEbus +5V STDBY pin or to one cell of the onboard battery. For example, the primary system backup source may be a battery connected to the VMEbus +5V STDBY pin and the secondary source may be the onboard battery. If the system source should fail or the board is removed from the chassis, the onboard battery takes over.

Caution

For proper SRAM operation, some jumper combination must be installed on the Backup Power Source Select header (refer to the jumper information in Chapter 1). If one of the jumpers is set to select the battery, a battery must be present on the MVME162P4. The SRAM may malfunction if inputs to the EM1275 are left unconnected.

The SRAM is controlled by the Petra MC2 sector, and the access time is programmable. Refer to the description of the Petra MC2 emulation in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide,* listed in Appendix E, *Related Documentation*, for more detail.

About the Battery

The power source for the onboard SRAM is a coin-type Panasonic CR2032 device (or equivalent) with two lithium cells. The battery is socketed for easy removal and replacement. Small capacitors are provided so that the battery can be quickly replaced without data loss.

The service life of the batteries is very dependent on the ambient temperature of the board and the power-on duty cycle. The lithium battery supplied on the MVME162P4 should provide at least two years of backup time with the board powered off and with an ambient temperature of 40° C. If the power-on duty cycle is 50% (the board is powered on half of the time), the battery lifetime is four years. At lower ambient temperatures, the backup time is correspondingly longer.

If you intend to place the board in storage, putting the M48T58 in power-save mode by stopping the oscillator will prolong battery life. This is especially important at high ambient temperatures. To enter power-saving mode, execute the 162Bug **PS** command (refer to Chapter 3, *162Bug Firmware*) or its equivalent application-specific command. When restoring the board to service, execute the 162Bug **SET** command (**set *mmddyyhhmm***) after installation to restart the oscillator and initialize the clock.

The MVME162P4 is shipped with the battery disconnected (that is, with VMEbus +5V standby voltage selected as both primary and secondary power source). In order to use the battery as a power source, whether primary or secondary, it is necessary to reconfigure the jumpers on J22 before installing the board. Refer to *SRAM Backup Power Source (J22)* on page 1-12 for available jumper configurations.

The power leads from the battery are exposed on the solder side of the board. The board should not be placed on a conductive surface or stored in a conductive bag unless the battery is removed.

Warning

Lithium batteries incorporate inflammable materials such as lithium and organic solvents. If lithium batteries are mistreated or handled incorrectly, they may burst open and ignite, possibly resulting in injury and/or fire. When dealing with lithium batteries, carefully follow the precautions listed below in order to prevent accidents.

- Do not short circuit.
- Do not disassemble, deform, or apply excessive pressure.
- Do not heat or incinerate.
- Do not apply solder directly.
- Do not use different models, or new and old batteries together.
- Do not charge.
- Always check proper polarity.

To remove the battery from the board, carefully pry the battery from its socket.

Before installing a new battery, ensure that the battery pins are clean. Note the battery polarity and press the battery into the socket. When the battery is in the socket, no soldering is required.

EPROM and Flash Memory

The MVME162P4 implementation includes 1MB or 2MB Flash memory. Flash memory is a single Intel device (28F016SA on the MVME162P4) organized in a 1MB x 8 or 2Mb x 8 configuration. For information on programming Flash, refer to the Intel documents listed in Appendix E, *Related Documentation*.

The Flash write enable signal is controlled by:

- ❑ A bit in the Flash Access Time Control register in the Petra ASIC
- ❑ A board-level configuration jumper (J24) and configuration switch (S5, segment 4) which determine the status of Flash write protection on the board

Refer to the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide*, listed in Appendix E, *Related Documentation*, for specifics.

The EPROM location is a standard JEDEC 32-pin PLCC capable of 4-Mbit densities organized as a 512 Kb x 8 device. The setting of a configuration switch (line GPI3, segment 5 on S4), allows reset code to be fetched either from Flash memory (S4 segment 5 set to **OFF**) or from EPROMs (S4 segment 5 set to **ON**).

Note that MVME162P4 models ordered without the VMEbus interface are shipped with Flash memory blank (the factory uses the VMEbus to program the Flash memory with debugger code). To use the debugger firmware, be sure that configuration switch S4 is set for the EPROM memory map. Refer to chapters 1 and 3 for further details.

Battery-Backed-Up RAM and Clock

An M48T58 RAM and clock chip is used on the MVME162P4. This chip provides a time-of-day clock, oscillator, crystal, power fail detection, memory write protection, 8KB of RAM, and a battery in one 28-pin package. The clock provides 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. No interrupts are generated by the clock. Although the M48T58 is an 8-bit device, the interface provided by the Petra chip supports 8-, 16-, and 32-bit accesses to the M48T58. Refer to the description of the Petra MC2 function in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide* and to the M48T58 data sheet for detailed programming guidance and battery life information.

VMEbus Interface and VMEchip2

The VMEchip2 ASIC provides the local-bus-to-VMEbus interface, the VMEbus-to-local-bus interface, and the DMA controller functions of the local VMEbus. The VMEchip2 also provides the VMEbus system controller functions. Refer to the VMEchip2 description in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide, listed in* Appendix E, *Related Documentation*, for detailed programming information.

Note that the Abort switch logic in the VMEchip2 is not used. The GPI inputs to the VMEchip2 which are located at $FFF40088 bits 7-0 are not used. Instead, the Abort switch interrupt is integrated into the Petra MC2 sector at location $FFF42043. The GPI inputs are integrated into the Petra MC2 sector at location $FFF4202C, bits 23-16.

