ARTISAN®
TECHNOLOGY GROUP

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

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

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

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

Custom engineering
so your equipment works exactly as you specify.

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

Expert team | Trust guarantee | 100% satisfaction

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

Find the *Emerson / Motorola MVME162PA-344SE* at our website: *Click HERE*

MVME162FX 400/500-Series VME Embedded Controller Installation and Use

V162FXA/IH4

Edition of March 2000

Notice

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

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 The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of Motorola, Inc.

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

Limited and Restricted Rights Legend

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

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

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

Preface

MVME162FX 400/500-Series 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 400/500-Series MVME162FX embedded controller. The board is referred to as the “MVME162FX” throughout this document. As of the publication date, the information presented in this manual applies to the following MVME162FX models:

With 01-W3960B*xxx* Series Base Board		
MVME162-403	MVME162-412	MVME162-432
	MVME162-413	MVME162-433
MVME162-410	MVME162-430	
MVME162-411	MVME162-431	MVME162-453
MVME162-510A	MVME162-520A	MVME162-530A
MVME162-511A	MVME162-521A	MVME162-531A
MVME162-512A	MVME162-522A	MVME162-532A
MVME162-513A	MVME162-523A	MVME162-533A
With 01-W3182F*xxx* Series Base Board		
MVME162-510B		MVME162-510C
MVME162-512B		MVME162-512C

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

Conventions Used in This Manual

The following typographical conventions are used in this document:

bold

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

italic

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

`courier`

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

<Enter>, <Return> or <CR>

Represents the carriage return or Enter key.

CTRL

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

|

Separates two or more items that you can select from (one only).

[ ]

Encloses an optional item that may occur zero or one time.

{ }

Encloses an optional item that may occur zero or more times.

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.

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

MC68040™ is a trademark of Motorola, Inc.

IndustryPack™ and IP™ are trademarks of GreenSpring Computers, Inc.

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

Recent Updates

This edition of the MVME162FX VME Embedded Controller Installation and Use document incorporates the following changes:

Date	Description of Change
March 2000	A chapter of pin assignments was added to the manual. The frontmatter and the Related Publications sections were updated, and the overall manual contents reorganized, to reflect current MCG practice in those areas. Details of console cable routing to the base board and transition modules were clarified in response to user feedback.

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 violates safety standards of design, manufacture, and intended use of the equipment. Motorola, Inc. assumes no liability for the customer's failure to comply with these requirements.

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 the presence of flammable gases or fumes. Operation of any electrical equipment in such an environment constitutes a definite safety hazard.

Keep Away From Live Circuits.

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

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, avoid rough handling or jarring of the equipment. Handling of a CRT should be done only by qualified maintenance personnel using approved safety mask and gloves.

Do Not Substitute Parts or Modify Equipment.

Because of the danger of introducing additional hazards, 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 safety features are maintained.

Observe Dangerous Procedure Warnings.

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

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

Flammability

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

EMI Caution



CAUTION

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

Lithium Battery Caution

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



CAUTION

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



CAUTION

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



CAUTION

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

CE Notice (European Community)

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

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

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

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

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

Disclaimer of Warranty

Unless otherwise provided by written agreement with Motorola, Inc., the software and the documentation are provided on an “as is” basis and without warranty. This disclaimer of warranty is in lieu of all warranties whether express, implied, or statutory, including implied warranties of merchantability or fitness for any particular purpose.

© Copyright Motorola 2000
All Rights Reserved

Printed in the United States of America
March 2000

Contents

CHAPTER 1 Hardware Preparation and Installation

Introduction 1-1
Getting Started 1-1
- Overview of Installation Procedure 1-1
- Equipment Required 1-2
- Guidelines for Unpacking 1-2
- ESD Precautions 1-3

Preparing the Board 1-4
- Base Board Configurations 1-5
- VME System Controller Selection 1-8
- SIM Selection for Serial Port B 1-9
  - Removal of Existing SIM 1-10
  - Installation of New SIM 1-11
- Serial Port 1/Console Clock 1-11
- Serial Port 2 Clock 1-12
- Flash Write Protection 1-13
- SRAM Backup Power Source 1-13
- EPROM Size 1-14
- General-Purpose Readable Jumpers 1-15
- IP Bus Clock 1-16
- IP Bus Strobe 1-17
- IP DMA Snoop Control 1-18

Installation Instructions 1-19
- IP Installation on the MVME162FX 1-19
- MVME162FX Installation 1-20
- System Considerations 1-22
- Serial Connections 1-24

CHAPTER 2 Startup and Operation

Introduction 2-1
- Front Panel Switches and Indicators 2-1

Initial Conditions 2-2
Applying Power 2-3
- Pre-Startup Checklist 2-4

Bringing up the Board 2-5

Autoboot 2-9
ROMboot 2-11
Network Boot 2-12
Restarting the System 2-12
Reset 2-13
Abort 2-14
Break 2-14
Diagnostic Facilities 2-14

CHAPTER 3 162Bug Firmware

Introduction 3-1
162Bug Overview 3-1
162Bug Implementation 3-3
Memory Requirements 3-4
Using 162Bug 3-5
Debugger Commands 3-6
Modifying the Environment 3-9
CNFG - Configure Board Information Block 3-9
ENV - Set Environment 3-11
Configuring the 162Bug Parameters 3-11
Configuring the IndustryPacks 3-19

CHAPTER 4 Functional Description

Introduction 4-1
Summary of Features 4-1
Processor and Memory 4-3
I/O Implementation 4-3
ASICs 4-4
Block Diagram 4-4
Functional Description 4-4
Data Bus Structure 4-4
Microprocessor 4-6
MC68xx040 Cache 4-6
No-VMEbus-Interface Option 4-7
Memory Options 4-7
DRAM 4-7
SRAM 4-8
About the Batteries 4-9
EPROM and Flash Memory 4-11

Battery-Backed-Up RAM and Clock 4-12
VMEbus Interface and VMEchip2 4-12
I/O Interfaces 4-13
Serial Communications Interface 4-13
IndustryPack (IP) Interfaces 4-17
Ethernet Interface 4-17
SCSI Interface 4-18
SCSI Termination 4-19
Local Resources 4-19
Programmable Tick Timers 4-19
Watchdog Timer 4-19
Software-Programmable Hardware Interrupts 4-20
Local Bus Timeout 4-20
Local Bus Arbiter 4-21
Connectors 4-21
Remote Status and Control 4-21

CHAPTER 5 Pin Assignments

Connector Pin Assignments 5-1
IndustryPack A, B, C, D Connectors 5-2
Remote Reset Connector 5-4
Memory Mezzanine Connector 1 5-5
Memory Mezzanine Connector 2 5-6
Serial Interface Module Connector 5-7
Serial Port 2 Connector 5-8
Serial Port 1/Console Connector 5-9
VMEbus Connectors 5-9

APPENDIX A Specifications

Board Specifications A-1
Cooling Requirements A-2
Special Considerations for Elevated-Temperature Operation A-3
EMC Regulatory Compliance A-4

APPENDIX B Troubleshooting

Solving Startup Problems B-1

APPENDIX C Network Controller Data

Network Controller Modules Supported C-1

APPENDIX D Disk/Tape Controller Data

Controller Modules Supported D-1
Default Configurations D-2
IOT Command Parameters D-5

APPENDIX E Related Documentation

MCG Documents E-1
Manufacturers' Documents E-2
Related Specifications E-3

List of Figures

Figure 1-1. MVME162FX Layout (01-W3960Bxxx Base Board) 1-6
Figure 1-2. MVME162FX Layout (01-W3182Fxxx Base Board) 1-7
Figure 1-3. Serial Interface Module, Connector Side 1-9
Figure 1-4. EIA-232-D Connections to MVME712M (Sheet 1 of 6) 1-25
Figure 1-5. EIA-530 Connections (Sheet 1 of 2) 1-31
Figure 1-6. EIA-232-D Connections to MVME712A/AM/-12/-13 (Sheet 1 of 4) 1-33
Figure 1-7. EIA-485/EIA-422 Connections 1-37
Figure 2-1. MVME162FX/Firmware System Startup 2-3
Figure 4-1. MVME162FX Block Diagram 4-5

List of Tables

Table 1-1. Startup Overview 1-1
Table 1-2. MVME162FX Configuration Settings 1-5
Table 1-2. Serial Interface Module Part Numbers 1-10
Table 1-3. J26/J2 Snoop Control Encoding 1-18
Table 2-1. MVME162FX Front Panel Controls 2-1
Table 2-2. Software-Readable Jumpers J22/J28 2-6
Table 3-1. Memory Offsets with 162Bug 3-4
Table 3-2. Debugger Commands 3-6
Table 3-3. ENV Command Parameters 3-11
Table 4-1. MVME162FX Features 4-1
Table 4-2. Local Bus Arbitration Priority 4-21
Table 5-1. IndustryPack Interconnect Signals 5-3
Table 5-2. Remote Reset Connector Pin Assignments 5-4
Table 5-3. Mezzanine Connector 1 Pin Assignments 5-5
Table 5-4. Mezzanine Connector 2 Pin Assignments 5-6
Table 5-5. Serial Port Module Connector Pin Assignments 5-7
Table 5-6. Serial Port 2 Connector Pin Assignments 5-8
Table 5-7. Serial Port 1 Connector Pin Assignments 5-9
Table 5-8. VMEbus Connector P1 Pin Assignments 5-10
Table 5-9. VMEbus Connector P2 Pin Assignment 5-11
Table A-1. MVME162FX Specifications A-1
Table B-1. Troubleshooting MVME162FX Boards B-1
Table E-1. Motorola Computer Group Documents E-1
Table E-2. Manufacturers' Documents E-2
Table E-3. Related Specifications E-3

Hardware Preparation and Installation

1

Introduction

This chapter provides unpacking instructions, hardware preparation guidelines, and installation instructions for the MVME162FX VME Embedded Controller. Hardware preparation of the MVME712 series transition modules compatible with this board is covered in separate manuals.

Getting Started

This section supplies an overview of startup procedures applicable to the MVME162FX. 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.
Configure jumpers or serial interface modules on the MVME162FX as necessary.	*Base Board Configurations* on page 1-5.
Ensure that IP modules are properly installed on the board.	*IP Installation* on page 1-19.
Install the board in a chassis.	*MVME162FX Installation* on page 1-20.
Connect a display terminal.	*Serial Connections* on page 1-24.

Table 1-1. Startup Overview (Continued)

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

Equipment Required

The following equipment is necessary to complete an MVME162FX 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 MVME162FX board.

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

If no ESD station is available, you can avoid damage resulting from ESD by wearing an antistatic wrist strap (available at electronics stores). Place the strap around your wrist and attach the grounding end (usually a piece of copper foil or an alligator clip) to an electrical ground. An electrical ground can be a piece of metal that literally runs into the ground (such as an unpainted metal pipe) or a metal part of a grounded electrical appliance. An appliance is grounded if it has a three-prong plug and is plugged into a three-prong grounded outlet. You cannot use the chassis in which you are installing the MVME162FX 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 MVME162FX, you may need to reconfigure hardware to some extent before installing the module.

Most options on the MVME162FX are under software control: By setting bits in control registers after installing the module in a system, you can modify its configuration. (The MVME162FX registers are described in Chapter 3 under *ENV - Set Environment*, and/or in the *MVME162FX Embedded Controller Programmer's Reference Guide* as listed in "Related Documentation" in Appendix E.)

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

Base Board Configurations

Figure 1-1 illustrates the placement of the jumper headers, connectors, panel controls, and various other components on MVME162FX models with 01-W3960B*xxx* series base boards. Figure 1-2 illustrates the placement of the equivalent items on MVME162FX models with 01-W3182F*xxx* series base boards. (Refer to the table in the Preface for the corresponding model numbers.)

Manually configurable items on both versions of the board are listed in the following table.

Table 1-2. MVME162FX Configuration Settings

Function	Reference Designator		Factory Default
	01-W3960	01-W3182	
VME System Controller Selection	J1	J1	2-3
SIM Selection for Serial Port B	J10	J15	SIMM06
Serial Port 1/Console Clock	J11	J16	No jumpers
Serial Port 2 Clock	J12	J17	No jumpers
Flash Write Protection	N/A	J24	1-2
SRAM Backup Power Source	J20	J22	1-3, 2-4
EPROM Size	J21	J23	2-3
General-Purpose Readable Jumpers	J22	J28	All jumpers on
IP Bus Clock	J24	J14	1-2
IP Bus Strobe	J25	J19	Off
IP DMA Snoop Control	J26	J2	On-On

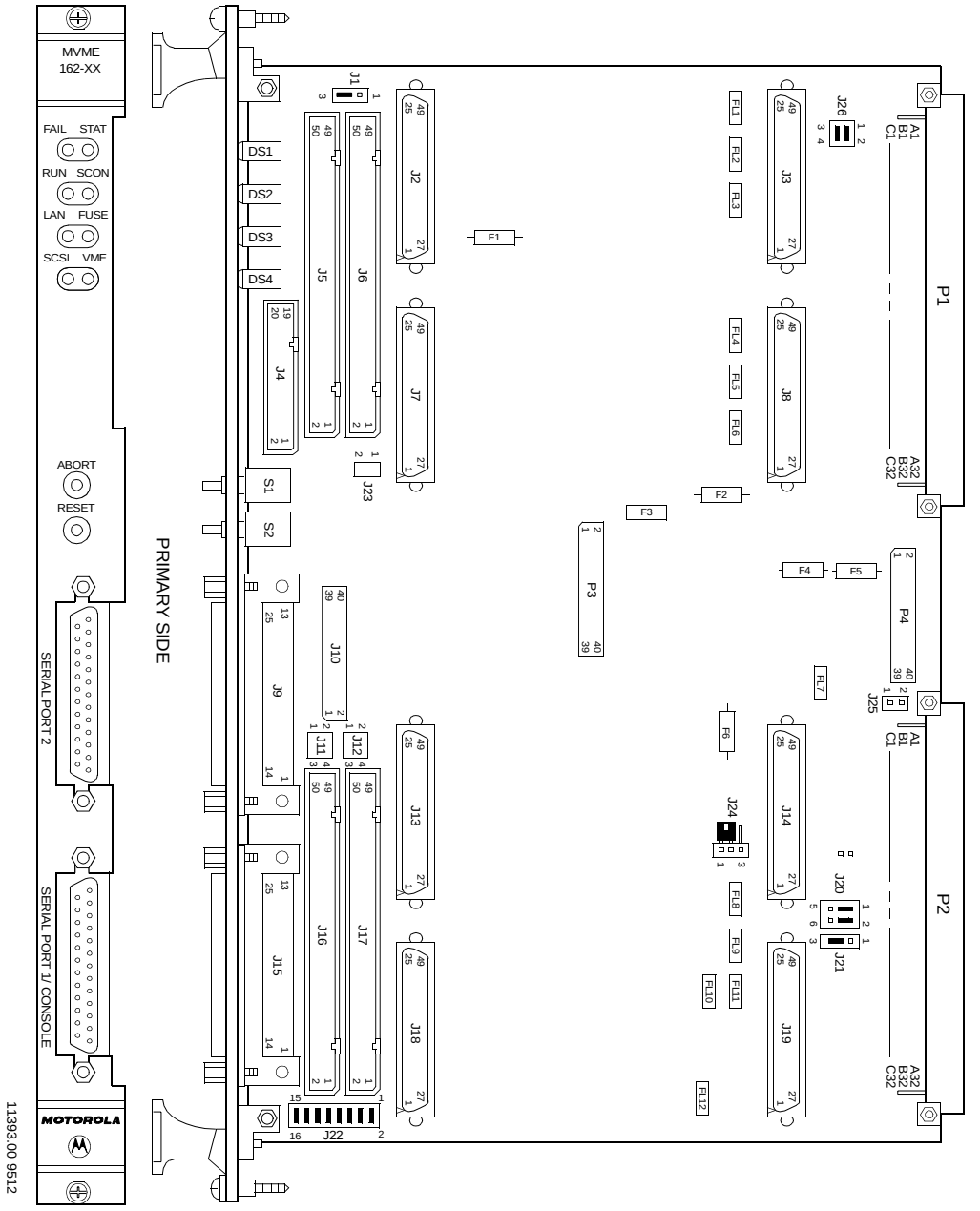


Figure 1-1. MVME162FX Layout (01-W3960B*xxx* Base Board)