I/O Interfaces

The MVME162P4 provides onboard I/O for many system applications. The I/O functions include serial ports, IndustryPack (IP) interfaces, and optional interfaces for LAN Ethernet transceivers and SCSI mass storage devices.

Serial Communications Interface

The MVME162P4 uses a Zilog Z85230 serial port controller to implement the two serial communications interfaces. Each interface supports CTS, DCD, RTS, and DTR control signals, as well as the TXD and RXD transmit/receive data signals and TXC/RXC synchronous clock signals.

The Z85230 sypports synchronous (SDLC/HDLC) and asynchronous protocols. The MVME162P4 hardware supports asynchronous serial baud rates of 110b/s to 38.4Kb/s.

The Z85230 supplies an interrupt vector during interrupt acknowledge cycles. The vector is modified based upon the interrupt source within the Z85230. Interrupt request levels are programmed via the Petra MC2 function (the MC2 emulation can handle up to four Z85230 chips). Refer to the Z85230 data sheet and to the MC2 programming model in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide* for information. Both are listed in Appendix E, *Related Documentation*

MVME162P4 Serial Port 1

Port A on the Z85230 is interfaced as DCE (data circuit-terminating equipment) with the EIA-232-D interface and is routed to:

1. The DB25 connector marked **SERIAL PORT 1/CONSOLE** on the front panel of the MVME162P4. **SERIAL PORT 1/CONSOLE** is an EIA-232-D DCE port.

Note This port can be connected to the TX and RX clocks which may be present on the DB25 connector. These connections are made via jumper header J16 on the MVME162P4 board. (The TxC and RxC clock lines are not available on the MVME712*x* transition modules.)

2. One of the following output connectors on the MVME712*x* transition module:

   MVME712M: The DB25 connector marked **SERIAL PORT 2** on the front panel. **SERIAL PORT 2** can be configured as an EIA-232-D DTE or DCE port, via jumper headers J16 and J17.

   MVME712A **or** *MVME712-12*: The DB9 connector marked **SERIAL PORT 2** on the front panel. **SERIAL PORT 2** is hardwired as an EIA-232-D DTE port.

   MVME712AM **or** *MVME712-13*: The DB9 connector marked **SERIAL PORT 2** ***OR*** the RJ11 jack on the front panel. **SERIAL PORT 2** is hardwired as EIA-232-D DTE; the RJ11 jack utilizes the built-in modem. Setting the jumper headers J26 and J27 on the MVME712AM/-13 configures the output as EIA-232-D DTE at **SERIAL PORT 2** or as a modem at the RJ11 jack.

Figure 1-3 (sheets 1 and 2) in Chapter 1 illustrates the two configurations available for Port A when the MVME162P4 is used with an MVME712M. Figure 1-5 (sheets 1 and 2) shows the two configurations available for Port A when the MVME162P4 is used with an MVME712A/AM/-12/-13.

MVME162P4 Serial Port 2

The configuration of port B on the Z85230 is determined via a Serial Interface Module (SIM) which is installed at connector J15 on the MVME162P4 board. Five SIMs are available:

SIMM05 — DTE with EIA-232-D interface
SIMM06 — DCE with EIA-232-D interface
SIMM07 — DTE with EIA-530 interface
SIMM08 — DCE with EIA-530 interface
SIMM09 — EIA-485 interface, or DCE or DTE with EIA-422 interface

Port B is routed, via the SIM, to:

1. The DB25 connector marked **SERIAL PORT 2** on the front panel of the MVME162P4. **SERIAL PORT 2** will be an EIA-232-D DCE or DTE port, or an EIA-530 DCE or DTE port, or an EIA-485 port, or an EIA-422 DCE or DTE port, depending upon which SIM is installed.

Note Port B is factory-configured for asynchronous communication. For synchronous communication, this port can be connected to the TX and RX clocks which may be present on the DB25 connector. These connections are made via jumper header J17 on the MVME162P4 board. (The TxC and RxC clock lines are available at the MVME712M **SERIAL PORT 4** port via header J15, but are not available on the other MVME712*x* transition modules.)

2. One of the following output connectors on the MVME712*x* transition module:

   MVME712M: The DB25 connector marked **SERIAL PORT 4** on the front panel. **SERIAL PORT 4** can be configured as an EIA-232-D

DTE or DCE port, via the jumper headers J18 and J19 on the MVME712M.

MVME712A, AM, -12, **or** *-13*: The DB9 connector marked **SERIAL PORT 4** on the front panel. **SERIAL PORT 4** is hard-wired as an EIA-232-D DTE port.

Figure 1-3 (sheets 3 through 6) in Chapter 1 illustrates the four configurations available for Port B when the MVME162P4 is used with an MVME712M. Note that the port configurations shown in Figure 1-3 sheets 5 and 6 are not recommended for synchronous applications because of the incorrect clock direction. Figure 1-4 (sheets 1 and 2) shows an MVME162P4 with the two configurations available with EIA-530 SIMs. Figure 1-5 (sheets 3 and 4) shows the two configurations available for Port B when the MVME162P4 is used with an MVME712A/AM/-12/-13. Figure 1-6 shows an MVME162P4 with the configuration available with the EIA-485 /EIA-422 SIM.

Caution

Do not connect serial data devices to the equivalent ports on the MVME712*x* transition module and the MVME162P4 front panel at the same time. This could result in simultaneous transmission of conflicting data.

Caution

Do not connect peripheral devices to Port 1, Port 3, or the Centronics printer port on the MVME712x transition module. In the EIA-232-D case, none of these ports are connected to any MVME162P4 circuits. In the EIA-530 case, attempting to use these ports would produce certain connections with the potential to damage the MVME162P4 or the peripherals.