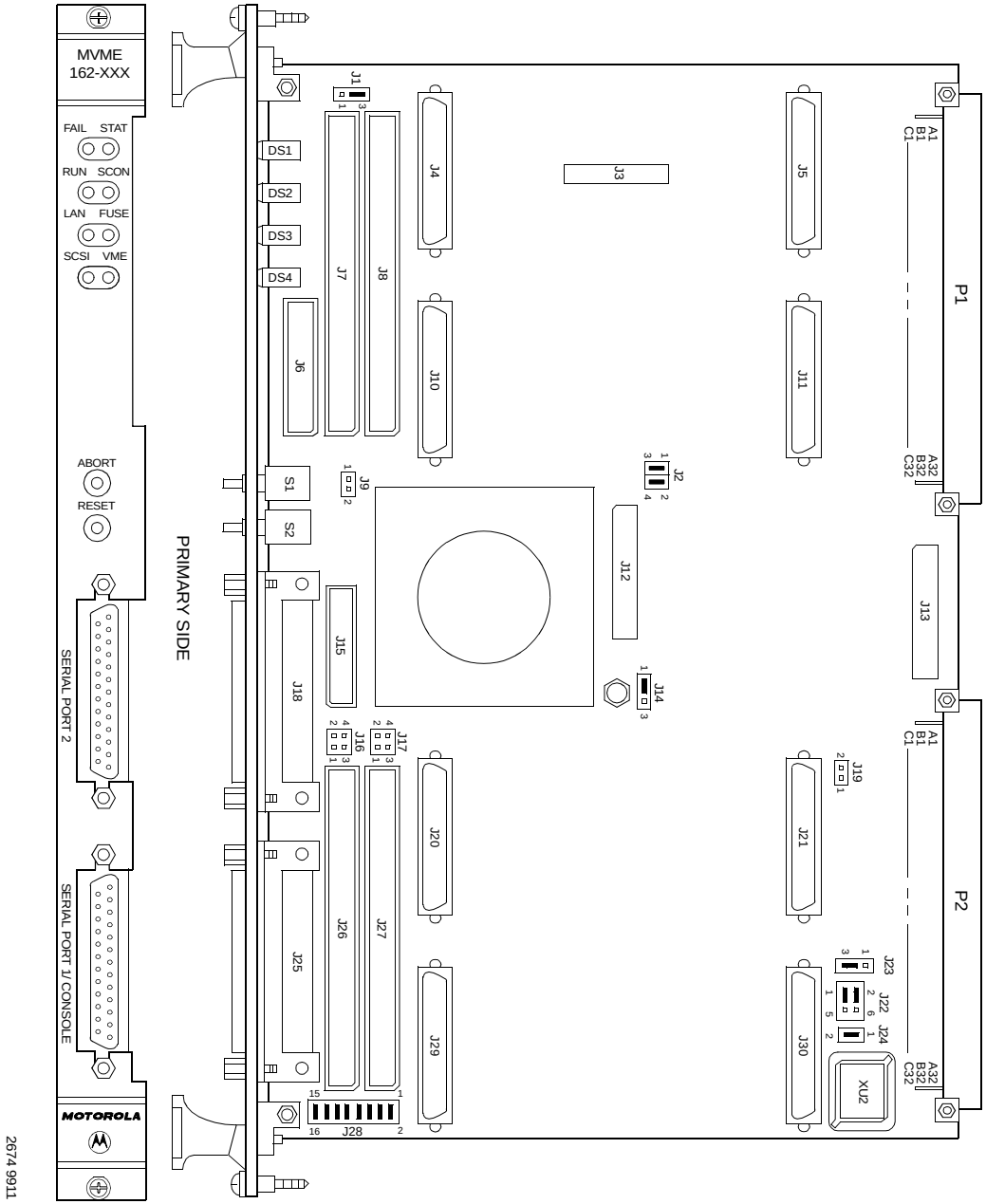


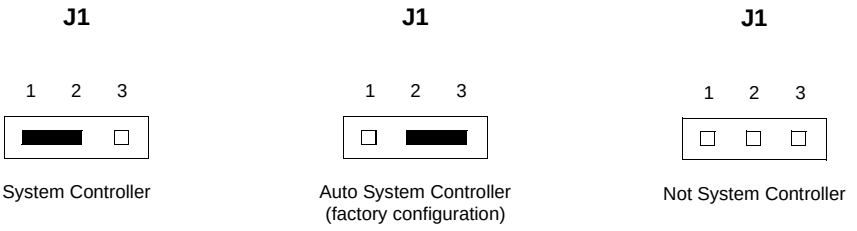
Figure 1-2. MVME162FX Layout (01-W3182F*xxx* Base Board)

VME System Controller Selection

The MVME162FX (in both 01-W3960B*xxx* and 01-W3182F*xxx* base board versions) is factory-configured in "automatic" system controller mode with a jumper across J1 pins 2-3. In this configuration, the MVME162FX 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 MVME162FX to function as system controller in all cases, move the jumper to pins 1-2. If the MVME162FX is not to be the system controller under any circumstances, remove the jumper from J1.

Note On MVME162FX 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.



SIM Selection for Serial Port B

Port B of the MVME162FX Z85230 serial communications controller is manually configurable by means of serial interface modules (SIMs) that you install 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) to the 25-pin DB25 front panel connector marked SERIAL PORT 2.

On 01-W3960B*xxx* series base boards, the SIM connector is J10 (see Figure 1-1). On 01-W3182F*xxx* series base boards, the SIM connector is J15 (see Figure 1-2).

Figure 1-3 illustrates the secondary side (bottom) of a serial interface module, showing the J1 connector which plugs into SIM connector J10 or J15 respectively on the MVME162FX. Figure 1-4 (sheets 3-6), Figure 1-5, Figure 1-6 (sheets 3 and 4), and Figure 1-7 illustrate the nine configurations available for Port B.

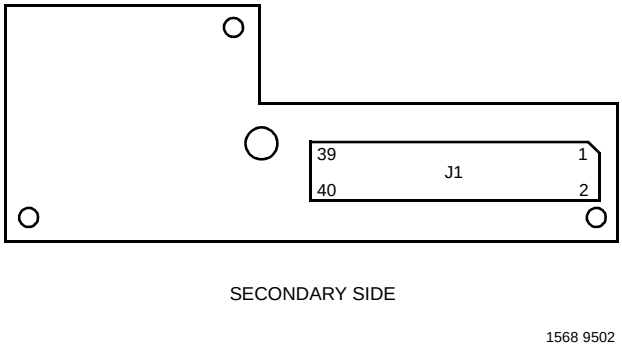


Figure 1-3. Serial Interface Module, Connector Side

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 MVME162FX connector J10/J15. Turn the SIM so that the keys line up and place it gently on connector J10/J15, aligning the mounting hole(s) at the SIM corners (or center) with the matching standoff(s) on the MVME162FX.
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). Screw it or them into the standoff(s) but do not overtighten it or them.

The signal relationships and signal connections in the various serial configurations available for ports A and B are illustrated in Figures 1-4 through 1-7.

Serial Port 1/Console Clock

On 01-W3960B*xxx* series base boards, the **SERIAL PORT 1/CONSOLE** header is J11 (see Figure 1-1). On 01-W3182F*xxx* series base boards, this header is J16 (see Figure 1-2).

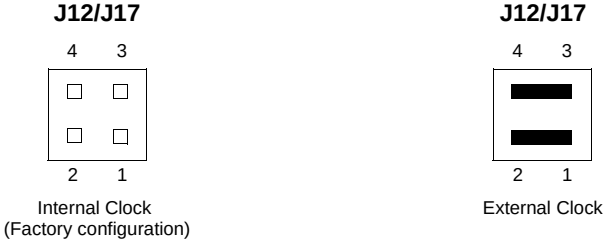
The MVME162FX is shipped from the factory with the **SERIAL PORT 1/CONSOLE** header configured for asynchronous communications (i.e., jumpers removed). To select synchronous communications for the **SERIAL PORT 1/CONSOLE** connection, install jumpers across pins 1-2 and pins 3-4.



Serial Port 2 Clock

On 01-W3960B*xxx* series base boards, the **SERIAL PORT 2** header is J12 (see Figure 1-1). On 01-W3182F*xxx* series base boards, this header is J17 (see Figure 1-2).

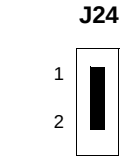
The MVME162FX is shipped from the factory with the **SERIAL PORT 2** header configured for asynchronous communications (i.e., jumpers removed). To select synchronous communications for the **SERIAL PORT 2** connection, install jumpers across pins 1-2 and pins 3-4.



Flash Write Protection

No header for Flash write protection is present on 01-W3960B*xxx* series base boards. On 01-W3182F*xxx* series base boards, this header is J24 (see Figure 1-2).

When the Flash write protect jumper is installed (factory configuration), Flash memory can be written to via the normal software routines. When the jumper is removed, Flash memory cannot be written to.



Flash Write Protect
(Factory configuration)

SRAM Backup Power Source

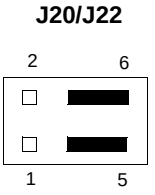
This header determines the source for onboard static RAM backup power on the MVME162FX. On 01-W3960B*xxx* series base boards, the SRAM backup power source header is J20 (see Figure 1-1). On 01-W3182F*xxx* series base boards, this header is J22 (see Figure 1-2).

The backup power configurations available for onboard SRAM through header J22 are illustrated in the following diagram. In the factory configuration, VMEbus +5V standby voltage serves as primary and secondary power source (the onboard battery is disconnected).

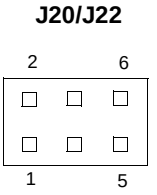
Note For MVME162FXs without the optional VMEbus interface (i.e., without the VMEchip2 ASIC), you must select the onboard battery as the backup power source.

Caution

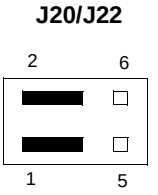
Removing all jumpers may temporarily disable the SRAM.
Do not remove all jumpers from J20/J22, except for storage.



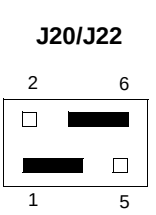
Primary Source Onboard Battery
Secondary Source Onboard Battery



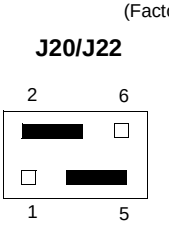
Backup Power Disabled
(For storage only)



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



Primary Source VMEbus +5V STBY
Secondary Source Onboard Battery



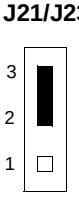
Primary Source Onboard Battery
Secondary Source VMEbus +5V STBY

EPROM Size

Header J21 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J23 (on 01-W3182F*xxx* series base boards; see Figure 1-2) selects the EPROM size. The MVME162FX is factory-configured for a 4Mbit EPROM.



4Mbit EPROM
(Factory configuration)



8Mbit EPROM

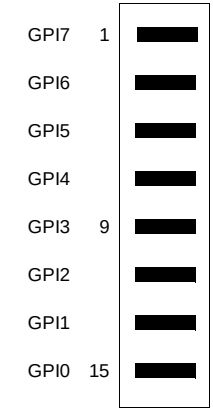
General-Purpose Readable Jumpers

Header J22 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J28 (on 01-W3182F*xxx* series base boards; see Figure 1-2) provides eight readable jumpers. These jumpers are read as a register (at $FFF4202D) in the MC2chip LCSR (local control/status register). The bit values are read as a zero when the jumper is installed and as a one when the jumper is removed.

With the factory-installed 162Bug firmware in place, four jumpers are user-definable (pins 1-2, 3-4, 5-6, 7-8). If the 162Bug firmware is removed, seven jumpers are user-definable (pins 1-2, 3- 4, 5-6, 7-8, 11-12, 13-14, 15-16).

Note Pins 9-10 (GPI3) are reserved to select either the Flash memory map (jumper installed) or the EPROM memory map (jumper removed). They are not user-definable. See Chapter 3 for more information.

In most cases, the MVME162FX is shipped from the factory with J22/J28 set to all zeros (jumpers on all pins). *On boards built with the no-VMEbus option*, however, no jumper is installed across pins 9-10.



	162Bug INSTALLED	USER CODE INSTALLED
GPI7	USER-DEFINABLE	USER-DEFINABLE
GPI6	USER-DEFINABLE	USER-DEFINABLE
GPI5	USER-DEFINABLE	USER-DEFINABLE
GPI4	USER-DEFINABLE	USER-DEFINABLE
GPI3	IN=FLASH; OUT=EPROM	IN=FLASH; OUT=EPROM
GPI2	REFER TO 162Bug MANUAL	USER-DEFINABLE
GPI1	REFER TO 162Bug MANUAL	USER-DEFINABLE
GPI0	REFER TO 162Bug MANUAL	USER-DEFINABLE

Flash Selected (factory configuration except on non-VMEbus models)

IP Bus Clock

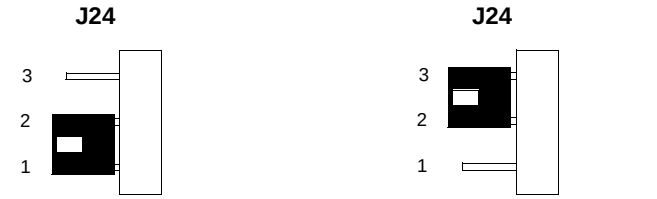
Header J24 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J14 (on 01-W3182F*xxx* series base boards; see Figure 1-2) selects the speed of the IP bus clock. The IP bus clock speed may be 8MHz or it may be set synchronous to the processor bus clock (25MHz or 32MHz for the MC68040 and MC68LC040). The default factory configuration has a jumper installed on pins 1-2, denoting an 8MHz clock.

If the jumper is installed on J24/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 (IP2 chip, 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.

01-W3960B*xxx* Boards



IP Bus Clock = 8MHz (Factory configuration)

IP Bus Clock = Processor Bus Speed (from MPU Bus Clock)

01-W3182F*xxx* Boards



IP Bus Clock = 8MHz (Factory configuration)

IP Bus Clock = Processor Bus Speed (from MPU Bus Clock)

IP Bus Strobe

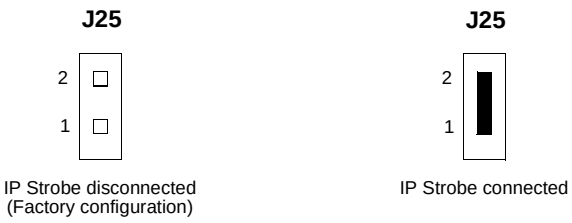
Some IP bus implementations make use of the Strobe∗ signal (pin 46) as an input to the IP modules from the IP2 chip. Other IP interfaces require that the strobe be disconnected.

With a jumper installed between pins 1-2 of the IP Bus Strobe Select header, a programmable frequency source is connected to the Strobe∗ signal on the IP bus (for details, refer to the IP2 chip programming model in the *MVME162FX Embedded Controller Programmer's Reference Guide*).

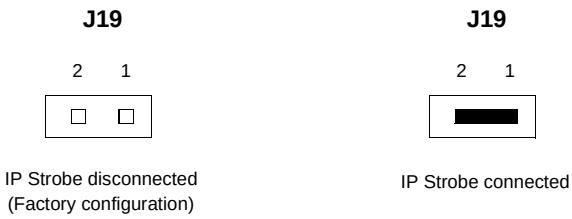
On 01-W3960B*xxx* series base boards, the IP Bus Strobe Select header is J25 (see Figure 1-1). On 01-W3182F*xxx* series base boards, this header is J19 (see Figure 1-2).

If the jumper is removed from J25/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.

01-W3960B*xxx* Boards



01-W3182F*xxx* Boards

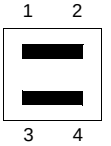


IP DMA Snoop Control

This jumper header defines the state of the snoop control bus when an IP DMA controller is local bus master. On 01-W3960B*xxx* series base boards, the IP DMA Snoop Control header is J26 (see Figure 1-1). On 01-W3182F*xxx* series base boards, this header is J2 (see Figure 1-2). J26/J2 pins 3 and 4 control Snoop Control signal 0. J26/J2 pins 1 and 2 control Snoop Control signal 1.

01-W3960B*xxx* Boards

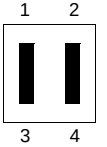
J26



Snoop Inhibited (factory configuration)

01-W3182F*xxx* Boards

J2



Snoop Inhibited (factory configuration)

The following table lists the snoop operations represented by the setting of J26/J2.

Table 1-3. J26/J2 Snoop Control Encoding

Pins 1-2 (SC1)	Pins 3-4 (SC0)	Requested Snoop Operation	
		Alternate Bus Master Read Access	Alternate Bus Master Write Access
0	0	Inhibit Snooping	Inhibit Snooping
0	1	Supply Dirty Data, Leave Dirty Data	Sink Byte/Word/Longword
1	0	Supply Dirty Data, Leave Dirty Data	Invalidate Line
1	1	Reserved (Snoop Inhibited)	Reserved (Snoop Inhibited)

Note Jumper installed = logic 0. Jumper removed = logic 1.

Installation Instructions

This section covers:

- Installation of IndustryPacks (IPs) on the MVME162FX
- Installation of the MVME162FX 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 are set as appropriate.

IP Installation on the MVME162FX

The MVME162FX accommodates up to four IndustryPack (IP) modules. Install the IP modules on the MVME162FX as follows:

1. Each IP module has two 50-pin connectors that plug into a set of two corresponding 50-pin connectors on the MVME162FX. See Figure 1-1 and Figure 1-2 for the IP connector locations on the MVME162FX. Refer to the *IndustryPack A, B, C, D Connectors* section for details of the IP-module-to-connector correspondence.
   - Orient the IP module(s) so that the tapered connector shells mate properly. Plug IP_a, IP_b, IP_c, and IP_d into the connectors designated for each module. If you are using a double-sized IP, plug IP_ab into the four connectors assigned to IP_a and IP_b; plug IP_cd into the four connectors assigned to IP_c and IP_d.
2. Four additional 50-pin IDC connectors are provided behind the MVME162FX 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. (On 01-W3182F*xxx* series base boards, for instance, 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 the IDC connectors 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 MVME162FX front panel and attach them to the appropriate external equipment, depending on the nature of the particular IP(s).

MVME162FX Installation

With EPROM, SIM, and IP modules installed and headers properly configured, proceed as follows to install the MVME162FX 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 MVME162FX.
   – If you intend to use the MVME162FX 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 MVME162FX as system controller, it can occupy any unused double-height card slot.

4. Slide the MVME162FX 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 MVME162FX 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 MVME162FX.

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

8. Connect the P2 Adapter Board or LCP2 Adapter Board and cable(s) to MVME162FX 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 MVME162FX 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 MVME162FX 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 MVME162FX may not function properly without its main board connected to VMEbus backplane connectors P1 and P2.

Whether the MVME162FX operates as 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 MC68040/MC68LC040 software. For specifics, refer to the memory maps in the *Programmer's Reference Guide*.

The MVME162FX 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 MVME162FX 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 MVME162FX waits indefinitely for the VMEbus cycle to complete. This will cause the system to lock up. There is only one situation in which the system might lack this global bus timeout: when the MVME162FX is not the system controller and there is no global bus timeout elsewhere in the system.

Multiple MVME162FXs may be installed in a single VME chassis. In general, hardware multiprocessor features are supported.

Note If you are installing multiple MVME162FXs in an MVME945 chassis, do not install an MVME162FX 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 MVME162FX processor to broadcast a signal to any other MVME162FX 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:

Circuit and Fuse	
01-W3960B*xxx* Boards	**01-W3182F*xxx* Boards**
IndustryPack +5V (F4, F5)	IndustryPack +5V (F1, F6)
Remote reset connector +5V (F1	Remote reset connector +5V (F2)
LAN AUI +12V (F2)	LAN AUI +12V (F3)
IndustryPack ±12V (F3, F6)	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 MVME162FX 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 MVME162FX 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 MVME162FX hardware supports asynchronous serial baud rates of 110b/s to 38.4Kb/s.

For additional information on the MVME162FX 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.

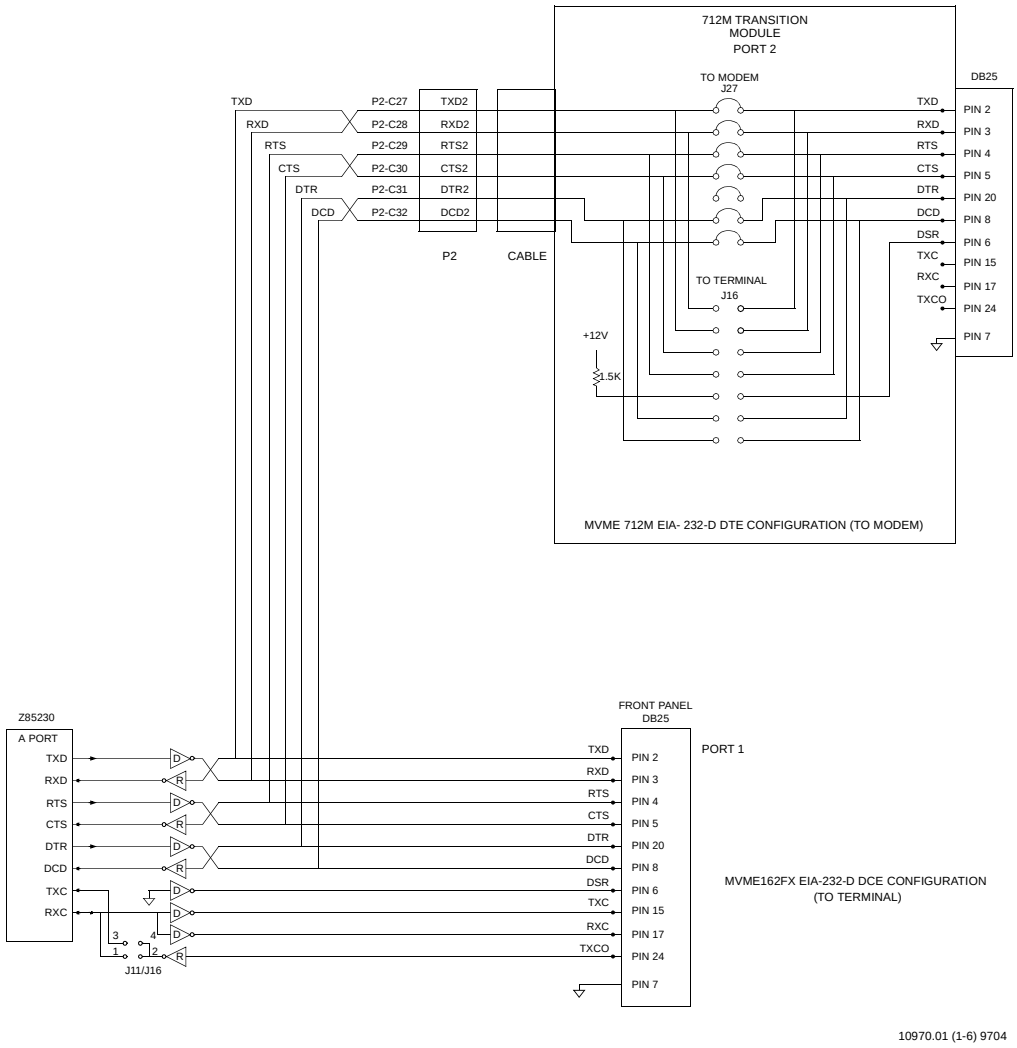


Figure 1-4. EIA-232-D Connections to MVME712M (Sheet 1 of 6)

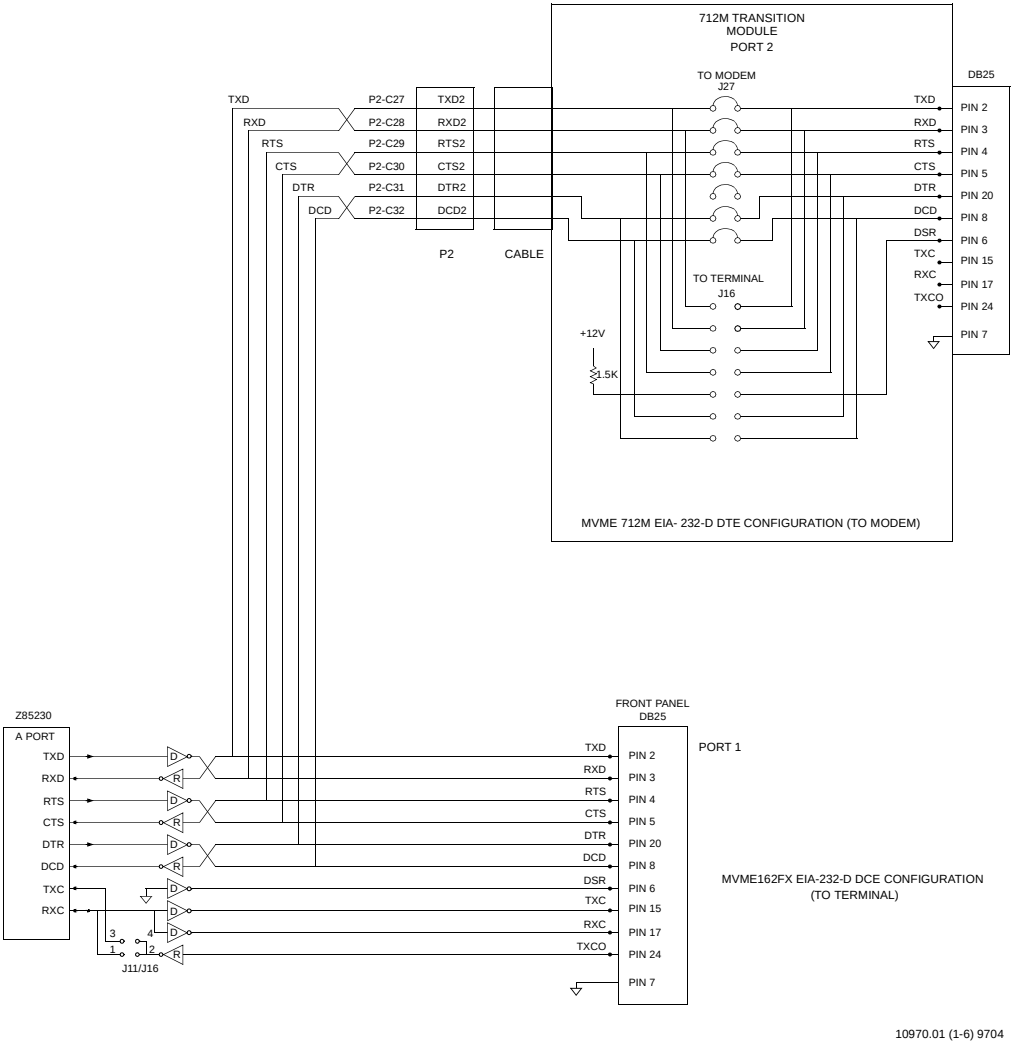


Figure 1-4. EIA-232-D Connections to MVME712M (Sheet 2 of 6)

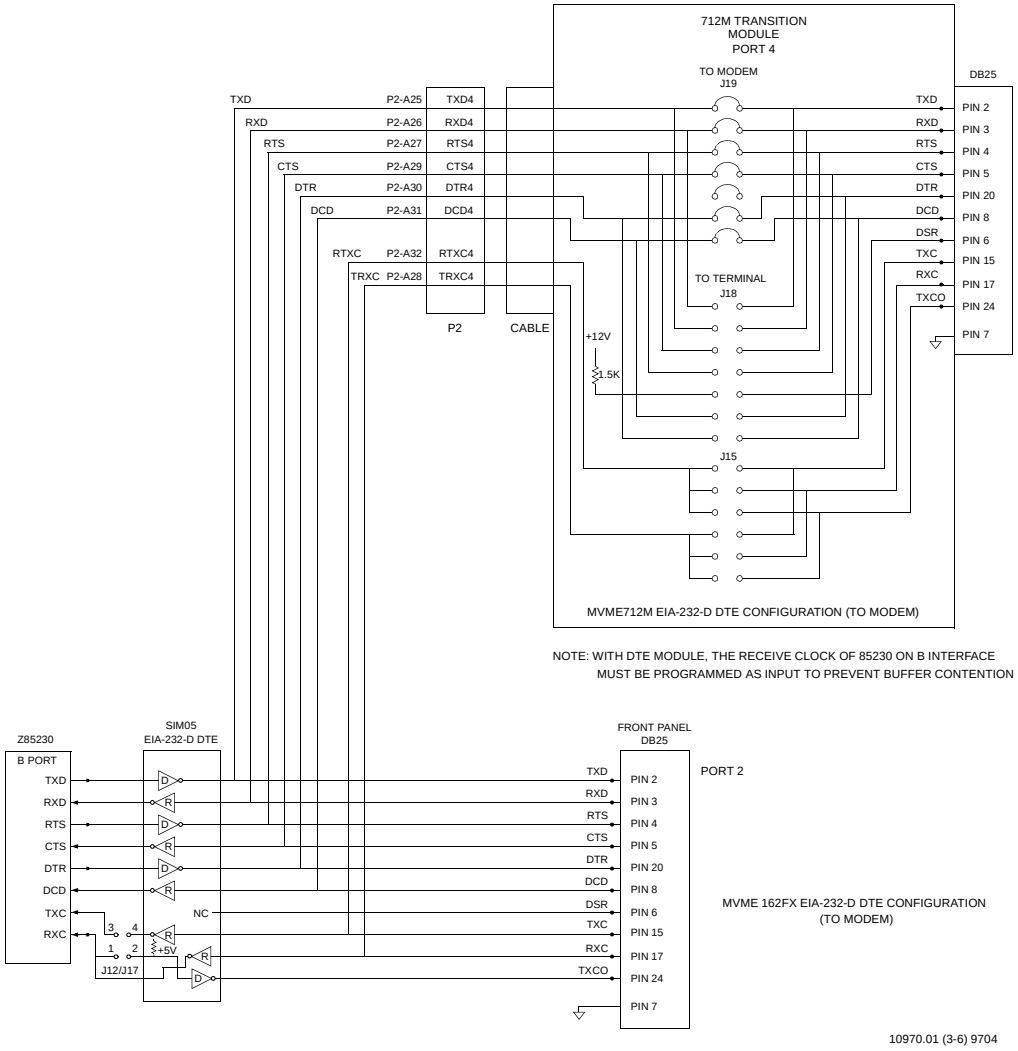


Figure 1-4. EIA-232-D Connections to MVME712M (Sheet 3 of 6)

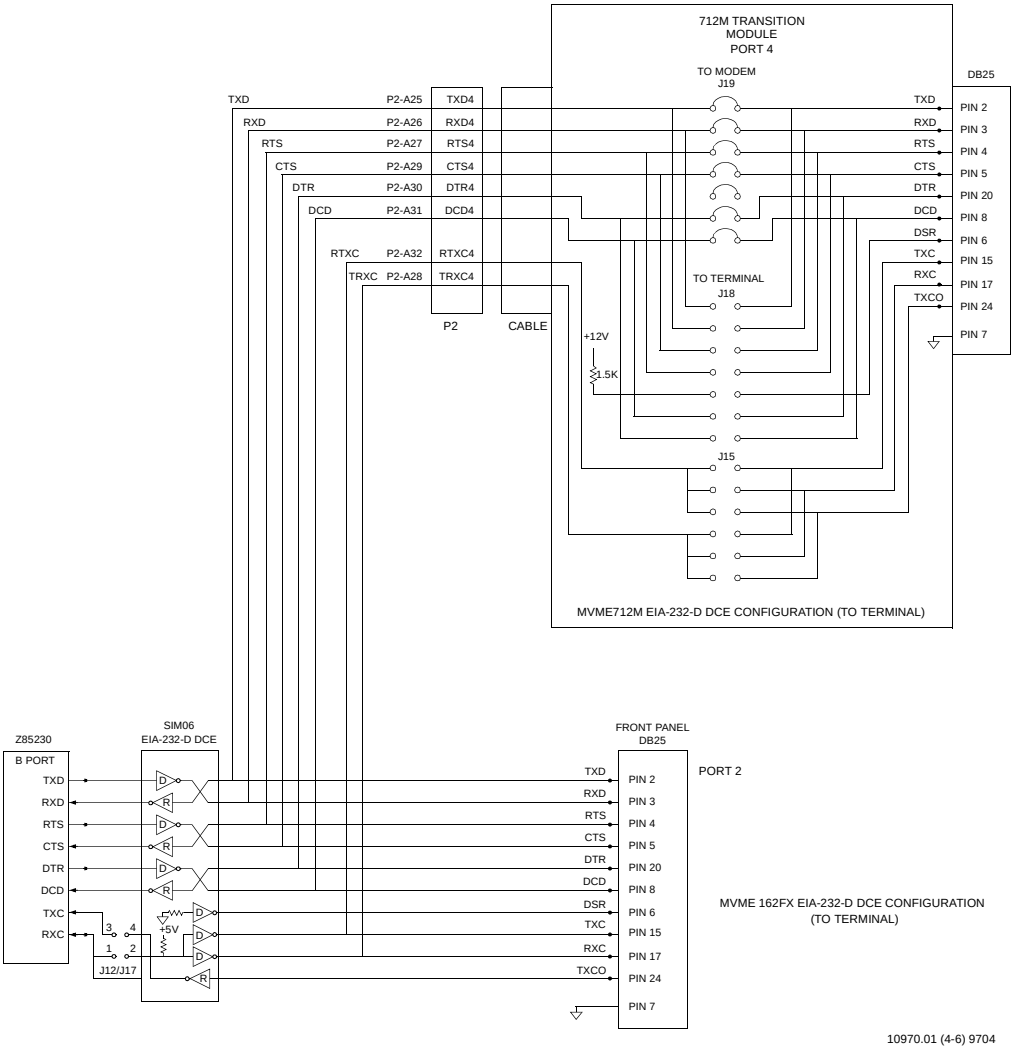


Figure 1-4. EIA-232-D Connections to MVME712M (Sheet 4 of 6)