Caution

When using an EIA-530 SIM or an EIA-485/EIA-422 SIM, do not connect the MVME162P4 to an MVME712*x* board. Neither the P2 adapter nor the transition boards support the EIA-530, EIA-485, and EIA-422 signals.

IndustryPack (IP) Interfaces

The IP2 function in the Petra ASIC on the MVME162P4 supports four IndustryPack (IP) interfaces: these are accessible from the front panel. The IP2 function also includes four DMA channels (one for each IP, or two for each double-wide IP), 32 or 30MHz (32 MHz for MC68LC0*x*0 or 30 MHz for MC680*x*0) or 8MHz IndustryPack clock selection (jumper selectable), and one programmable timebase strobe which is connected to the four interfaces. Refer to the IP2 programming model in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide,* listed in Appendix E, *Related Documentation*, for details of the IP interface. Refer to Chapter 5, *Pin Assignments* for the pin assignments of the IP connectors.

Notes MVME162P4 boards do *not* monitor power supply +5 Vdc power and assert IP reset if the power falls too low. Instead, IP reset is handled by the162Bug firmware's **ENV** command, as described in Chapter 3. The IP reset is also driven active by the power-up reset signal.

Two IP modules plugged into the same MVME162P4 board can *not* use the Strobe* signal unless the jumper is removed from J19. This will disconnect the Strobe* output from the Petra/IP2 ASIC.

Ethernet Interface

The MVME162P4 uses the Intel 82596CA LAN coprocessor to implement the optional Ethernet transceiver interface. The 82596CA accesses local RAM using DMA operations to perform its normal functions. Because the 82596CA has small internal buffers and the VMEbus has an undefined latency period, buffer overrun may occur if the DMA is programmed to access the VMEbus. Therefore, the 82596CA should not be programmed to access the VMEbus.

Every MVME162P4 that is built with an Ethernet interface is assigned an Ethernet Station Address. The address is $08003E2*xxxxx* where *xxxxx* is the unique 5-nibble number assigned to the board (that is, every MVME162P4 has a different value for *xxxxx*).

Each board has an Ethernet Station Address displayed on a label attached to the VMEbus P2 connector. In addition, the six bytes including the Ethernet address are stored in the BBRAM configuration area. That is, 08003E2*xxxxx* is stored in the BBRAM. The upper four bytes (08003E2*x*) are read at $FFFC1F2C; the lower two bytes (*xxxx*) are read at $FFFC1F30. The MVME162 debugger has the capability to retrieve or set the Ethernet address.

If the data in BBRAM is lost, use the number on the label on backplane connector P2 to restore it.

The Ethernet transceiver interface is located on the MVME162P4 main board, and the industry-standard DB15 connector is located on the MVME712*x* transition board.

Support functions for the 82596CA LAN coprocessor are provided by the Petra MC2 sector. Refer to the 82596CA user's guide and to the description of the MC2 function in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide*, listed in Appendix E, *Related Documentation*, for detailed programming information.

SCSI Interface

The MVME162P4 may have provision for mass storage subsystems through the industry-standard SCSI bus. These subsystems may include hard and floppy disk drives, streaming tape drives, and other mass storage devices. The optional SCSI interface is implemented using the NCR 53C710 SCSI I/O controller.

Support functions for the 53C710 are provided by the Petra MC2 sector. Refer to the NCR 53C710 user's guide and to the description of the MC2 function in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide* for detailed programming information.

SCSI Termination

It is important that the SCSI bus be properly terminated at both ends.

In the case of the MVME162P4, sockets are provided for terminators on the P2 or LCP2 adapter board. If the SCSI bus ends at the adapter board, termination resistors must be installed on the adapter board. +5V power to the SCSI bus TERM power line and termination resistors is supplied through a fuse located on the adapter board.

Local Resources

The MVME162P4 includes many resources for the local processor. These include tick timers, software-programmable hardware interrupts, a watchdog timer, and a local bus timeout.

Programmable Tick Timers

Six 32-bit programmable tick timers with 1μs resolution are available: two in the VMEchip2 ASIC and four in the Petra/MC2 chip. The tick timers may be programmed to generate periodic interrupts to the processor. Refer to the VMEchip2 and Petra/MC2 descriptions in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide* for detailed programming information.

Watchdog Timer

A watchdog timer function is provided in both the Petra/MC2 chip and the VMEchip2 ASIC. When the watchdog timer is enabled, it must be reset by software within the programmed time or it times out. The watchdog timer can be programmed to generate a SYSRESET signal, a local reset signal, or a board fail signal if it times out. Refer to the VMEchip2 and Petra/MC2 descriptions in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide*, listed in Appendix E, *Related Documentation*, for detailed programming information.

The watchdog timer logic is duplicated in the VMEchip2 and Petra/MC2 ASICs. Because the watchdog timer function in the VMEchip2 is a superset of that function in the Petra/MC2 chip (system reset function), the timer in the VMEchip2 is to be used in all cases except for versions of the MVME162P4 which do not include the VMEbus interface (that is, boards ordered with a “No VMEbus Interface” option).

Software-Programmable Hardware Interrupts

The VMEchip2 ASIC supplies eight software-programmable hardware interrupts. These interrupts allow software to create a hardware interrupt. Refer to the VMEchip2 description in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide* for detailed programming information.

Local Bus Timeout

The MVME162P4 provides timeout functions in the VMEchip2 ASIC and the Petra/MC2 chip for the local bus. When the timer is enabled and a local bus access times out, a Transfer Error Acknowledge (TEA) signal is sent to the local bus master. The timeout value is selectable by software for 8 µsec, 64 µsec, 256 µsec, or infinity. The local bus timer does not operate during VMEbus bound cycles. VMEbus bound cycles are timed by the VMEbus access timer and the VMEbus global timer. Refer to the VMEchip2 and Petra/MC2 descriptions in the *MVME1x2P4 VME Embedded Controller Programmer's Reference Guide,* (listed in Appendix E, *Related Documentation*) for detailed programming information.

The access timer logic is duplicated in the VMEchip2 and Petra/MC2 ASICs. Because the local bus timer in the VMEchip2 can detect an offboard access and the Petra/MC2 local bus timer cannot, the timer in the VMEchip2 ASIC is used in all cases except for versions of the MVME162P4 which do not include the VMEbus interface (that is, boards ordered with a "No VMEbus Interface" option).

Local Bus Arbiter

The local bus arbiter implements a fixed priority (see Table 4-2).

Table 4-2. Local Bus Arbitration Priority

Device	Priority	Note
LAN	0	Highest
Industry Pack DMA	1	
SCSI	2	...
VMEbus	3	Next Lowest
MC680*x*0/MC68LC0*x*0	4	Lowest

Connectors

The MVME162P4 has two 96-position DIN connectors: P1 and P2. P1 rows A, B, C, and P2 row B provide the VMEbus interconnection. P2 rows A and C provide the connection to the SCSI bus, serial ports, and Ethernet.

The serial ports on the MVME162P4 are also connected to two 25-pin DB-25 female connectors, J18 and J25, on the front panel. The four IP modules connect to the MVME162P4 by four pairs of 50-pin connectors. Four 50-pin connectors behind the front panel are for external connections to IP signals.

Pin assignments for the connectors on the MVME162P4 are listed in Chapter 5, *Pin Assignments*.

Remote Status and Control

The remote status and control connector, J6, is a 20-pin connector located behind the front panel of the MVME162P4. It provides system designers with flexibility in accessing critical indicator and reset functions. When the board is enclosed in a chassis and the front panel is not visible, this connector allows the Reset, Abort, and LED functions to be extended to the control panel of the system, where they are visible. Alternatively, this allows a system designer to construct a **RESET**/**ABORT/LED** panel that can be located remotely from the MVME162P4.

Pin Assignments 5

Connector Pin Assignments

This chapter summarizes the pin assignments for the following groups of interconnect signals on the MVME162P4:

Connector	Location	Table
IndustryPack A, B, C, D connectors	J4/5/8, J7/10/11, J20/21/27, J26/29/30	Table 5-1
Remote Reset connector	J6	Table 5-2
Serial port module connector	J15	Table 5-3
Serial Port 2 connector	J18	Table 5-4
Serial Port 1 connector	J25	Table 5-5
VMEbus connector P1	P1	Table 5-6
VMEbus connector P2	P2	Table 5-7

The tables in this chapter furnish pin assignments only. For detailed descriptions of the interconnect signals, consult the support information for the MVME162P4 (available through your Motorola sales office).

IndustryPack A, B, C, D Connectors

Up to four IndustryPack (IP) modules may be installed on the MVME162P4. For each IP module, there are two 50-pin plug connectors on the board:

- Module A, J4/5
- Module B, J10/11
- Module C, J20/21
- Module D, J29/30

For external cabling to the IP modules, four 50-pin IDC connectors (module A, J8; module B, J7; module C, J27; module D, J26) are provided behind the MVME162P4 front panel. The pin assignments are the same for both types of connector.

Table 5-1. IndustryPack Interconnect Signals

1	GND	CLK	2
3	RESET∗	IPD0	4
5	IPD1	IPD2	6
7	IPD3	IPD4	8
9	IPD5	IPD6	10
11	IPD7	IPD8	12
13	IPD9	IPD10	14
15	IPD11	IPD12	16
17	IPD13	IPD14	18
19	IPD15	BS∗	20
21	BS1∗	–12V	22
23	+12V	+5V	24
25	GND	GND	26
27	+5V	R/W∗	28
29	IDSEL∗	DMAREQ0∗	30
31	MEMSEL∗	DMAREQ1∗	32
33	INTSEL∗	DMACK∗	34
35	IOSEL∗	No Connection	36
37	IPA1	DMAEND∗	38
39	IPA2	ERROR∗	40
41	IPA3	INT_REQ0∗	42
43	IPA4	INT_REQ1∗	44
45	IPA5	STROBE∗	46
47	IPA6	ACK∗	48
49	No Connection	GND	50

Remote Reset Connector - J6

The MVME162P4 has a 20-pin connector (J6) mounted behind the front panel. When the MVME162P4 board is enclosed in a chassis and the front panel is not visible, this connector enables you to extend the reset, abort and LED functions to the control panel of the system, where they remain accessible.

Table 5-2. Remote Reset Connector J6 Pin Assignments

1	P5VF	LANLED*	2
3	P12VLED*	SCSILED*	4
5	VMELED*	No connection	6
7	RUNLED*	STSLED*	8
9	FAILSTAT*	No connection	10
11	SCONLED*	ABORTSW*	12
13	RESETSW*	GND	14
15	GND	GPI1	16
17	GPI2	GPI3	18
19	No connection	GND	20

Serial Port Module Connector - J15

Port B of the Z85230 serial communications controller on the MVME162P4 board is configurable via serial interface modules (SIMs) that are installed at connector J15. The serial interface modules currently available are listed and described in Chapter 2, *Startup and Operation*. The pin assignments for J15 are listed in the following table.