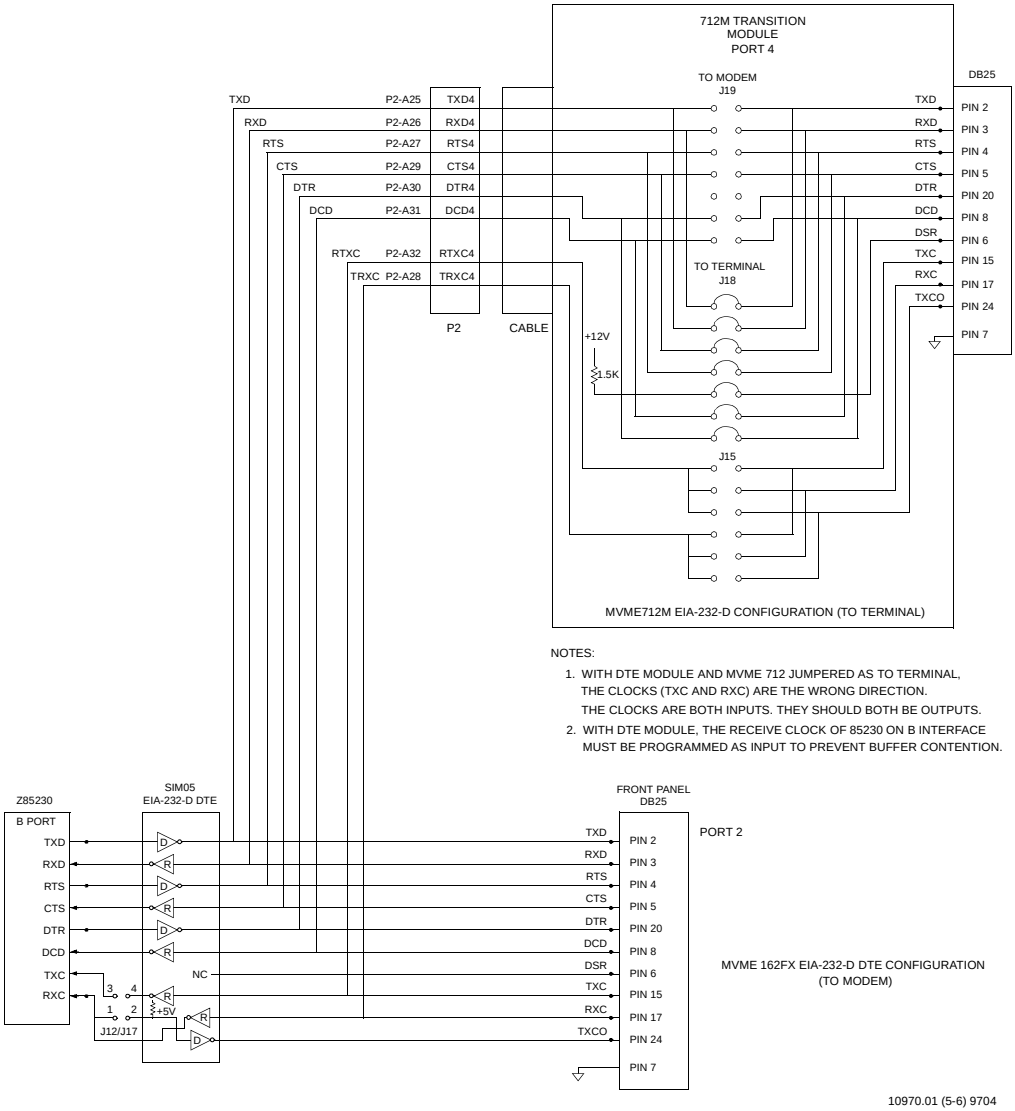


Figure 1-4. EIA-232-D Connections to MVME712M (Sheet 5 of 6)

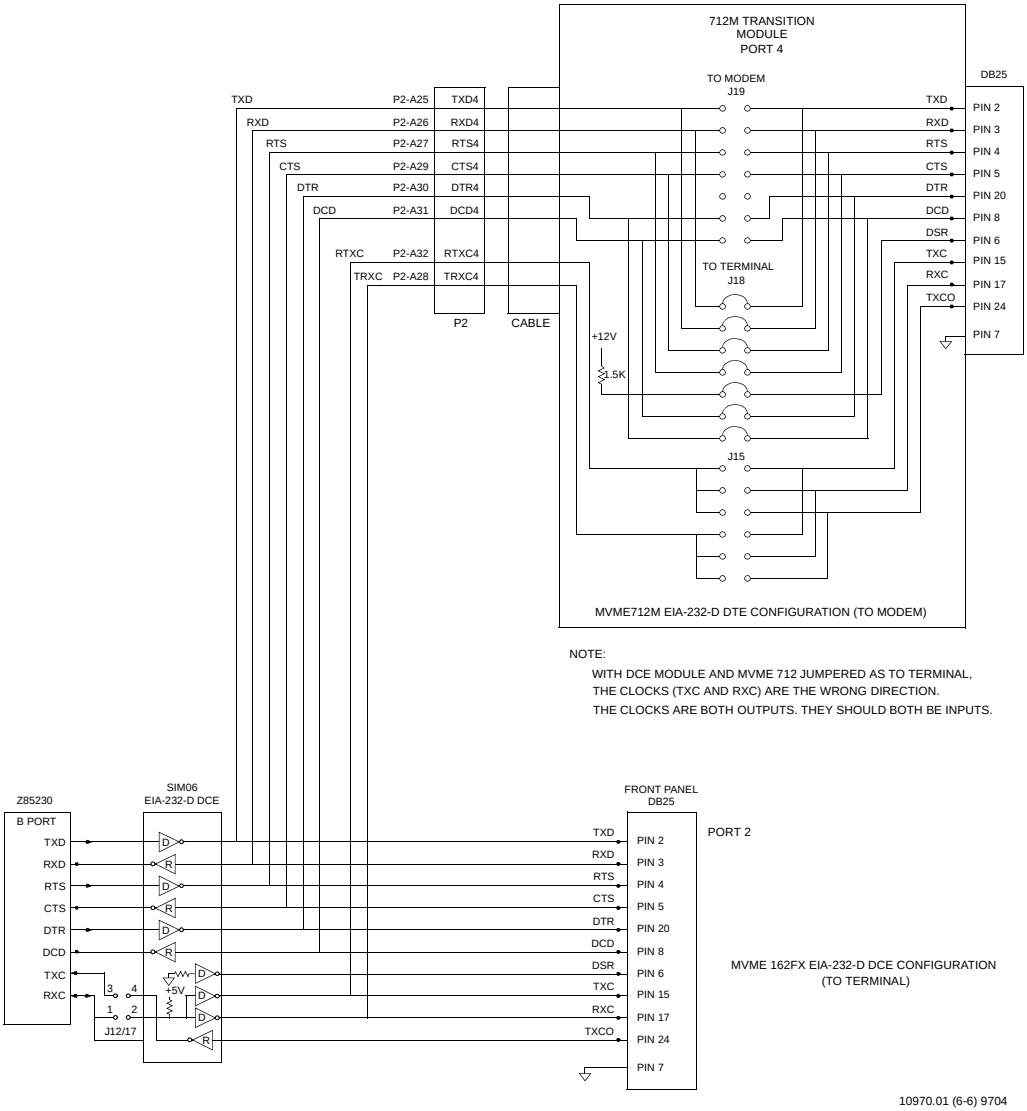


Figure 1-4. EIA-232-D Connections to MVME712M (Sheet 6 of 6)

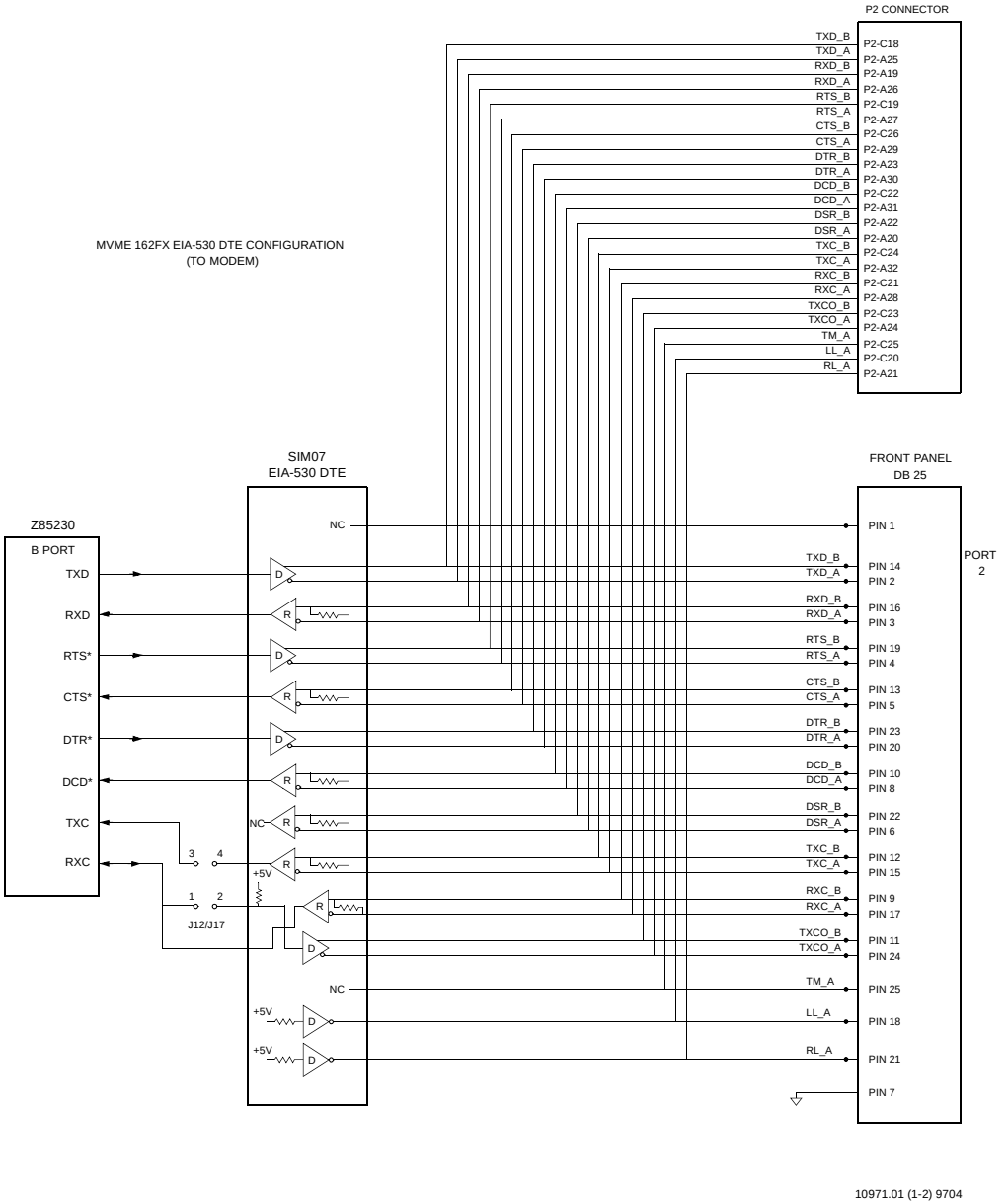


Figure 1-5. EIA-530 Connections (Sheet 1 of 2)

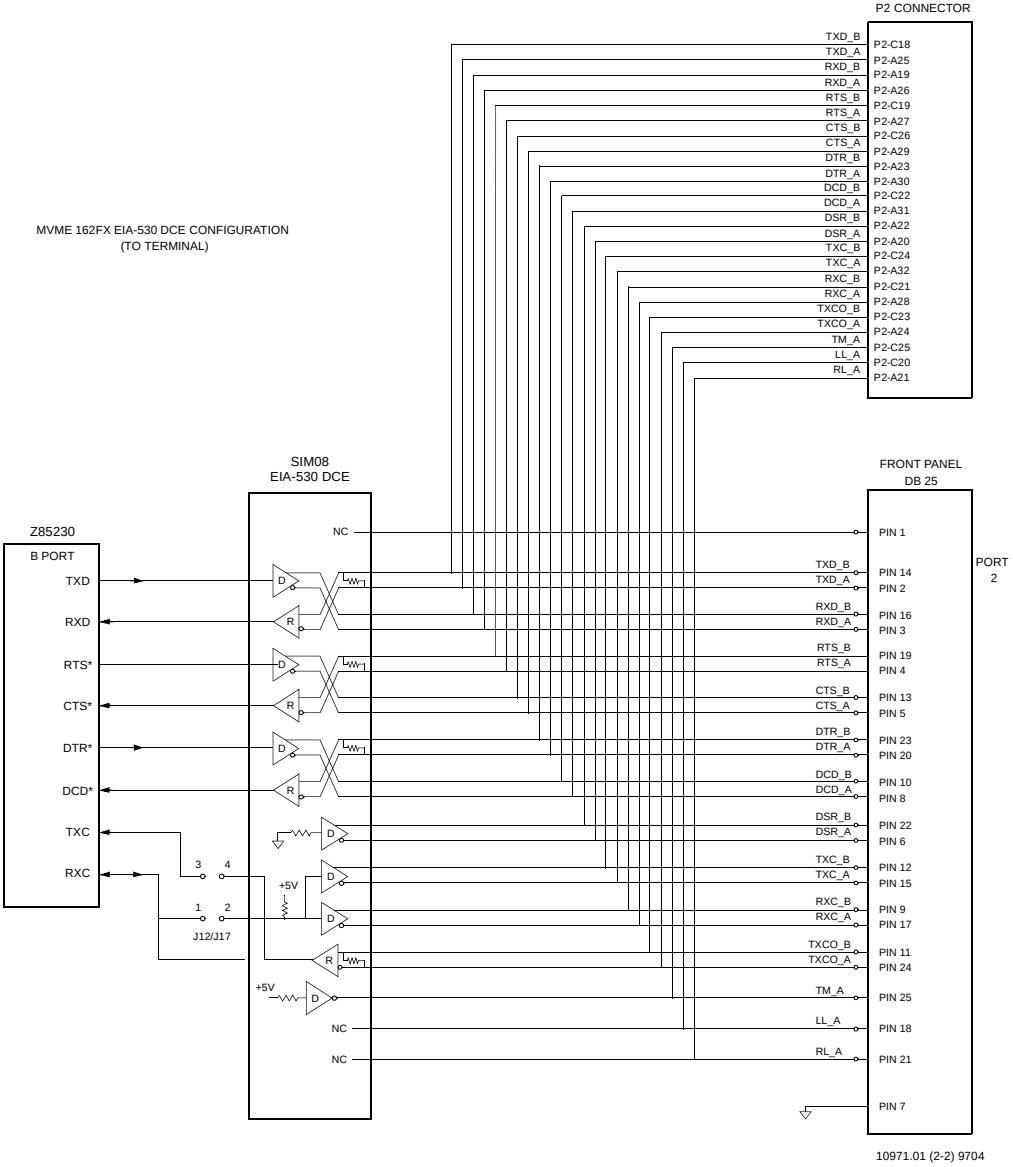


Figure 1-5. EIA-530 Connections (Sheet 2 of 2)

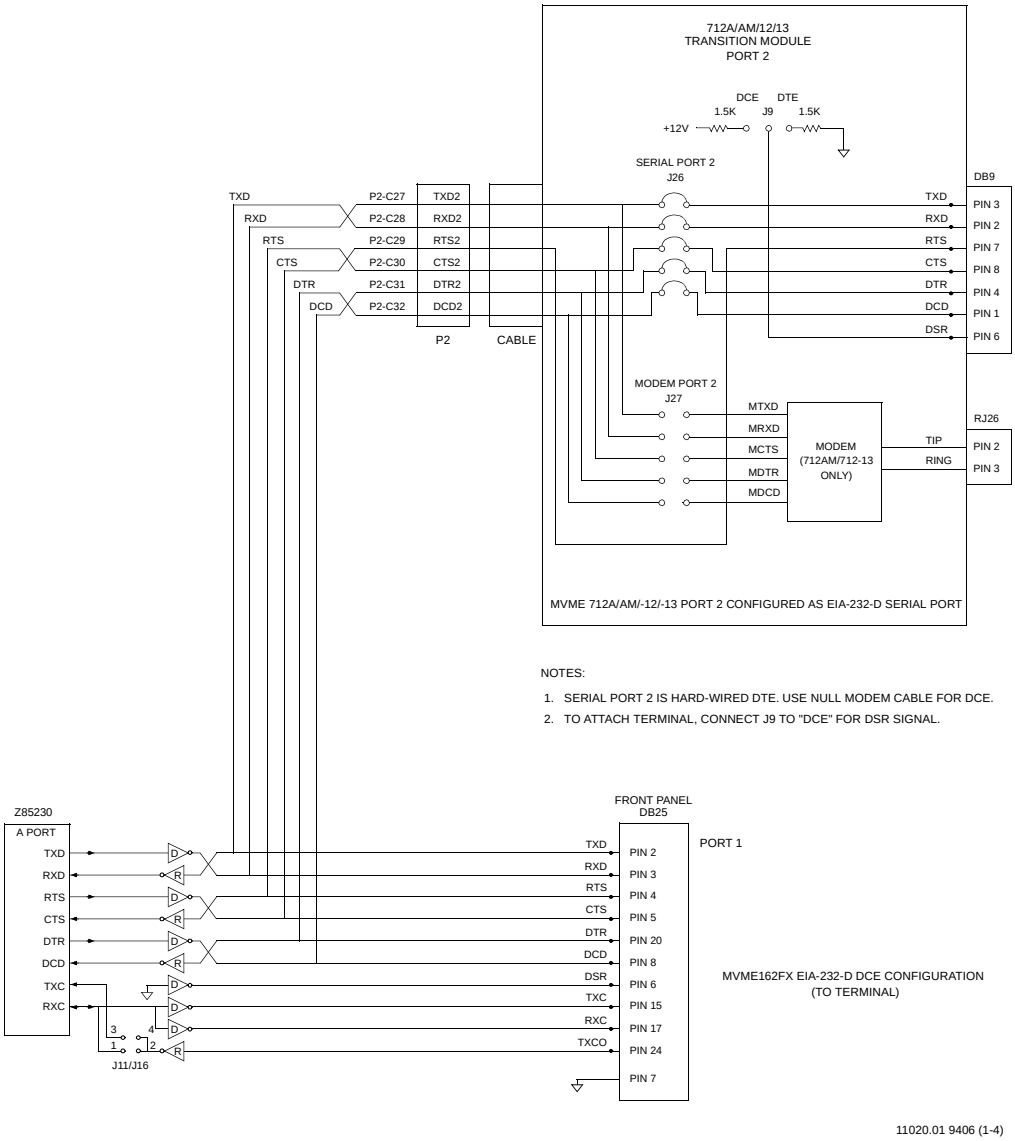


Figure 1-6. EIA-232-D Connections to MVME712A/AM/-12/-13 (Sheet 1 of 4)