Table 5-3. Serial Port Module Connector Pin Assignments

1	–12V	GND	2
3	DBPIN1	DBPIN14	4
5	TXDB	TXCB	6
7	RXDB	LTXDB	8

Table 5-3. Serial Port Module Connector Pin Assignments

9	LTXCB	LRXDB	10
11	DBPIN16	DTRB	12
13	RXCB	CTSB	14
15	LRTSB	LRXCB	16
17	LCTSB	12VMODULE	18
19	DBPIN18	DSRB	20
21	DBPIN19	RTSB	22
23	DCDB	LDTRB	24
25	LDCDB	DBPIN21	26
27	DBPIN9	DBPIN22	28
29	DBPIN10	DBPIN23	30
31	DBPIN11	TXCOB	32
33	LTRXCB	DBPIN12	34
35	DBPIN25	DBPIN13	36
37	GND	GND	38
39	+5V	+5V	40

Serial Port 2 Connector - J18

A DB25 socket connector located on the front panel of the MVME162P4 provides the interface to Serial Port 2. The pin assignments for the connector are as follows.

Table 5-4. Serial Connector J18 Pin Assignments

1	DBPIN1	DBPIN14	14
2	TXDB	TXCB	15
3	RXDB	DBPIN16	16
4	RTSB	RXCB	17
5	CTSB	DBPIN18	18

Table 5-4. Serial Connector J18 Pin Assignments

6	DSRB	DBPIN19	19
7	GND	DTRB	20
8	DCDB	DBPIN21	21
9	DBPIN9	DBPIN22	22
10	DBPIN10	DBPIN23	23
11	DBPIN11	TXCOB	24
12	DBPIN12	DBPIN25	25
13	DBPIN13		

Serial Port 1/Console Connector - J25

A DB25 socket connector located on the front panel of the MVME162P4 provides the interface to Serial Port 1. The pin assignments for the connector are as follows.

Table 5-5. Serial Connector J25 Pin Assignments

1	No Connection	No Connection	14
2	TXDA	TXCA	15
3	RXDA	No Connection	16
4	RTSA	RXCA	17
5	CTSA	No Connection	18
6	DSRA	No Connection	19
7	GND	DTRA	20
8	DCDA	No Connection	21
9	No Connection	No Connection	22
10	No Connection	No Connection	23
11	No Connection	TXCOA	24
12	No Connection	No Connection	25
13	No Connection		

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 Row B supplies the base board with power, with the upper 8 VMEbus address lines, and with an additional 16 VMEbus data lines. P2 rows A and C are not used in the MVME162P4 implementation. The pin assignments for P1and P2 are listed in Table 5-6 and Table 5-7 respectively.

Table 5-6. 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

Table 5-6. VMEbus Connector P1 Pin Assignments (Continued)

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	P5VSTDBY	+12V	31
32	+5V	+5V	+5V	32

Table 5-7. VMEbus Connector P2 Pin Assignment

	Row A	Row B	Row C	
1	DB0*	+5V	C–	1
2	DB1*	GND	C+	2
3	DB2*	Not Used	T–	3
4	DB3*	VA24	T+	4
5	DB4*	VA25	R–	5
6	DB5*	VA26	R+	6
7	DB6*	VA27	P12VLAN	7
8	DB7*	VA28	No Connection	8
9	DBP*	VA29	No Connection	9
10	ATN*	VA30	No Connection	10
11	BSY*	VA31	No Connection	11
12	ACK*	GND	No Connection	12
13	RST*	+5V	No Connection	13

Table 5-7. VMEbus Connector P2 Pin Assignment (Continued)

14	MSG*	VD16	No Connection	14
15	SEL*	VD17	No Connection	15
16	DC*	VD18	No Connection	16
17	REQ*	VD19	No Connection	17
18	OI*	VD20	DBPIN14	18
19	DBPIN16	VD21	DBPIN18	19
20	DSRB	VD22	DBPIN19	20
21	DBPIN21	VD23	DBPIN9	21
22	DBPIN22	GND	DBPIN10	22
23	DBPIN23	VD24	DBPIN11	23
24	TXCOB	VD25	DBPIN12	24
25	TXDB	VD26	DBPIN25	25
26	RXDB	VD27	DBPIN13	26
27	RTSB	VD28	RXDA	27
28	RXCB	VD29	TXDA	28
29	CTSB	VD30	CTSA	29
30	DTRB	VD31	RTSA	30
31	DCDB	GND	DCDA	31
32	TXCB	+5V	DTRA	32

Specifications A

Board Specifications

The following table lists the general specifications for the MVME162P4 VME embedded controller. The subsequent sections detail cooling requirements and EMC regulatory compliance.

A complete functional description of the MVME162P4 boards appears in Chapter 4. Specifications for the optional IndustryPack modules can be found in the documentation for those modules.

Table A-1. MVME162P4 Specifications

<table>
<tr><th colspan="2">Characteristics</th><th>Specifications</th></tr>
<tr><td>Power requirements (with EPROM; without IPs)</td><td colspan="2">+5Vdc (±5%), 3.5A typical, 4.5A maximum
+12 Vdc (± 5%), 100 mA maximum
–12 Vdc (± 5%), 100 mA maximum</td></tr>
<tr><td>Operating temperature</td><td colspan="2">0°C to 70°C exit air with forced-air cooling (refer to *Cooling Requirements* and *Special Considerations for Elevated-Temperature Operation*)</td></tr>
<tr><td>Storage temperature</td><td colspan="2">–40°C to +85° C</td></tr>
<tr><td>Relative humidity</td><td colspan="2">5% to 90% (noncondensing)</td></tr>
<tr><td>Vibration (operating)</td><td colspan="2">2 Gs RMS, 20Hz-2000Hz random</td></tr>
<tr><td>Altitude (operating)</td><td colspan="2">5000 meters (16,405 feet)</td></tr>
<tr><td rowspan="4">Physical dimensions (base board only)</td><td>Height</td><td>Double-high VME board, 9.2 in. (233 mm)</td></tr>
<tr><td>Front panel width</td><td>0.8 in. (20 mm)</td></tr>
<tr><td>Front panel height</td><td>10.3 in. (262 mm)</td></tr>
<tr><td>Depth</td><td>6.3 in. (160 mm)</td></tr>
</table>

Cooling Requirements

The Motorola MVME162P4 VME Embedded Controller 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.

Special Considerations for Elevated-Temperature Operation

The following information is for users whose applications for the MVME162P4 may subject it to high temperatures.

The MVME162P4 uses commercial-grade devices. Therefore, it can operate in an environment with ambient air temperatures from 0° C to 70° C. Several factors influence the ambient temperature seen by components on the MVME162P4. Among them are inlet air temperature; air flow characteristics; number, types, and locations of IP modules; power dissipation of adjacent boards in the system, etc.

A temperature profile of the MVME162P4 (MVME162P-*xxx*) was developed in an MVME945 12-slot VME chassis. This board was loaded with one GreenSpring IP-Dual P/T module (position A) and three GreenSpring IP-488 module (positions B, C, and D). One 25W load board was installed adjacent to each side of the board under test. The exit air velocity was approximately 200 LFM between the MVME162P4 and the IP-Dual P/T module. Under these conditions, a 10° C rise between the inlet and exit air was observed. At 70° C exit air temperature (60° C inlet air), the junction temperatures of devices on the MVME162P4 were calculated (from the measured case temperatures) and did not exceed 100° C.

Caution

For elevated-temperature operation, perform similar measurements and calculations to determine the actual operating margin for your specific environment.

To facilitate elevated-temperature operation:

1. Position the MVME162P4 in the chassis to allow for maximum airflow over the component side of the board.

2. Do not place boards with high power dissipation next to the MVME162P4.

3. Use low-power IP modules only. The preferred locations for IP modules are position *A* (J4 and J5) and position *D* (J29 and J30).

EMC Regulatory Compliance

The MVME162P4 was tested *without* IndustryPacks 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 MVME162P4 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 MVME162P4 VME embedded controller, 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 MVME162P4 Boards

Condition	Possible Problem	Try This:
I. Nothing works, no display on the terminal.	A. If the RUN (or FUSE) 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 System Considerations, as described in this manual. 5. Review the Installation and Startup procedures, as described in 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. For VMEmodules, the processor module (controller) should be in the first (leftmost) slot. 2. Also check that the "system controller" function on the board is enabled, per this manual.
	C. The "system console" terminal may be configured incorrectly.	Configure the system console terminal as described in this manual.

Table B-1. Troubleshooting MVME162P4 Boards (Continued)

Condition	**Possible Problem**	**Try This:**
II. There is a display on the terminal, but input from the keyboard has no effect.	A. The keyboard may be connected incorrectly.	Recheck the keyboard connections and power.
	B. Board jumpers or switches may be configured incorrectly.	Verify the settings of the board jumpers and configuration switches as described in this manual.
	C. You may have invoked flow control by pressing a HOLD or PAUSE key, or by typing**:** **<CTRL>-S**	Press the HOLD or PAUSE key again. If this does not free up the keyboard, type in: **<CTRL>-Q**
III. Debug prompt `162-Bug>` does not appear at powerup, and the board does not autoboot.	A. Debugger EPROM/Flash may be missing.	1. Disconnect *all* power from your system. 2. Check that the proper debugger device is installed. 3. Set switch S4 segment 5 to ON. This enables use of the EPROM instead of the Flash memory. 4. Reconnect power. 5. Restart the system by “double-button reset”: press the RESET and ABORT switches at the same time; release RESET first, wait seven seconds, then release ABORT. 6. 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 `162-Bug>` appears at powerup, but the board does not autoboot.	A. The initial debugger environment parameters may be set incorrectly.	1. Start the onboard calendar clock and timer. Type: **set *mmddyyhhmm* <CR>** where the characters indicate the month, day, year, hour, and minute. The date and time will be displayed. ! Caution Performing the next step (**env;d**) will change some parameters that may affect your system’s operation.
	B. There may be some fault in the board hardware.	

Table B-1. Troubleshooting MVME162P4 Boards (Continued)

Condition	Possible Problem	Try This:
IV. *Continued*		2. At the command line prompt, type in: **env;d <CR>** This restores 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 the 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: `162-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 long as 10 minutes, depending on RAM size. They are complete when the prompt returns. (The onboard self-test is a valuable tool in isolating defects.) 8. The system may indicate that it has passed all the self-tests. Or, it may indicate a test that failed. If neither happens, enter: **de <CR>** Any errors should now be displayed. If there are any errors, go to step VI. If there are no errors, go to step V.

Table B-1. Troubleshooting MVME162P4 Boards (Continued)

Condition	Possible Problem	Try This:
V. The debugger is in system mode and the board autoboots, or the board has passed self-tests.	A. No apparent problems — troubleshooting is done.	No further troubleshooting steps are required.
VI. The board has failed one or more of the tests listed above, 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.		

Network Controller Data

C

Network Controller Modules Supported

The 162Bug firmware supports the following VMEbus network controller modules. The default address for each module type and position is shown to indicate where the controller must reside to be supported by 162Bug. The controllers are accessed via the specified CLUN and DLUNs listed here. The CLUN and DLUNs are used in conjunction with the debugger commands **NBH**, **NBO**, **NIOP**, **NIOC**, **NIOT**, **NPING**, and **NAB**; they are also used with the debugger system calls .NETRD, .NETWR, .NETFOPN, .NETFRD, .NETCFIG, and .NETCTRL.