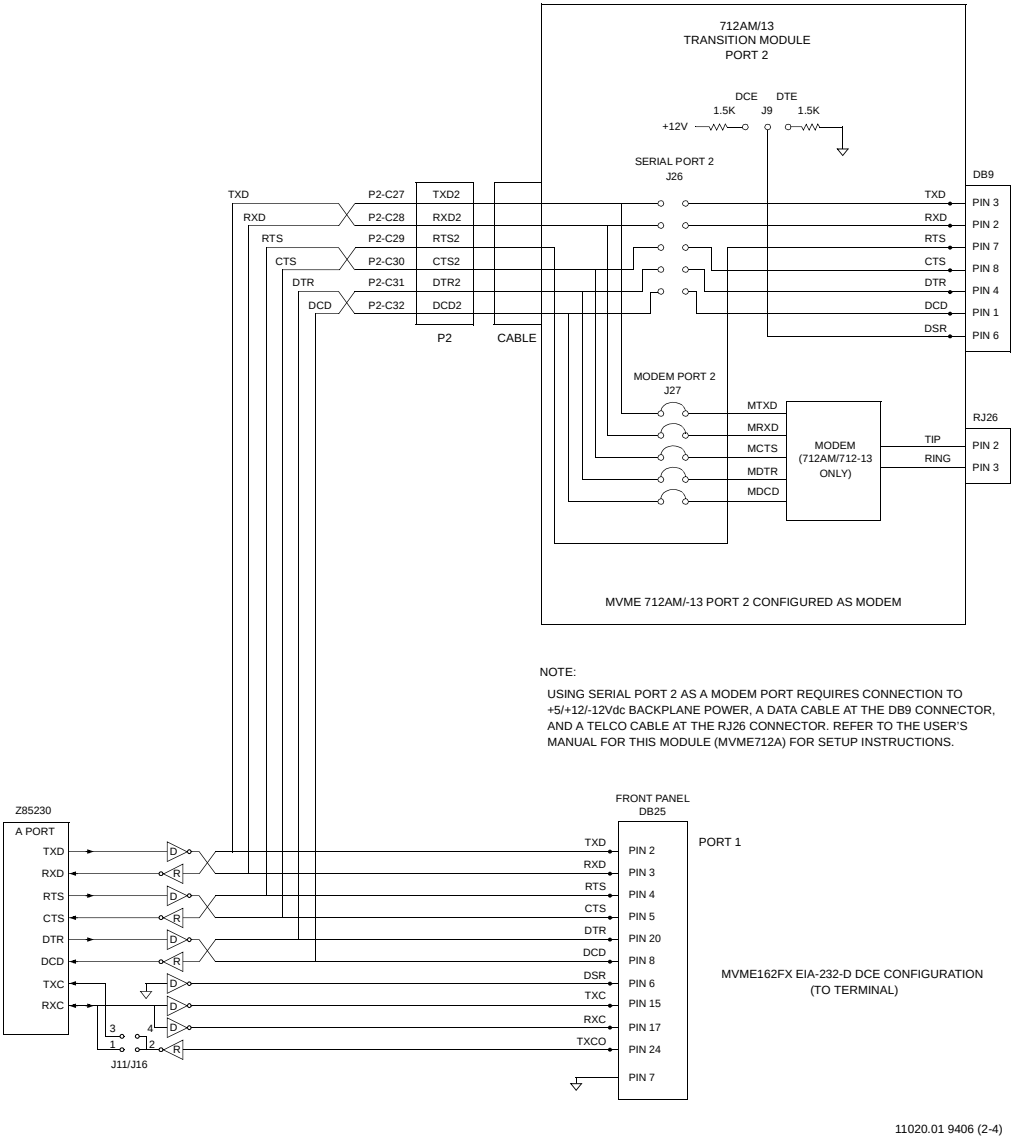


Figure 1-6. EIA-232-D Connections to MVME712A/AM/-12/-13 (Sheet 2 of 4)

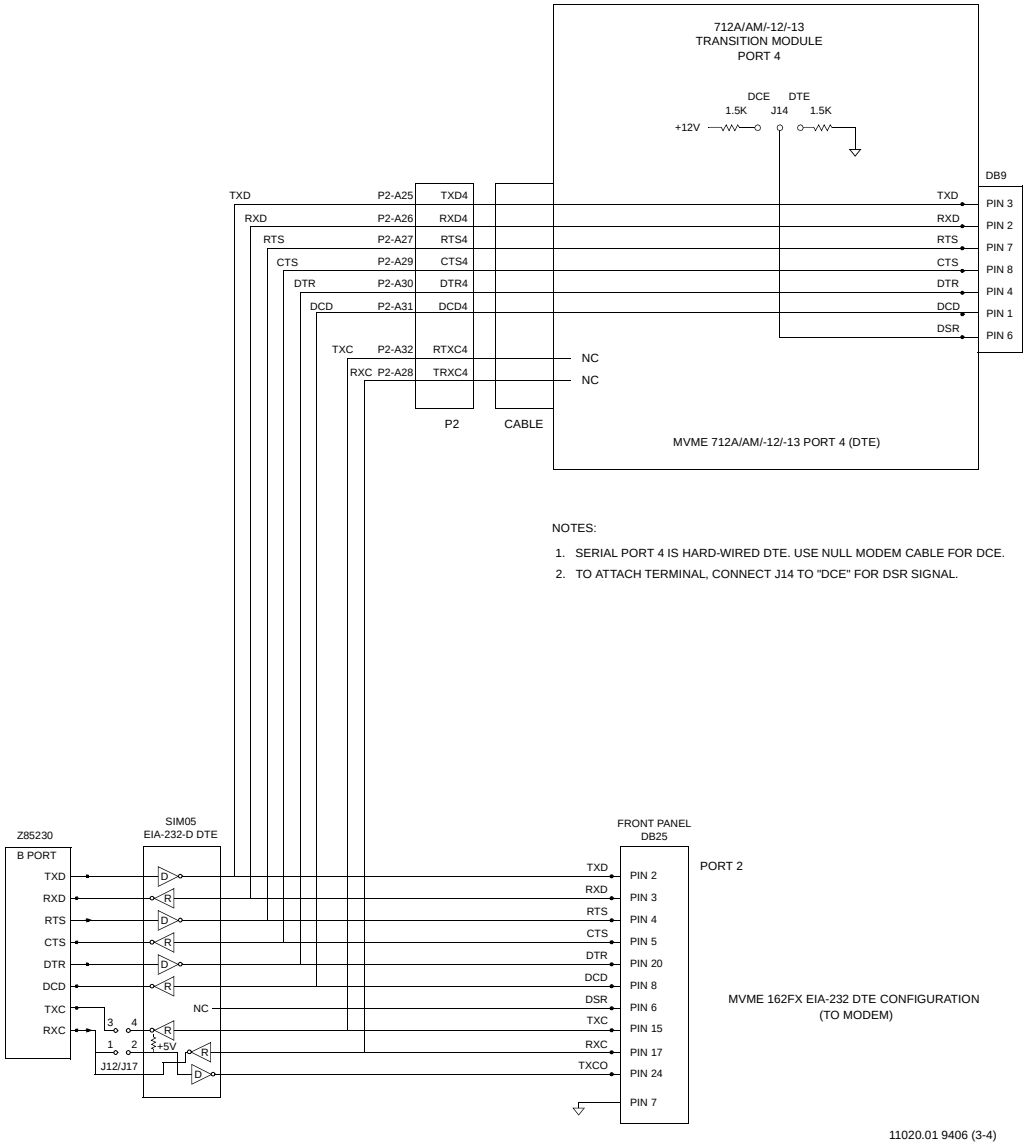


Figure 1-6. EIA-232-D Connections to MVME712A/AM/-12/-13 (Sheet 3 of 4)

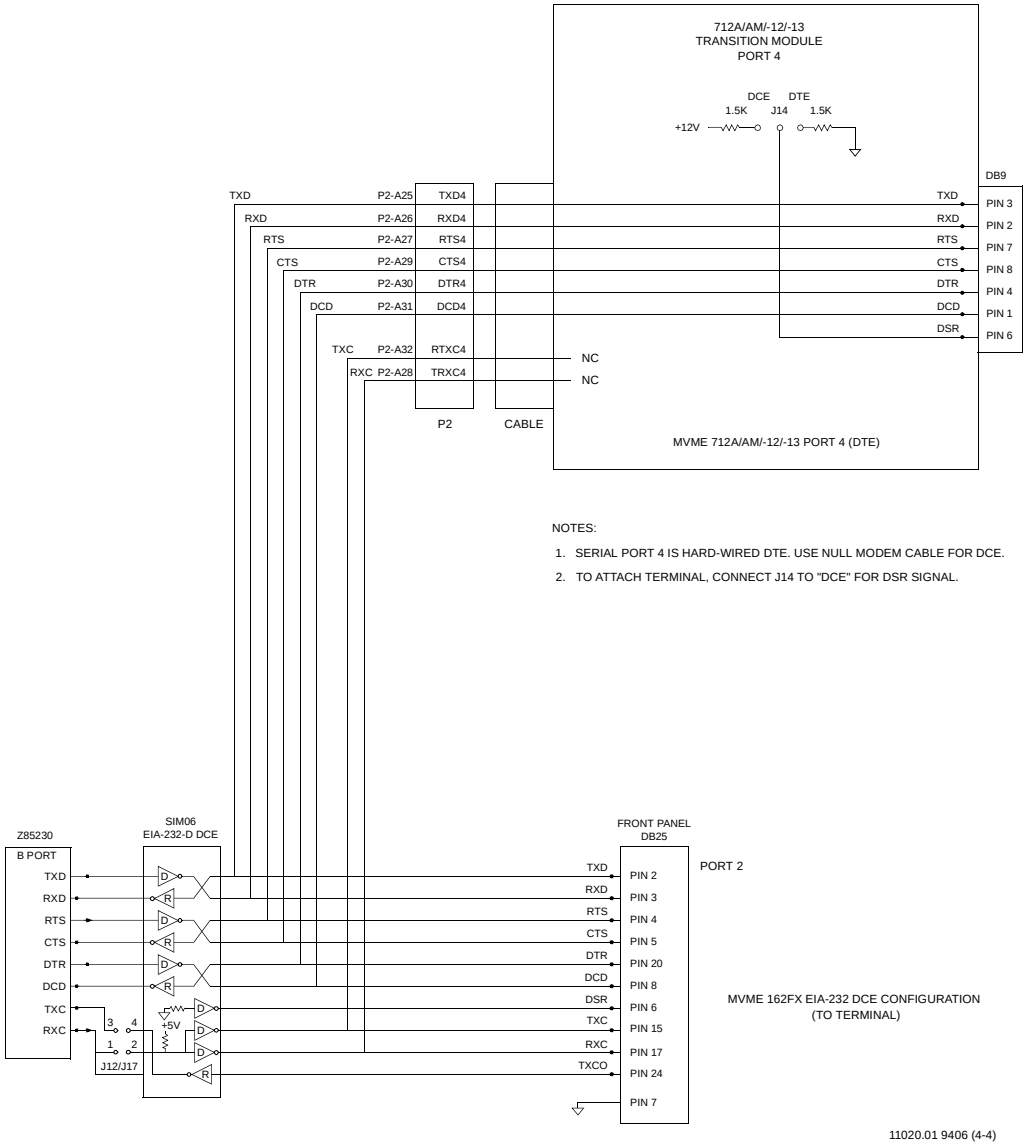


Figure 1-6. EIA-232-D Connections to MVME712A/AM/-12/-13 (Sheet 4 of 4)

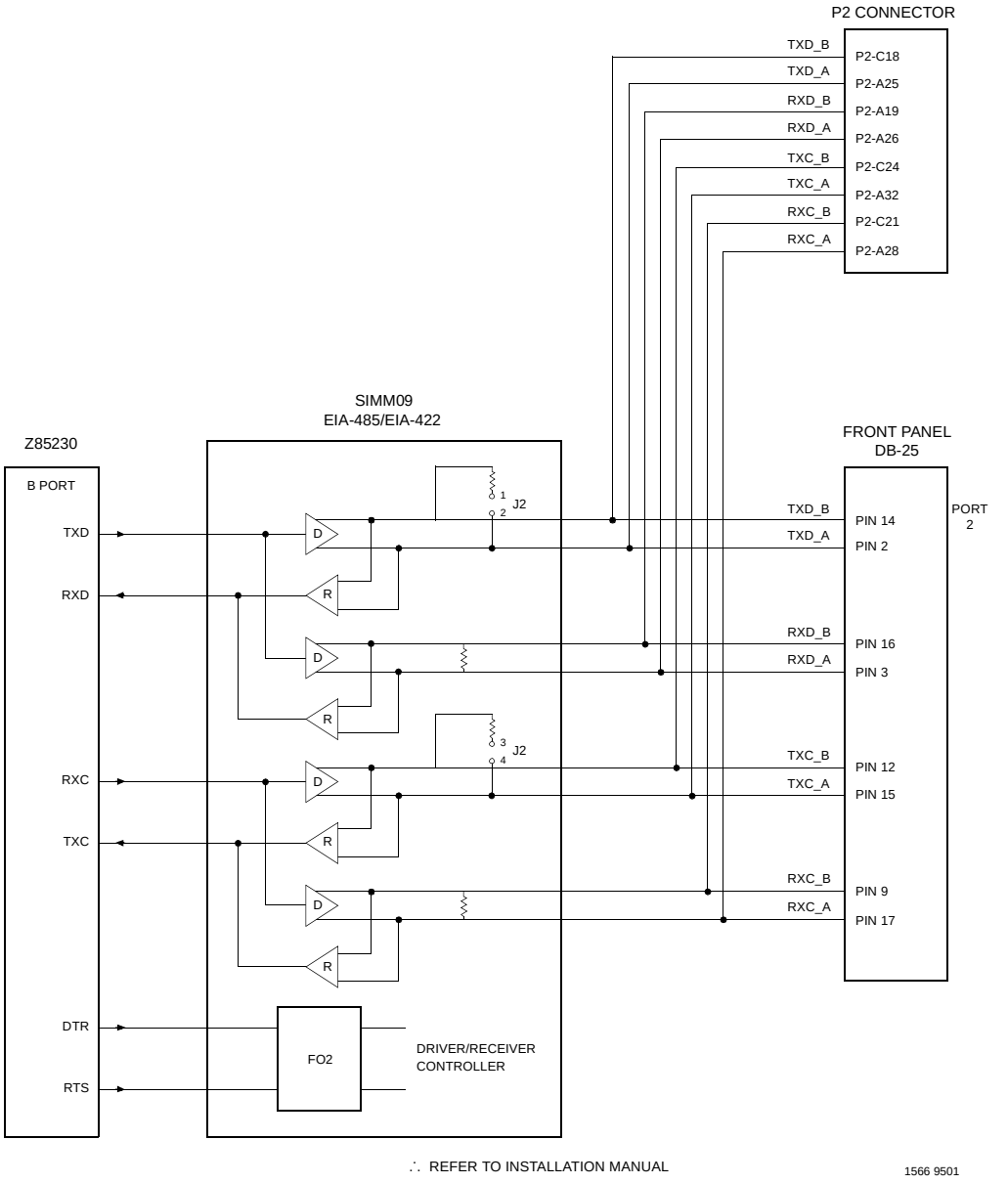


Figure 1-7. EIA-485/EIA-422 Connections

Startup and Operation 2

Introduction

This chapter provides information on powering up the MVME162FX 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 *MVME162FX Embedded Controller Programmer's Reference Guide*.

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 MVME162FX front panel.

Table 2-1. MVME162FX 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 MVME162FX 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 MVME162FX 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 MVME162FX 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. MVME162FX Front Panel Controls

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 MVME162FX's MPU, hardware, and firmware along with the rest of the system.

The Flash-resident firmware initializes the devices on the MVME162FX 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 MVME162FX hardware. The figure below charts the flow of the basic initialization sequence that takes place during system startup.

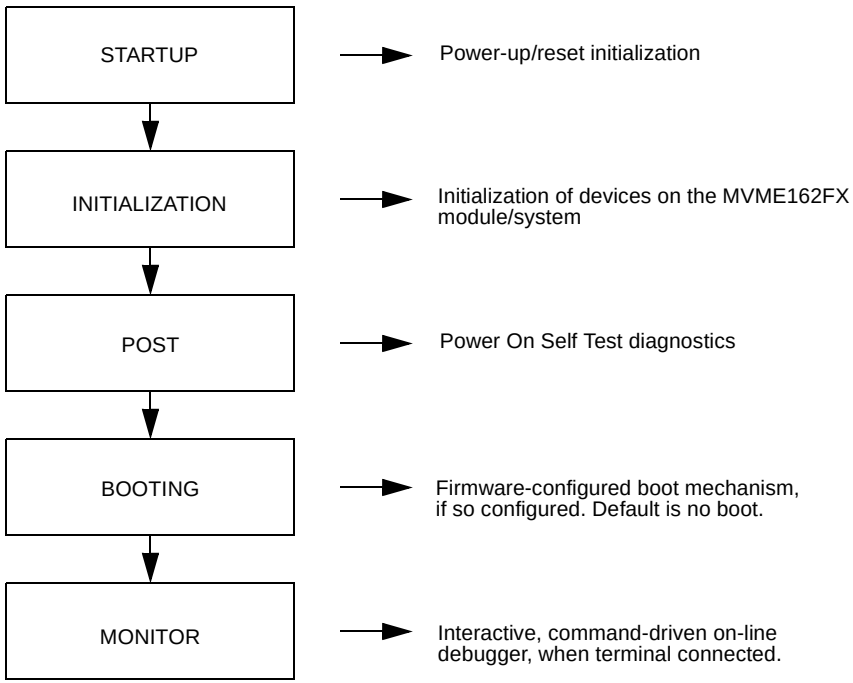


Figure 2-1. MVME162FX/Firmware System Startup

2

Pre-Startup Checklist

Before you power up the MVME162FX system, be sure that the following conditions exist:

1. Configuration jumpers on the MVME162FX VME Embedded Controller and associated equipment are set as required for your particular application.
2. The MVME162FX 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. (The console port is Serial Port 1 on the MVME162FX module, *or* Serial Port 2 on the MVME712*x* series transition 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

Follow the steps below to operate the 162Bug firmware with the MVME162FX controller. 162Bug is factory-installed in the Flash memory of the MVME162FX, except in the no-VMEbus case.

Caution

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

1. Turn all equipment power OFF. Refer to *Base Board Configurations* on page 1-5 and verify that the board is configured as necessary for your particular application.

   The general-purpose readable jumpers — header J22 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J28 (on 01-W3182F*xxx* series base boards; see Figure 1-2) — affect 162Bug operation as described below. The default configuration has all eight jumpers installed in positions GPI0 through GPI7. Models with no VMEbus interface have no jumper at GPI3 (pins 9-10).

   These readable jumpers can be read as a register at location $FFF4202D on the Memory Controller (MC2 chip) ASIC. The bit values are read as a 1 when the jumper is off, and as a 0 when the jumper is on. (For additional information on the MC2 chip, refer also to the *MVME162FX Embedded Controller Programmer's Reference Guide*.)

   With the factory-installed 162Bug firmware in place, four jumper positions are user-definable (GPI4, GPI5, GPI6, GPI7). The 162Bug firmware reserves/defines the four lower-order bits (GPI0 to GPI3). If the 162Bug firmware is removed, seven jumper positions are user-definable (GPI0-3, GPI4-7).

   Table 2-2 describes the bit assignments on J22/J28.

2. Configure the MVME162FX system controller function as required by installing or removing a jumper between pins 1-2 on header J1.

Table 2-2. Software-Readable Jumpers J22/J28

Bit No.	Function
Bit #0 (GPI0)	When set to 1 (high), instructs the debugger to use local static RAM for its work page (i.e., variables, stack, vector tables, etc.).
Bit #1 (GPI1)	When set to 1 (high), instructs the debugger to use the default setup/operation parameters in Flash or EPROM instead of the user setup/operation parameters in Non-Volatile RAM (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/EPROM defaults.
Bit #2 (GPI2)	Reserved for future use.
Bit #3 (GPI3)	When set to 0 (low), informs the debugger that it is executing out of Flash memory. When set to 1 (high), as set in non-VMEbus models, informs the debugger that it is executing out of the PROM.
Bit #4 (GPI4)	Open to your application.
Bit #5 (GPI5)	Open to your application.
Bit #6 (GPI6)	Open to your application.
Bit #7 (GPI7)	Open to your application.

3. You may configure Port B of the Z85230 serial communications controller with a serial interface module (SIM), which you install on the MVME162FX board at connector J10 (for 01-W3960B*xxx* series base boards) or J15 (for 01-W3182F*xxx* series base boards). 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.

4. Headers J11/J12 (for 01-W3960B*xxx* series base boards) or J16/J17 (for 01-W3182F*xxx* series base boards) configure serial ports 1 and 2 to drive or receive clock signals provided by the TXC and RXC signal lines. The MVME162FX is factory-configured for asynchronous communication: it comes with no jumpers on these headers. Refer to the instructions in Chapter 1 if your application requires configuring ports 1 and 2 for synchronous communication.

5. Header J24 (for 01-W3960B*xxx* series base boards) or J14 (for 01-W3182F*xxx* series base boards) 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 J24/J14 pins 1-2, denoting an 8MHz clock. Verify that this setting is appropriate for your application.

6. Header J25 (for 01-W3960B*xxx* series base boards) or J19 (for 01-W3182F*xxx* series base boards) enables or disables the IP bus strobe function on the MVME162FX. The factory configuration has no jumper on J25/J19; the Strobe∗ signal is disconnected from the IP2 chip. Verify that the strobe line need not be connected in your application.

7. Header J26 (for 01-W3960B*xxx* series base boards) or J2 (for 01-W3182F*xxx* series base boards) defines the state of the snoop control bus when an IP DMA controller is local bus master. The factory configuration has both jumpers installed for snoop inhibition. Verify that this setting is appropriate for your application.

8. On 01-W3182F*xxx* series base boards, header J24 defines the state of Flash memory write protection. The factory configuration has the jumper installed to permit writing to Flash. Verify that this setting is appropriate for your application.

   The 01-W3960B*xxx* series base boards have no jumper header for Flash write protection.

9. The EPROM size select header (J21 for 01-W3960B*xxx* series base boards, or J23 for 01-W3182F*xxx* series base boards) should be jumpered between pins 1-2. This sets it up for a 4Mbit x 8 EPROM density, the factory default.

10. Refer to the setup procedure for your particular chassis or system for details concerning the installation of the MVME162FX.

11. Connect the terminal to be used as the 162Bug system console to the default debug EIA-232-D port at Serial Port 1 on the front panel of the MVME162FX, *or* Serial Port 2 on the MVME712*x* series transition module. (For other connection options, refer to *Serial Connections* in Chapter 1.) Set the terminal up as follows:
    - Eight bits per character
    - One stop bit per character
    - Parity disabled (no parity)
    - Baud rate 9600 baud (power-up default for MVME162FX ports)

    After power-up, you can reconfigure the baud rate of the debug port by using the 162Bug Port Format (**PF**) command.

Note In order for high-baud-rate serial communication between 162Bug and the terminal to work, the terminal must do some form of handshaking. If the terminal you are using does not perform hardware handshaking via the CTS line, then it must do XON/XOFF handshaking. If you get garbled messages and missing characters, you should check the terminal to make sure XON/XOFF handshaking is enabled.

12. 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 3 above. After power-up, you can reconfigure the port(s) by programming the MVME162FX Z85230 Serial Communications Controller (SCC), or by using the 162Bug **PF** command.

13. 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 put 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.

14. Before using the MVME162FX 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 firmware, you may copy the EPROM version of 162Bug to Flash memory. To copy the EPROM version of 162Bug to Flash memory, first remove the jumper at J22/J28 pins 9-10, make sure that 162Bug is in Board mode, and copy the EPROM contents to Flash memory with the **PFLASH** command as follows:

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

Then reinstall the jumper at J22/J28 pins 9-10. (162Bug always executes from memory location FF800000; the state of J28 determines whether that location is in EPROM or Flash.)

Autoboot

Autoboot is a software routine that is contained in the 162Bug Flash/EPROM to provide an independent mechanism for booting an operating system. This 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 on 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 the software **ABORT** or **RESET** switches.

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 MVME162FX. 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 MVME162FX 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 MVME162FX 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.*

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 on 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 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 MVME162FX initiates a system reset. You can also initiate a reset by pressing and quickly releasing the **RESET** switch on the MVME162FX front panel, or reset the board in software.

For details on resetting the MVME162FX board through software, refer to the *MVME162FX 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 MVME162FX 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 MVME162FX 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 MVME162FX 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 MVME162FX. 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 MVME162FX 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 under *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 MVME162FX 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 MVME162FX 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 MVME162FX 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 28F008SA 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 MVME162FX 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 jumper header J22 (for 01-W3960B*xxx* series base boards) or J28 (for 01-W3182F*xxx* series base boards) 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 J22/J28. Refer to Step 14 (Note) under *Bringing up the Board* on page 2-5 for further details.

Memory Requirements

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

The 162Bug firmware executes from address $FF800000 whether in Flash or EPROM. With the GPI3 jumper at J22 (for 01-W3960B*xxx* series base boards) or J28 (for 01-W3182F*xxx* series base boards) installed, Flash memory appears at address $FF800000 and is the part executed at reset. The EPROM socket is mapped to address $FFA00000 in this configuration. This is the factory ship configuration *except* in the no-VMEbus case.

If you remove the GPI3 jumper at J22/J28, the address spaces of the Flash and EPROM are swapped. The EPROM contents then become the firmwware segment that executes at reset.

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

Table 3-1. Memory Offsets with 162Bug

Type of Memory Present	Default DRAM Base Address	Default SRAM Base Address
Single DRAM mezzanine	$00000000	$FFE00000 (onboard SRAM)

The DRAM is neither ECC nor parity type, but unprotected.

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 MVME162FX is reset, the target Program Counter (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 1.

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

For example, if a breakpoint has been specified, then control returns to the debugger when the breakpoint is encountered during execution of the user program. Alternatively, the user program could return to the debugger by means of the System Call Handler routine RETURN (described in the *Debugging Package for Motorola 68K CISC CPUs User's Manual*, Chapter 5).

A debugger command is made up of the following parts:

- The command name, either uppercase or lowercase (e.g., **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

Table 3-2. Debugger Commands (Continued)

Command	Description
DS	One Line Disassembler
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
NBO	Network Boot Operating System
NIOC	Network I/O Control
NIOP	Network I/O Physical

Table 3-2. Debugger Commands (Continued)