Controller Type	CLUN	DLUN	Address	Interface Type
MVME162FX	$00	$00	$FFF46000	Ethernet
MVME376	$02	$00	$FFFF1200	Ethernet
MVME376	$03	$00	$FFFF1400	Ethernet
MVME376	$04	$00	$FFFF1600	Ethernet
MVME376	$05	$00	$FFFF5400	Ethernet
MVME376	$06	$00	$FFFF5600	Ethernet
MVME376	$07	$00	$FFFFA400	Ethernet
MVME374	$10	$00	$FF000000	Ethernet
MVME374	$11	$00	$FF100000	Ethernet
MVME374	$12	$00	$FF200000	Ethernet
MVME374	$13	$00	$FF300000	Ethernet
MVME374	$14	$00	$FF400000	Ethernet
MVME374	$15	$00	$FF500000	Ethernet

Disk/Tape Controller Data D

Controller Modules Supported

The following VMEbus disk/tape controller modules are supported by the 162Bug. The default address for each controller type is First Address. The controller can be addressed by First CLUN during execution of the **BH**, **BO**, or **IOP** commands, or during execution of the .DSKRD or .DSKWR TRAP #15 calls. Note that if another controller of the same type is used, the second one must have its address changed by its onboard jumpers and/or switches, so that it matches Second Address and can be called up by Second CLUN.

Controller Type	First CLUN	First Address	Second CLUN	Second Address
CISC Embedded Controller	$00 (Note 1)	--	--	--
MVME320 - Winchester/Floppy Controller	$11 (Note 2)	$FFFFB000	$12 (Note 2)	$FFFFAC00
MVME323 - ESDI Winchester Controller	$08	$FFFFA000	$09	$FFFFA200
MVME327A - SCSI Controller	$02	$FFFFA600	$03	$FFFFA700
MVME328 - SCSI Controller	$06	$FFFF9000	$07	$FFFF9800
MVME328 - SCSI Controller	$16	$FFFF4800	$17	$FFFF5800
MVME328 - SCSI Controller	$18	$FFFF7000	$19	$FFFF7800
MVME350 - Streaming Tape Controller	$04	$FFFF5000	$05	$FFFF5100

Notes:

1. If an MVME162P4 with an SCSI port is used, the MVME162P4 module has CLUN 0.

2. For MVME162P4s, the first MVME320 has CLUN $11; the second MVME320 has CLUN $12.

Default Configurations

Note SCSI Common Command Set (CCS) devices are the only ones tested by Motorola Computer Group.

CISC Embedded Controllers -- 7 Devices

Controller LUN	Address	Device LUN	Device Type
0	$*XXXXXXXX*	00 10 20 30 40 50 60	SCSI Common Command Set (CCS), which may be any of these: - Fixed direct access - Removable flexible direct access (TEAC style) - CD-ROM - Sequential access

MVME320 -- 4 Devices

Controller LUN	Address	Device LUN	Device Type
11	$FFFFB000	0	Winchester hard drive
		1	Winchester hard drive
12	$FFFFAC00	2	5-1/4" DS/DD 96 TPI floppy drive
		3	5-1/4" DS/DD 96 TPI floppy drive

MVME323 -- 4 Devices

Controller LUN	Address	Device LUN	Device Type
8	$FFFFA000	0	ESDI Winchester hard drive
		1	ESDI Winchester hard drive
9	$FFFFA200	2	ESDI Winchester hard drive
		3	ESDI Winchester hard drive

MVME327A -- 9 Devices

Controller LUN	Address	Device LUN	Device Type
2	$FFFFA600	00	SCSI Common Command Set
		10	(CCS), which may be any of these:
3	$FFFFA700	20	- Fixed direct access
		30	- Removable flexible direct access
		40	(TEAC style)
		50	- CD-ROM
		60	- Sequential access
		80	Local floppy drive
		81	Local floppy drive

MVME328 -- 14 Devices

Controller LUN	Address	Device LUN	Device Type
6	$FFFF9000	00	SCSI Common Command Set
		08	(CCS), which may be any of these:
		10	- Removable flexible direct access
7	$FFFF9800	18	(TEAC style)
		20	- CD-ROM
		28	- Sequential access
16	$FFFF4800	30	
		40	Same as above, but these
17	$FFFF5800	48	will only be available if
		50	the daughter card for the
		58	second SCSI channel is present.
18	$FFFF7000	60	
		68	
		70	
19	$FFFF7800		

MVME350 -- 1 Device

Controller LUN	Address	Device LUN	Device Type
4	$FFFF5000	0	QIC-02 streaming tape drive
5	$FFFF5100		

IOT Command Parameters

The following table lists the proper **IOT** command parameters for floppies used with boards such as the MVME328 and MVME162P4.

IOT Parameter	**Floppy Types and Formats**						
	DSDD 5	**PCXT 8**	**PCXT 9**	**PCXT9_3**	**PCAT**	**PS2**	**SH D**
Sector Size 0- 128 1- 256 2- 512 3-1024 4-2048 5-4096 =	1	2	2	2	2	2	2
Block Size: 0- 128 1- 256 2- 512 3-1024 4-2048 5-4096 =	1	1	1	1	1	1	1
Sectors/Track	10	8	9	9	F	12	24
Number of Heads =	2	2	2	2	2	2	2
Number of Cylinders =	50	28	28	50	50	50	50
Precomp. Cylinder =	50	28	28	50	50	50	50
Reduced Write Current Cylinder =	50	28	28	50	50	50	50
Step Rate Code =	0	0	0	0	0	0	0
Single/Double DATA Density =	D	D	D	D	D	D	D
Single/Double TRACK Density =	D	D	D	D	D	D	D
Single/Equal_in_all Track Zero Density =	S	E	E	E	E	E	E
Slow/Fast Data Rate =	S	S	S	S	F	F	F

IOT Parameter	Floppy Types and Formats						
	DSDD5	**PCXT8**	**PCXT9**	**PCXT9_3**	**PCAT**	**PS2**	**SHD**
Other Characteristics							
Number of Physical Sectors	0A00	0280	02D0	05A0	0960	0B40	1680
Number of Logical Blocks (100 in size)	09F8	0500	05A0	0B40	12C0	1680	2D00
Number of Bytes in Decimal	653312	327680	368460	737280	1228800	1474560	2949120
Media Size/Density	5.25/DD	5.25/DD	5.25/DD	3.5/DD	5.25/HD	3.5/HD	3.5/ED

Notes

1.All numerical parameters are in hexadecimal unless otherwise noted.

2.The DSDD5 type floppy is the default setting for the debugger.

Related Documentation E

MCG Documents

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

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

Table E-1. Motorola Computer Group Documents

Document Title	Motorola Publication Number
MVME1x2P4 VME Embedded Controller Programmer's Reference Guide	V1x2PFXA/PG
MVME162Bug Diagnostics User's Manual	V162DIAA/UM
Debugging Package for Motorola 68K CISC CPUs User's Manual (Parts 1 and 2)	68KBUG1/D 68KBUG2/D
Single Board Computers SCSI Software User's Manual	SBCSCSI/D
MVME712M Transition Module and P2 Adapter Board Installation and Use	VME712MA/IH
MVME712-12, MVME712-13, MVME712A, MVME712AM, and MVME712B Transition Modules and LCP2 Adapter Board User's Manual	MVME712A/D
SIMM09 Serial Interface Module Installation Guide	SIMM09A/IH

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

Manufacturers' Documents

For additional information, refer to the following table for manufacturers' data sheets or user's manuals. As an additional help, a source for the listed document is provided. Please note that, while these sources have been verified, the information is subject to change without notice.

Table E-2. Manufacturers' Documents

Document Title and Source	Publication Number
M68000 Family Reference Manual	M68000FR
MC68040 Microprocessor User's Manual	M68040UM
Literature Distribution Center for Motorola Telephone: 1-800- 441-2447 FAX: (602) 994-6430 or (303) 675-2150 E-mail: ldcformotorola@hibbertco.com Web: http://www.mot.com/SPS	

Table E-2. Manufacturers’ Documents (Continued)

Document Title and Source	Publication Number
82596CA Local Area Network Coprocessor Data Sheet 82596CA Local Area Network Coprocessor User’s Manual 28F016SA Flash Memory Data Sheet Intel Corporation Web: http://developer.intel.com/design	290218 296853 209435
SYM 53C710 (was NCR 53C710) SCSI I/O Processor Data Manual SYM 53C710 (was NCR 53C710) SCSI I/O Processor Programmer’s Guide Symbios Logic Inc. 1731 Technology Drive, Suite 600 San Jose, CA 95110 NCR Managed Services Center — Telephone: 1-800-262-7782 Web: http://www.symbios.com/techsupport	NCR53C710DM NCR53C710PG
M48T58(B) TIMEKEEPER™ and 8K x 8 Zeropower™ RAM Data Sheet SGS-Thomson Microelectronics Group Marketing Headquarters (or nearest Sales Office) 1000 East Bell Road Phoenix, Arizona 85022 Telephone: (602) 867-6100 Web: http://www.st.com/stonline/books	M48T58
Z85230 Serial Communications Controller Product Brief Zilog Inc. 210 Hacienda Avenue Campbell, CA 95008-6609 Web: http://www.zilog.com/products	Z85230pb.pdf

Related Specifications

For additional information, refer to the following table for related specifications. As an additional help, a source for the listed document is provided. Please note that, while these sources have been verified, the information is subject to change without notice.

Table E-3. Related Specifications

Document Title and Source	**Publication Number**
VME64 Specification VITA (VMEbus International Trade Association) 7825 E. Gelding Drive, Suite 104 Scottsdale, AZ 85260 Telephone: (602) 951-8866 Web: http://www.vita.com	ANSI/VITA 1-1994
NOTE: An earlier version of the VME specification is available as: Versatile Backplane Bus: VMEbus Institute of Electrical and Electronics Engineers, Inc. Publication and Sales Department 345 East 47th Street New York, New York 10017-21633 Telephone: 1-800-678-4333	ANSI/IEEE Standard 1014-1987

Table E-3. Related Specifications (Continued)

Document Title and Source	**Publication Number**
OR *Microprocessor system bus for 1 to 4 byte data* Bureau Central de la Commission Electrotechnique Internationale 3, rue de Varembé Geneva, Switzerland	 *IEC 821 BUS*
ANSI Small Computer System Interface-2 (SCSI-2), Draft Document X3.131-198X, Revision 10c Global Engineering Documents 15 Inverness Way East Englewood, CO 80112-5704	X3.131-198X Rev. 10c
IndustryPack Logic Interface Specification, Revision 1.0 VITA (VMEbus International Trade Association) 7825 E. Gelding Drive, Suite 104 Scottsdale, AZ 85260 Telephone: (602) 951-8866 Web: http://www.vita.com	ANSI/VITA 4-1995
Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange (EIA-232-D) Global Engineering Documents Suite 400 1991 M Street, NW Washington, DC 20036 Telephone: 1-800-854-7179 Telephone: (303) 397-7956 Web: http://global.ihs.com	ANSI/EIA-232-D Standard

Index

I N D E X