Command	Description
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
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 MVME162FX'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.* 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 MVME162FX board. The board structure for the MVME162FX is as follows:

```
162-Bug>cnfg
Board (PWA) Serial Number = "000000061050"
Board Identifier = "MVME162-513A "
Artwork (PWA) Identifier = "01-W3960B01A "
MPU Clock Speed = "3200"
Ethernet Address = 08003E200000
Local SCSI Identifier = "07"
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 *MVME162FX Embedded Controller Programmer's Reference Guide* 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 MVME162FX that affect these parameters appears in your *MVME162FX 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.

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.

Table 3-3. ENV Command Parameters (Continued)

ENV Parameter and Options	Default	Meaning of Default
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.
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.

Table 3-3. ENV Command Parameters (Continued)

ENV Parameter and Options	Default	Meaning of Default
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.
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 MVME162FX 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 MVME162FX. Default Memory Search Ending Address is the calculated size of local memory.

Table 3-3. ENV Command Parameters (Continued)

ENV Parameter and Options	Default	Meaning of Default
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.
Memory Search Delay Address	FFFFD20F	Default address is $FFFFD20F. This is the MVME162FX GCSR GPCSR0 as accessed through VMEbus A16 space; it assumes the MVME162FX GRPAD (group address) and BDAD (board address within group) switches are set to "on". This byte-wide value is initialized to $FF by MVME162FX hardware after a System or Power-On reset. In a multi-162FX environment, where the work pages of several Bugs reside in the memory of the primary (first) MVME162FX, 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.

Table 3-3. ENV Command Parameters (Continued)

ENV Parameter and Options	Default	Meaning of Default
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.
Size of Parity Memory	00100000	The size of the Parity type dynamic RAM mezzanine, if any. The default is the calculated size of the Dynamic memory mezzanine board.
Size of ECC Memory Board 0	00000000	The size of the first ECC type memory mezzanine. The default is the calculated size of the memory mezzanine.
Size of ECC Memory Board 1	00000000	The size of the second ECC type memory mezzanine. The default is the calculated size of the memory mezzanine.
Base Address of Static Memory	FFE00000	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.
Size of Static Memory	00080000	The size of the SRAM type memory present. The default is the calculated size of the onboard SRAM or an SRAM type mezzanine.

Table 3-3. ENV Command Parameters (Continued)

ENV Parameter and Options	Default	Meaning of Default
ENV 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 *MVME162FX Embedded Controller Programmer's Reference Guide* 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 MVME162FX. There are two slave address decoders set. They are set up as follows:		
Slave Enable #1 [Y/N]	Y	Yes, set up and enable Slave Address Decoder #1.
Slave Starting Address #1	00000000	Base address of the local resource that is accessible by the VMEbus. Default is the base of local memory, $0.
Slave Ending Address #1	000FFFFF	Ending address of the local resource that is accessible by the VMEbus. Default is the end of calculated memory.
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.

Table 3-3. ENV Command Parameters (Continued)

ENV Parameter and Options	Default	Meaning of Default
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).
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 Master Address Decoder #3. This is the default for boards containing at least 16MB of calculated RAM.

Table 3-3. ENV Command Parameters (Continued)

ENV Parameter and Options	Default	Meaning of Default
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.
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.

Table 3-3. ENV Command Parameters (Continued)

ENV Parameter and Options	Default	Meaning of Default
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.
ROM Access Time Code	03	Defines the ROM access time. The default is $03, which sets an access time of four clock cycles of the local bus.
Flash Access Time Code	02	Defines the Flash access time. The default is $02, which sets an access time of 3 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 MVME162FX 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 MVME162FXs.

The *MVME162FX 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.

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 (Dynamic RAM)
$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 MVME162FX VME embedded controller on a block diagram level. The *Summary of Features* provides an overview of the MVME162FX, 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 MVME162FX blocks, including programmable registers in the ASICs and peripheral chips, can be found in the *MVME162FX Embedded Controller Programmer's Reference Guide* (part number V162FXA/PG). Refer to that manual for a functional description of the MVME162FX in greater depth.

Summary of Features

The following table summarizes the features of the MVME162FX VME embedded controller.

Table 4-1. MVME162FX Features

Feature	Description
Microprocessor	MVME162FX: 25MHz or 32MHZ MC68040 or MC68LC040 processor
Form factor	6U VMEbus
Memory	4/8/16MB shared DRAM, unprotected
	512KB SRAM with battery backup
Flash memory	MVME162FX: One Intel 28F008SA 1MB x 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)
Switches	**RESET** and **ABORT** switches on front panel

Table 4-1. MVME162FX Features (Continued)

Feature	Description
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 MC2chip 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 MVME162FX is based on the MC68040/MC68LC040 microprocessor. Various versions of the boards are built with 4MB, 8MB, or 16MB of unprotected DRAM (dynamic RAM).

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 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 MVME162FX are routed to the VMEbus P2 connector. The main board is connected through a P2 adapter board and cables to the transition boards. The MVME162FX 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 MVME162FX 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 MVME162FX 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 MVME162FX:

- **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.

- **MC2 ASIC**. Supplies four tick timers and interfaces to the LAN chip, SCSI chip, serial port chip, BBRAM, and EPROM/Flash as well as the DRAM and/or SRAM mezzanine board.

- **IP2 ASIC**. Provides control and status information for up to four single-wide or two double-wide IndustryPack modules that can be plugged into the MVME162FX main board.

Block Diagram

The block diagram in Figure 4-1 on page 4-5 illustrates the MVME162FX's overall architecture.

Functional Description

This section contains a functional description of the major blocks on the MVME162FX.

Data Bus Structure

The local bus on the MVME162FX is a 32-bit synchronous bus that is based on the MC68040 or MC68060 bus, and which supports burst transfers and snooping. The various local bus master and slave devices use

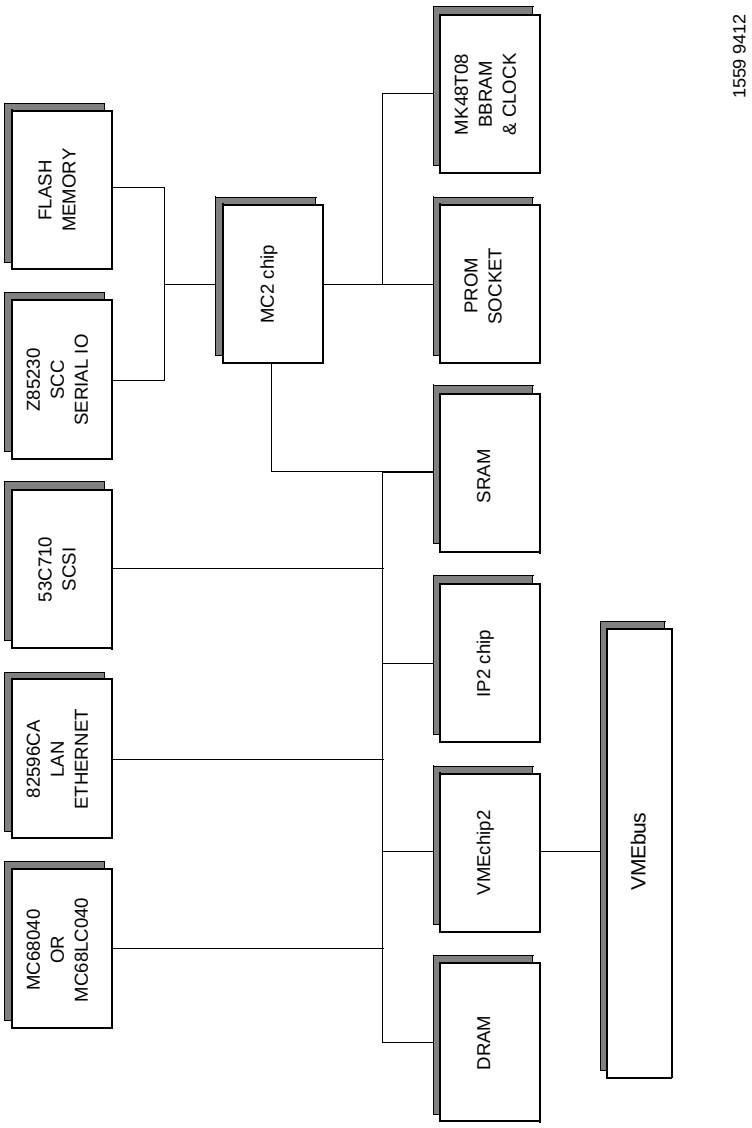


Figure 4-1. MVME162FX Block Diagram

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 slave; not all combinations pass the common sense test, however. Refer to the *MVME162FX 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

MVME162AFX models may be ordered with an MC68040 or MC68LC040 microprocessor.

The MC68040 has on-chip instruction and data caches and a floating-point processor. (The 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 MVME162FX 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 J26 (on 01-W3960B*xxx* series base boards; refer to Figure 1-1) or J2 (on 01-W3182F*xxx* series base boards; refer to Figure 1-2). J26/J2 determine 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 MVME162FX local bus slaves that support the snoop/caching mode are defined in the "Local Bus Memory Map" in the *MVME162FX Embedded Controller Programmer's Reference Guide.*

No-VMEbus-Interface Option

The MVME162FX board can be operated as an embedded controller without the VMEbus interface. On boards ordered with this option, the VMEchip2 ASIC and the VMEbus buffers are not populated. Also, the bus grant daisy chain and the interrupt acknowledge daisy chain have zero-ohm bypass resistors installed.

To support this feature, certain logic in the VMEchip2 has been duplicated in the MC2 chip. This logic is inhibited in the MC2 chip if the VMEchip2 is present. The enables for these functions are controlled by software and MC2 chip hardware initialization.

Note that MVME162FX models ordered without the VMEbus interface are shipped with a flash memory blank (the factory uses the VMEbus to program the flash memory with debugger code). To use the 162Bug package, MVME162Bug, be sure that jumper header J22 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J28 (on 01-W3182F*xxx* series base boards; see Figure 1-2) is configured for the EPROM memory map. Refer to Chapter 2 for further details. For ordering information, contact your local Motorola sales office.

Memory Options

The following memory options are available on the different versions of MVME162FX boards.

DRAM

The design of the MVME162FX board allows a 4MB, 8MB, or 16MB DRAM option. The DRAM architecture is non-interleaved for 4MB and 8MB, while the 16MB architecture is interleaved. The 4MB DRAM option is located entirely on the base board; the 8MB and 16MB options include 4MB or 12MB on a mezzanine module. The memory is not parity protected.

If the base board is populated with 4MB, the compatible mezzanines are those that start at four (i.e. the mezzanine address map starts at 4MB.) On 4MB base boards, therefore, the memory on the mezzanine appears contiguous to the memory on the base board.

Software can determine how much memory is on the base board by examining the seventh bit (bit 23) of the register at offset address $25. This bit is set upon MC2 chip initialization by a hardware state machine.

The following table defines the base board and memory mezzanine combinations with which the controller may be populated .

Mezzanine MB	Base Board MB	Available DRAM	Interleaved	MC2 chip Size (@ register offset $25)
0	4	4	N	100
4 (bank 2)	4	8	N	101
12 (banks 2, 3, and 4)	4	16	Y	111

For specifics on DRAM performance and for detailed programming information, refer to the MC2 chip programming model in the *MVME162FX Embedded Controller Programmer's Reference Guide.*

SRAM

The MVME162FX 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 chip programming model in the *MVME162FX Embedded Controller Programmer's Reference Guide.* 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 an Electro Marketing EM1275 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 MVME162FX provides a jumper header, J20 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J22 (on 01-W3182F*xxx* series base boards; see Figure 1-2) that allows either power source of the EM1275 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 installed on the MVME162FX. The SRAM may malfunction if inputs to the EM1275 are left unconnected.

The SRAM is controlled by the MC2 chip, and the access time is programmable. Refer to the MC2 chip description in the *MVME162FX Embedded Controller Programmer's Reference Guide* for more detail.

About the Batteries

The power source for the onboard SRAM is an Electro Marketing EM1275 device (or equivalent, such as RAYOVAC FB1225) with two BR1225-type lithium cells. The battery is socketed for easy removal and replacement. Small capacitors are provided so that the batteries 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 batteries supplied on the MVME162FX 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 *Debugger Commands* in Chapter 3) 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 MVME162FX is shipped with the batteries disconnected (i.e., with VMEbus +5V standby voltage selected as both primary and secondary power source). If you intend to use the battery as a power source, whether primary or secondary, it is necessary to reconfigure the jumpers on J20/J22 before installing the board. Refer to *SRAM Backup Power Source* on page 1-13 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 module, carefully pull the battery from the 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 MVME162FX implementation includes 1MB or 2MB Flash memory. Flash memory is a single Intel device (28F008SA on the MVME162FX) organized in a 1MB x 8 configuration. For information on programming Flash, refer to the Intel documents listed under *Manufacturer's Documents* in the *Related Documentation* appendix.

The Flash write enable signal is controlled by:

- A bit in the Flash Access Time Control register in the MC2 chip ASIC
- On 01-W3182F*xxx* series base boards only (see Figure 1-2), a board-level configuration jumper (J24) which determines the status of Flash write protection on the board

Refer to the *MVME162FX Embedded Controller Programmer's Reference Guide* 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 jumper (line GPI3, pins 9-10 on J22/J28) allows reset code to be fetched either from Flash memory (jumper installed) or from EPROMs (jumper removed).

Note that MVME162FX 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 jumper header J22/J28 is configured 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 MVME162FX. 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 MC2 chip supports 8-, 16-, and 32-bit accesses to the M48T58. Refer to the description of the MC2 function in the *MVME162FX 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 *MVME162FX Embedded Controller Programmer's Reference Guide* 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 MC2 chip ASIC at location $FFF42043. The GPI inputs are integrated into the MC2 chip ASIC at location $FFF4202C, bits 23-16.

I/O Interfaces

The MVME162FX 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 MVME162FX 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 supports synchronous (SDLC/HDLC) and asynchronous protocols. The MVME162FX 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 MC2 function (the MC2 chip can handle up to four Z85230 chips). Refer to the Z85230 data sheet listed listed under *Manufacturer's Documents* in the *Related Documentation* appendix, and to the MC2 programming model in the *MVME162FX Embedded Controller Programmer's Reference Guide*, for information.

MVME162FX 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:

- The DB25 connector marked **SERIAL PORT 1/CONSOLE** on the front panel of the MVME162FX. **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 J11 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J16 (on 01-W3182F*xxx* series base boards; see Figure 1-2) on the MVME162FX board. (The TxC and RxC clock lines are not available on the MVME712*x* transition modules.)

- 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 J27 on the transition module.

  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 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 1-4 (sheets 1 and 2) in Chapter 1 illustrates the two configurations available for Port A when the MVME162FX is used with an MVME712M. Figure 1-6 (sheets 1 and 2) shows the two configurations available for Port A when the MVME162FX is used with an MVME712A/AM/-12/-13.

MVME162FX Serial Port 2

The configuration of port B on the Z85230 is established via a Serial Interface Module (SIM) which is installed at connector J10 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J15 (on 01-W3182F*xxx* series base boards; see Figure 1-2) on the MVME162FX 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:

- The DB25 connector marked **SERIAL PORT 2** on the front panel of the MVME162FX. **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 J12 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J17 (on 01-W3182F*xxx* series base boards; see Figure 1-2) on the MVME162FX 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.)

- 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 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 1-4 (sheets 3 through 6) in Chapter 1 illustrates the four configurations available for Port B when the MVME162FX is used with an MVME712M. Note that the port configurations shown in Figure 1-4 sheets 5 and 6 are not recommended for synchronous applications because of the incorrect clock direction. Figure 1-5 (sheets 1 and 2) shows an MVME162FX with the two configurations available with EIA-530 SIMs. Figure 1-6 (sheets 3 and 4) shows the two configurations available for Port B when the MVME162FX is used with an MVME712A/AM/-12/-13. Figure 1-7 shows an MVME162FX 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 MVME162FX 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 MVME162FX circuits. In the EIA-530 case, attempting to use these ports would produce certain connections with the potential to damage the MVME162FX or the peripherals.

Caution When using an EIA-530 SIM or an EIA-485/EIA-422 SIM, do not connect the MVME162FX to an MVME712*x* board. Neither the P2 adapter nor the transition boards support the EIA-530, EIA-485, or EIA-422 signals.

IndustryPack (IP) Interfaces

The IP2 chip ASIC on the MVME162FX 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), 25 or 32MHz (32MHz for MC68LC040 or 25MHz for MC68040) 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 *MVME162FX Embedded Controller Programmer's Reference Guide* for details of the IP interface. Refer to Chapter 5 for the pin assignments of the IP connectors.

Notes MVME162FX 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 MVME162FX board can *not* use the Strobe∗ signal unless the jumper is removed from J25 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J19 (on 01-W3182F*xxx* series base boards; see Figure 1-2). This will disconnect the Strobe∗ output from the IP2 ASIC.

Ethernet Interface

The MVME162FX 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 MVME162FX 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 (i.e., every MVME162FX 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 MVME162FX 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 MC2 chip ASIC. Refer to the 82596CA user's guide and to the description of the MC2 function in the *MVME162FX Embedded Controller Programmer's Reference Guide* for detailed programming information.

SCSI Interface

The MVME162FX 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 MC2 chip ASIC. Refer to the NCR 53C710 user's guide and to the description of the MC2 function in the *MVME162FX 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 MVME162FX, 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 MVME162FX 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 MC2 chip. The tick timers may be programmed to generate periodic interrupts to the processor. Refer to the VMEchip2 and MC2 chip descriptions in the *MVME162FX Embedded Controller Programmer's Reference Guide* for detailed programming information.

Watchdog Timer

A watchdog timer function is provided in both the 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 MC2 chip descriptions in the *MVME162FX Embedded Controller Programmer's Reference Guide* for detailed programming information.

The watchdog timer logic is duplicated in the VMEchip2 and MC2 chip ASICs. Because the watchdog timer function in the VMEchip2 is a superset of that function in the MC2 chip (system reset function), the timer

in the VMEchip2 is to be used in all cases except for versions of the MVME162FX which do not include the VMEbus interface (i.e., 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 *MVME162FX Embedded Controller Programmer's Reference Guide* for detailed programming information.

Local Bus Timeout

The MVME162FX provides timeout functions in the VMEchip2 and the 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 MC2 chip descriptions in the *MVME162FX Embedded Controller Programmer's Reference Guide* for detailed programming information.

The access timer logic is duplicated in the VMEchip2 and MC2 ASICs. Because the local bus timer in the VMEchip2 can detect an offboard access and the MC2 chip's local bus timer cannot, the timer in the VMEchip2 ASIC is used in all cases except for versions of the MVME162FX which do not include the VMEbus interface (i.e., 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
MC68040/MC68LC040	4	Lowest

Connectors

The MVME162FX 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 MVME162FX are also connected to two 25-pin DB-25 female connectors on the front panel, J9/J18 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J15/J25 (on 01-W3182F*xxx* series base boards; see Figure 1-2). The four IP modules connect to the MVME162FX by four pairs of 50-pin connectors. Four 50-pin connectors behind the front panel are for external connections to IP signals. The memory mezzanine board is plugged into two 40-pin connectors.

Pin assignments for the connectors on the MVME162FX are listed in Chapter 5.

Remote Status and Control

The remote status and control connector, J4 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J6 (on 01-W3182F*xxx* series base boards; see Figure 1-2) is a 20-pin connector located behind the front panel of the MVME162FX. 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 MVME162FX.

Pin Assignments

5

Connector Pin Assignments

This chapter summarizes the pin assignments for the following groups of interconnect signals on the MVME162FX:

Connector	Reference Designators		Table
	01-W3960	01-W3182	
IndustryPack A, B, C, D connectors	J2/3/6, J5/7/8, J13/14/17, J16/18/19	J4/5/8, J7/10/11, J20/21/27, J26/29/30	Table 5-1
Remote Reset connector	J4	J6	Table 5-2
Memory Mezzanine connector 1	P3	J12	Table 5-3
Memory Mezzanine connector 2	P4	J13	Table 5-4
Serial interface module connector	J10	J15	Table 5-5
Serial Port 2 connector	J9	J18	Table 5-6
Serial Port 1 connector	J15	J25	Table 5-7
VMEbus connector P1	P1	P1	Table 5-8
VMEbus connector P2	P2	P2	Table 5-9

The tables in this chapter furnish pin assignments only. For detailed descriptions of the interconnect signals, consult the support information for the MVME162FX (available through your Motorola sales office).

IndustryPack A, B, C, D Connectors

Up to four IndustryPack (IP) modules may be installed on the MVME162FX board. For each IP module, there are two 50-pin plug connectors on the board. For external cabling to the IP modules, four 50-pin IDC connectors (one for each module) are provided behind the MVME162FX front panel. The pin assignments are the same for both types of connector. The following table relates IP modules A through D to their respective connectors for both base board versions.

IP Module	Cable Connector		Board Connectors	
	01-W3960	01-W3182	01-W3960	01-W3182
A	J6	J8	J2/3	J4/5
B	J5	J7	J7/8	J10/11
C	J17	J27	J13/14	J20/21
D	J16	J26	J18/19	J29/30

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

The MVME162FX has a 20-pin remote reset connector, J4 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J6 (on 01-W3182F*xxx* series base boards; see Figure 1-2). This connector is mounted behind the front panel. When the MVME162FX board is enclosed in a chassis and the front panel is not visible, the remote reset 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 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

Memory Mezzanine Connector 1

Memory mezzanine connector 1, P3 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J12 (on 01-W3182F*xxx* series base boards; see Figure 1-2) is a standard double-row 40-pin socket connector mounted on the MVME162FX Embedded Controller PWB. It connects to a corresponding 40-pin plug connector on the ECC DRAM mezzanine board and (together with memory mezzanine connector 2) carries the DRAM address, data, and control signals to and from the mezzanine board. The following table lists the pin assignments for memory mezzanine connector 1.

Table 5-3. Mezzanine Connector 1 Pin Assignments

1	LD0	LD1	2
3	LD2	LD3	4
5	LD4	LD5	6
7	LD6	LD7	8
9	GND	GND	10
11	LD8	LD9	12
13	LD10	LD11	14
15	LD12	LD13	16
17	LD14	LD15	18
19	GND	GND	20
21	LD16	LD17	22
23	LD18	LD19	24
25	LD20	LD21	26
27	LD22	LD23	28
29	GND	GND	30
31	LD24	LD25	32
33	LD26	LD27	34
35	LD28	LD29	36
37	LD30	LD31	38
39	+5V	+5V	40

Memory Mezzanine Connector 2

Memory mezzanine connector 2, P4 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J13 (on 01-W3182F*xxx* series base boards; see Figure 1-2) is a standard double-row 40-pin socket connector mounted on the MVME162FX Embedded Controller PWB. It connects to a corresponding 40-pin plug connector on the ECC DRAM mezzanine board and (together with memory mezzanine connector 1) carries the DRAM address, data, and control signals to and from the mezzanine board. The following table lists the pin assignments for memory mezzanine connector 2.

Table 5-4. Mezzanine Connector 2 Pin Assignments

1	+5V	+5V	2
3	MEZ_ADR0	MEZ_ADR1	4
5	MEZ_ADR2	MEZ_ADR3	6
7	MEZ_ADR4	MEZ_ADR5	8
9	GND	GND	10
11	MEZ_ADR6	MEZ_ADR7	12
13	MEZ_ADR8	MEZ_ADR9	14
15	MEZ_OE2*	MEZ_OE1*	16
17	MEZ_OE0*	MEZ_RAS*	18
19	GND	GND	20
21	MEZ_CAS0*	MEZ_CAS1*	22
23	MEZ_CAS2*	MEZ_CAS3*	24
25	MEZ_WR3*	MEZ_WR2*	26
27	MEZ_WR1*	MEZ_WR0*	28
29	GND	GND	30
31	DRAMPD0	DRAMPD1	32
33	DRAMPD2	DRAMPD3	34
35	MEZ_OE3*	MEZ0	36
37	MEZ1	MEZ2	38
39	GND	GND	40

Serial Interface Module Connector

Port B of the Z85230 serial communications controller on the MVME162FX board is configurable via serial interface modules (SIMs) that are installed at connector J10 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J15 (on 01-W3182F*xxx* series base boards; see Figure 1-2). The serial interface modules currently available are listed and described in Chapter 2. The pin assignments for J10/J15 are listed in the following table.

Table 5-5. Serial Port Module Connector Pin Assignments

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

A DB25 socket connector located on the front panel of the MVME162FX, J9 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J18 (on 01-W3182F*xxx* series base boards; see Figure 1-2) provides the interface to Serial Port 2. The pin assignments for the connector are as follows.

Table 5-6. Serial Port 2 Connector Pin Assignments

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

A DB25 socket connector located on the front panel of the MVME162FX, J15 (on 01-W3960B*xxx* series base boards; see Figure 1-1) or J25 (on 01-W3182F*xxx* series base boards; see Figure 1-2) provides the interface to Serial Port 1. The pin assignments for the connector are as follows.

Table 5-7. Serial Port 1 Connector 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

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 MVME162FX implementation. The pin assignments for P1and P2 are listed in Table 5-8 and Table 5-9 respectively.

Table 5-8. VMEbus Connector P1 Pin Assignments

	Row A	Row B	Row C	
1	VD0	VBBSY*	VD8	1
2	VD1	VBCLR*	VD9	2
3	VD2	VACFAIL*	VD10	3
4	VD3	VBGIN0*	VD11	4
5	VD4	VBGOUT0*	VD12	5
6	VD5	VBGIN1*	VD13	6
7	VD6	VBGOUT1*	VD14	7
8	VD7	VBGIN2*	VD15	8
9	GND	VBGOUT2*	GND	9
10	VSYSCLK	VBGIN3*	VSYSFAIL*	10
11	GND	VBGOUT3*	VBERR*	11
12	VDS1*	VBR0*	VSYSRESET*	12
13	VDS0*	VBR1*	VLWORD*	13
14	VWRITE*	VBR2*	VAM5	14
15	GND	VBR3*	VA23	15
16	VDTACK*	VAM0	VA22	16
17	GND	VAM1	VA21	17
18	VAS*	VAM2	VA20	18
19	GND	VAM3	VA19	19
20	VIACK*	GND	VA18	20
21	VIACKIN*	Not Used	VA17	21
22	VIACKOUT*	Not Used	VA16	22
23	VAM4	GND	VA15	23
24	VA7	VIRQ7*	VA14	24
25	VA6	VIRQ6*	VA13	25
26	VA5	VIRQ5*	VA12	26
27	VA4	VIRQ4*	VA11	27
28	VA3	VIRQ3*	VA10	28
29	VA2	VIRQ2*	VA9	29
30	VA1	VIRQ1*	VA8	30
31	–12V	P5VSTDBY	+12V	31
32	+5V	+5V	+5V	32

Table 5-9. 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
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

Board Specifications

The following table lists the general specifications for the MVME162FX VME embedded controller. The subsequent sections detail cooling requirements and EMC regulatory compliance.

A complete functional description of the MVME162FX boards appears in Chapter 4. Specifications for the optional IndustryPack modules can be found in the documentation for those modules.

Table A-1. MVME162FX Specifications

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

Cooling Requirements

The Motorola MVME162FX 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 MVME162FX may subject it to high temperatures.

The MVME162FX 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 MVME162FX. 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 MVME162FX (MVME162-*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 MVME162FX 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 MVME162FX 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 MVME162FX 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 MVME162FX.
3. Use low-power IP modules only. The preferred locations for IP modules are position *A* and position *D*.

EMC Regulatory Compliance

The MVME162FX 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 MVME162FX 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.

Solving Startup Problems

In the event of difficulty with your MVME162FX 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 MVME162FX 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 MVME162FX Boards

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 may be configured incorrectly.	Check the board jumpers 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. Remove the jumper from J21, pins 9-10. 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.	

Table B-1. Troubleshooting MVME162FX Boards

Condition	Possible Problem	Try This:
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.
	B. There may be some fault in the board hardware.	**Caution** Performing the next step (**env;d**) will change some parameters that may affect your system's operation. (continues>)

Table B-1. Troubleshooting MVME162FX Boards

Condition	Possible Problem	Try This:
IV. *Continued*		2. At the command line prompt, type in: **env;d <CR>** This sets up the default parameters for the debugger environment. 3. When prompted to Update Non-Volatile RAM, type in: **y <CR>** 4. When prompted to Reset Local System, type in: **y <CR>** 5. After 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.
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.

Table B-1. Troubleshooting MVME162FX Boards

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

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

C

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 MVME162FX with an SCSI port is used, the MVME162FX module has CLUN 0.
2. For MVME162FXs, 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.

D

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 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
3	$FFFFA700	80	Local floppy drive
		81	Local floppy drive

MVME328 -- 14 Devices

Controller LUN	Address	Device LUN	Device Type
6	$FFFF9000	00 08 10 18 20 28 30	SCSI Common Command Set (CCS), which may be any of these: - Removable flexible direct access (TEAC style) - CD-ROM - Sequential access
7	$FFFF9800		
16	$FFFF4800	40 48 50 58 60 68 70	Same as above, but these will only be available if the daughter card for the second SCSI channel is present.
17	$FFFF5800		
18	$FFFF7000		
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 MVME162FX.

IOT Parameter	Floppy Types and Formats						
	DSDD5	**PCXT8**	**PCXT9**	**PCXT9_3**	**PCAT**	**PS2**	**SHD**
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
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 Computer Group publications listed below are referenced in this manual. You can obtain paper or electronic copies of MCG publications by:

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

Table E-1. Motorola Computer Group Documents

Document Title	Motorola Publication Number
MVME162FX Embedded Controller Programmer's Reference Guide	V162FXA/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 locate and view the most up-to-date product information in PDF or HTML format, visit http://www.mcg.mot.com/literature.

Manufacturers' Documents

For additional information, refer to the following table for manufacturers' data sheets or user's manuals. As a further help, sources for the listed documents are also provided. Please note that in many cases, the information is preliminary and the revision levels of the documents are subject to change without notice.

Table E-2. Manufacturers' Documents

Document Title and Source	Publication Number
M68000 Family Reference Manual MC68040 Microprocessor User's Manual 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	M68000FR M68040UM
82596CA Local Area Network Coprocessor Data Sheet 82596CA Local Area Network Coprocessor User's Manual 28F008SA Flash Memory Data Sheet Intel Corporation Web: http://developer.intel.com/design	290218 296853 290429
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

Table E-2. Manufacturers' Documents (Continued)

Document Title and Source	Publication Number
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 manufacturers' data sheets or user's manuals. As a further help, sources for the listed documents are also provided. Please note that in many cases, the information is preliminary and the revision levels of the documents are 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

Table E-3. Related Specifications (Continued)

Document Title and Source	Publication Number
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
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

Numerics

162Bug firmware 4-7
- command line input 3-5
- commands 3-6
- default location 2-5
- disk/tape controller data D-1
- implementation 3-3
- memory requirements 3-3
- modifying NVRAM 3-9
- network controller data C-1
- overview 3-1
- stack space 3-4

27C040 EPROM 3-3
53C710 SCSI controller 4-18
82596CA LAN coprocessor 4-17

A

Abort process 2-14
ABORT switch, remote mounting 4-22
aborting program execution 2-1
address/data configurations 1-22
addressing modes 1-22
altitude (operating) A-1
ambient air temperature (effect on cooling) A-2
arbitration priority 4-21
arguments, firmware command 3-6
autoboot process 2-9
autojumpering (VME backplane) 1-21

B

backplane connectors P1 and P2 1-22
backplane jumpers 1-21
batteries 4-9
baud rate, default 2-8
BBRAM (battery-backed-up RAM) and clock 3-9, 4-12
BG (bus grant) signal 1-21
block diagram 4-4
board
- architecture 4-1
- configuration 1-4
- connectors 4-21
- dimensions A-1
- features 4-1
- installation 1-20
- preparation 1-1
- storage, preparation for 4-10

booting the system 2-9, 2-11, 2-12
BREAK key 2-14
Break process 2-14
bus grant (BG) signal 1-21

C

C programming language 3-3
cable connections 1-22
cables, I/O ports A-4
cache memory 4-6
chassis grounding A-4
CISC embedded controller(s) D-1
Clear To Send (CTS) signal 2-8
CLUN (controller LUN) C-1, D-1
command arguments, firmware 3-6
command identifier, firmware 3-5
command line input 3-5

commands, firmware
CNFG (Configure Board Information Block) 3-9
ENV (Set Bug/Operating System Environment) 3-11
conductive chassis rails (EMC compliance) A-4
configurable items, MVME162FX board 1-5
configuration, hardware 2-5
Configure Board Information Block (CNFG) firmware command 3-9
configuring
162Bug parameters 3-11
IndustryPack modules 3-19
IP base addresses 3-19
VMEbus interface 3-16
connection diagrams
EIA-232-D 1-25, 1-33
EIA-485/EIA-422 1-37
EIA-530 1-31
connector pin assignments 5-1
connectors 4-21
IndustryPack (IP) 1-19
console port 2-8
control/status registers 1-23
controller LUN (CLUN) C-1, D-1
controller modules (disk/tape) D-1
controller modules, network C-1
conventions used in the manual 4
cooling requirements A-2
CSR bit IP32 (IP bus clock) 1-16
CTS (Clear To Send) signal 2-8

D

data bus structure, MVME162FX 4-4
data circuit-terminating equipment (DCE) 4-13
data sheets E-2
date and time, setting 2-9, B-3
DCE (data circuit-terminating equipment) 4-13
debugger
commands 3-6
firmware (162Bug) 3-9
prompts 3-5
default baud rate 2-8
device LUN (DLUN) C-1, D-2
diagnostic facilities 2-14
dimensions, base board A-1
direct access devices D-4
direct memory access (DMA) 4-17
directories, switching 2-14
disk/tape controller modules supported D-1
DLUN (device LUN) C-1, D-2
DMA (direct memory access) 4-17
DRAM (dynamic RAM) 4-7
base address 1-22
options 4-7

E

EIA-232-D
connection diagrams 1-25, 1-33
ports 2-8
SIM part numbers 1-10
EIA-485/EIA-422 connection diagrams 1-37
EIA-530 connection diagrams 1-31
EIA-530/V.36 SIM part numbers 1-10
EMC regulatory compliance A-4
ENV command parameters 3-11
environmental parameters 3-9
EPROM
and Flash memory 4-11
contents, transferring 2-9
device 3-3
size select header (J21/J23) 1-14
socket 4-11
ESD (electrostatic discharge), precautions against 1-3
ESDI Winchester hard drive D-3
Ethernet
controller modules supported C-1
interface 4-17
station address 4-18

extended addressing 1-22

F

features, hardware 4-1
firmware
  directories 3-5
  documentation E-1
  initialization 2-2
Flash memory 3-3, 4-11
  write protection (J24) 1-13
flexible diskettes (controller data) D-2
floppy diskettes D-4
floppy drives (disk/tape controller) D-2, D-3
forced air cooling A-2
front panel switches and indicators 2-1
functional description 4-1
fuses, MVME162FX 1-23

G

GCSR (global control/status registers) 1-23
general-purpose readable jumpers header (J22/J28) 1-15
global
  bus timeout 1-23
  control/status registers (GCSR) 1-23
grounding A-4
grounding strap, use of 1-3

H

handshaking 2-8
hard disk drive D-3
hardware
  diagnostics 2-14
  features 4-1
  initialization 2-3
  interrupts 4-20
  preparation 1-1
headers, setting 2-5
high-temperature operation A-3
humidity, relative A-1

I

I/O interfaces 4-13
IACK (interrupt acknowledge) signal 1-21
indicators 2-1
IndustryPack (IP)
  interface functions 4-17
  modules, installation 1-19
  signals 5-2
IndustryPack modules
  base addresses of 3-19
  configuration 3-20
  configuring 3-19
  general control register 3-20
  memory size 3-20
initial conditions 2-4
installation
  considerations in 1-23
  instructions for 1-19
  SIMs 1-11
  transition modules 1-21
interconnect signals 5-1
interrupt acknowledge signal (IACK) 1-21
interrupt control registers 3-20
interrupt functions 4-13
Interrupt Stack Pointer (ISP) 3-5
interrupts, hardware 4-20
IOT command parameters D-5
IP (IndustryPack)
  bus clock header (J24/J14) 1-16
  bus strobe select header (J25/J19) 1-17, 2-7
  reset signal 4-17
  snoop control header (J26/J2) 2-7
IP2 interface function 4-17
IP32 CSR bit (IP bus clock) 1-16
ISP (Interrupt Stack Pointer) 3-5

J

J1 jumper 1-8
J10/J15 connector (SIM selection) 1-9
J11/J16 jumpers 1-11
J12/J17 jumpers 1-12

INDEX

J20/J22 jumper 1-13
J21/J23 jumper 1-14
J22/J28 jumpers 1-15
J24 jumper 1-13
J24/J14 jumpers 1-16
J25/J19 jumper 1-17
J26/J2 jumpers 1-18
J4/J6 connector 4-21
J9/J18 serial connector 4-21
jumper headers
  J1 (system controller selection) 1-8
  J11/J16 (serial port 1 clock) 1-11, 2-7
  J12/J17 (serial port 2 clock) 1-12, 2-7
  J20/J22 (SRAM backup power) 1-13
  J21/J23 (EPROM size selection) 1-14, 2-7
  J22/J28 (general-purpose readable jumpers) 3-3, 3-4, 4-7
  J24 (Flash memory write protection) 1-13, 2-7
  J24/J14 (IP bus clock) 1-16, 2-7
  J25/J19 (IP bus strobe) 1-17, 2-7
  J26/J2 (IP DMA snoop control) 1-18, 2-7
  location of 1-5
jumpers
  backplane 1-21
  setting 2-5
  user-definable 1-15, 2-5

L

LAN
  controller modules supported C-1
  interface 4-17
LEDs (light-emitting diodes) 2-1
  remote mounting 4-21
local
  bus 4-20
  bus arbiter 4-21
  bus timeout 4-20
  resources for the processor 4-19
Local Area Network (LAN) interface 4-17
location monitors, processor 1-23
logical unit number (LUN) (see CLUN or DLUN)
LUN (logical unit number) (see CLUN or DLUN)

M

manual conventions 4
MC2 chip
  local control/status registers 1-15
  SRAM and 4-9
MC68040 or MC68LC040 MPU 4-6
MC68xx040 cache memory 4-6
memory options 4-7
memory requirements, firmware 3-3
mezzanine signals 5-5, 5-6
microprocessor options 4-6
modem on transition boards 4-14
MPU options 4-6
MVME162Bug firmware 4-7
  documentation E-1
MVME162FX
  as network controller C-1
  block diagram 4-4
  board specifications A-1
  cooling requirements A-2
  EMC regulatory compliance A-4
  features 4-1
  fuses 1-23
  I/O interfaces 4-13
  installation 1-20
MVME320 disk/tape controllers D-2
MVME323 disk/tape controller D-3
MVME327A disk/tape controller D-3
MVME328 disk/tape controller D-4
MVME350 controller D-4
MVME374 network controller C-1
MVME376 network controller C-1
MVME712M
  installation 1-21
MVME712x series transition modules
  serial ports 4-16

INDEX

N

network boot process 2-12
network controller modules C-1
non-volatile RAM (NVRAM) 3-9, 3-11
no-VMEbus-interface option 1-15, 4-12
    firmware and 4-7
    timer functions 4-20

O

operating parameters 3-9
operating temperature A-1
option fields, in firmware command 3-6

P

P1 and P2 connectors 4-21, 5-9
parameters, ENV command 3-11
part numbers, SIM 1-10
peripherals, connecting 1-22
pin assignments, connector 5-1
port number(s), firmware 3-5
power requirements A-1
powering up the board 2-1
power-save mode 4-10
preparing the board 1-1
processor bus structure 4-4
processor cache memory 4-6
processor location monitors 1-23
programmable tick timers 4-19

Q

QIC-02 streaming tape drive D-4

R

readable jumpers 1-15
regulatory compliance A-4
regulatory guidelines A-4
related specifications E-3
relative humidity A-1
remote control/status connector (J4/J6) 4-21, 5-4
RESET switch, remote mounting 4-22
resetting the system 2-1, 2-13
RF emissions, minimizing 1-21, A-4
ROMboot process 2-11

S

safety procedures 1-4
SCC (serial communications controller) 2-8
SCSI
    common command set (CCS) devices D-2, D-4
    controller (53C710) 4-18
    direct access devices D-2
    interface 4-18
    sequential access devices D-2
    termination 4-19
    terminator configuration 4-19
    terminator power 1-24
SD command 2-14
sequential access devices D-4
serial
    communication parameters 2-4
    communications controller (SCC) 2-8
    communications interface 1-24, 4-13
    communications interface (port 1 signals) 5-9
    communications interface (port 2 signals) 5-8
    connectors 4-21
    interfaces and transition boards 4-17
    ports 2-8
serial interface module (SIM)
    considerations for transition modules 4-17
    installation 1-11
    part numbers 1-10, 4-15
    removal 1-10
    selection 1-9
serial port 2, MVME712x 4-14
serial port 4, MVME712x 4-16
Set Bug/Operating System Environment (ENV) firmware command 3-11
setting date and time 2-9, B-3
shielded cables A-4

INDEX

SIMs (serial interface modules) 1-9
size of base board A-1
slave address decoders, VMEbus 3-16
snoop control header 1-18
software-programmable hardware interrupts 4-20
specifications
  MVME162FX A-1
  related E-3
SRAM (static RAM)
  backup power source selection (J20/J22) 1-13
  battery options 4-9
  options 4-8
startup overview 1-1
startup problems, solving B-1
static variable space (firmware) 3-4
storage temperature A-1
streaming tape drive D-4
switches 2-1
switching directories 2-14
system considerations 1-22
system console setup 2-8
system controller jumper (J1) 1-8
System Fail (SYSFAIL*) signal, resetting 2-11
system reset 2-13
system startup 2-3

T

temperature
  operating A-1
  storage A-1
terminal
  configuration 2-4
  input/output control 3-5
terminator power, SCSI 1-24
tick timers 4-19
timeout
  global bus 1-23
  local bus 4-20
transition boards and serial interfaces 4-14, 4-17
troubleshooting procedures B-1
TX and RX clocks
  serial port 1 4-14
  serial port 2 4-15
typeface, meaning of 4
types of reset 2-13

V

vibration tolerance (operating) A-1
VMEbus
  connectors 4-21
  interface 4-12
  signals 5-9
VMEbus, "no" option 1-15
  firmware and 4-7
VMEchip2 ASIC 4-12

W

watchdog timers 4-19
Winchester hard drive D-3

X

XON/XOFF handshaking 2-8

Z

Z85230 serial communications controller (SCC) 2-8

INDEX

MVME162FX VME Embedded Controller Installation and Use

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

Gold-standard solutions

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

We buy equipment

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

Learn more!

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

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

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

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

ARTISAN® TECHNOLOGY GROUP