MVME172LX
VME Embedded Controller
Installation and Use

VME172LXA/IH4

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.

This material may be read online, downloaded for personal use, or included in another document as a URL (from this server). The text itself may not be published commercially in print or electronic form, edited or otherwise altered without the permission of Motorola, Inc.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 announced 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.

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 (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Motorola, Inc.
Computer Group
2900 South Diablo Way
Tempe, Arizona 85282-9602

Preface

This document provides general board level hardware description, hardware preparation and installation instructions, as well as debugger general information and instructions for the MVME172LX VME Embedded Controller (which is available in the versions listed below).

Assembly Item	Board Description
MVME172-213	4MB DRAM, 64 MHz MC68LC060, SCSI & Ethernet Interface
MVME172-223	4MB DRAM, 60 MHz MC68060, SCSI & Ethernet Interface
MVME172-233	4MB ECC DRAM, 64 MHz MC68LC060, SCSI & Ethernet Interface
MVME172-243	4MB ECC DRAM, 60 MHz MC68060, SCSI & Ethernet Interface
MVME172-253	16MB ECC DRAM, 64 MHz MC68LC060, SCSI & Ethernet Interface
MVME172-263	16MB ECC DRAM, 60 MHz MC68060, SCSI & Ethernet Interface
MVME172-303	8MB DRAM, 64 MHz MC68LC060, SCSI & Ethernet Interface
MVME172-313	8MB DRAM, 60 MHz MC68060, SCSI & Ethernet Interface
MVME172-323	8MB ECC DRAM, 64 MHz MC68LC060, SCSI & Ethernet Interface
MVME172-333	8MB ECC DRAM, 60 MHz MC68060, SCSI & Ethernet Interface
MVME172-343	32MB ECC DRAM, 64 MHz MC68LC060, SCSI & Ethernet Interface
MVME172-353	32MB ECC DRAM, 60 MHz MC68060, SCSI & Ethernet Interface
MVME172-363	16MB DRAM, 64 MHz MC68LC060, SCSI & Ethernet Interface
MVME172-373	16MB DRAM, 60 MHz MC68060, SCSI & Ethernet Interface

The MVME172LX VME Embedded Controller will be referred to as “MVME172LX” throughout this document.

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

Safety Summary
Safety Depends On You

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. The equipment is supplied with a three-conductor ac power cable. The power cable must be plugged into an approved three-contact electrical outlet. The power jack and mating plug of the power cable meet International Electrotechnical Commission (IEC) safety standards.

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.

Do Not Service or Adjust Alone.

Do not attempt internal service or adjustment unless another person capable of rendering first aid and resuscitation is present.

Use Caution When Exposing or Handling the CRT.

Breakage of the 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 the 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.

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.

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

WARNING

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

CE

European Notice: Board 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 (CISPR 22) Radio Frequency Interference

EN50082-1 (IEC801-2, IEC801-3, IEC801-4) Electromagnetic Immunity

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

This board product was 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.

Printed in the United States of America
October 1999

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
- MVME172LX Configuration 1-4
- System Controller Select Header (J1) 1-7
- IP Bus Clock Header (J11) 1-7
- SCSI Terminator Enable Header (J12) 1-8
- Thermal Sensing (J13) 1-8
- SRAM Backup Power Source Select Header (J14) 1-9
- Flash Write Protect Header (J16) 1-10
- IP Bus Strobe Select Header (J18) 1-10
- IP DMA Snoop Jumper (J19) 1-11
- EPROM/Flash Configuration Header (J20) 1-11
- General-Purpose Readable Jumpers Header (J21) 1-14
- Memory Mezzanine Options 1-16

Installation Instructions 1-17
- IP Installation on the MVME172LX 1-17
- MVME172LX Installation 1-18
- System Considerations 1-19
- Serial Connections 1-21

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 2-8
ROMboot 2-9
Network Boot 2-10
Restarting the System 2-11
Reset 2-11
Abort 2-12
Break 2-12
Diagnostic Facilities 2-13

CHAPTER 3 172Bug Firmware

Introduction 3-1
172Bug Overview 3-1
172Bug Implementation 3-3
Memory Requirements 3-3
Using 172Bug 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 172Bug Parameters 3-11
Configuring the IndustryPacks 3-19

CHAPTER 4 Functional Description

Introduction 4-1
Summary of Features 4-1
Block Diagram 4-3
Functional Description 4-5
Data Bus Structure 4-5
Microprocessor 4-5
MC68xx060 Cache 4-5
No-VMEbus-Interface Option 4-6
Memory Options 4-6
DRAM Options 4-6
SRAM Options 4-7
About the Batteries 4-8
EPROM and Flash Memory 4-9
Battery Backed Up RAM and Clock 4-10
VMEbus Interface and VMEchip2 4-10

I/O Interfaces 4-10
Serial Communications Interface 4-11
IP Interfaces 4-11
Ethernet Interface 4-12
SCSI Interface 4-12
SCSI Termination 4-13
Local Resources 4-13
Programmable Tick Timers 4-13
Watchdog Timer 4-14
Software-Programmable Hardware Interrupts 4-14
Local Bus Timeout 4-14
Local Bus Arbiter 4-15
Connectors 4-15

CHAPTER 5 Pin Assignments

Connector Pin Assignments 5-1
Remote Reset Connector - J2 5-2
IndustryPack A and B Connectors - J4/5/6, J3/7/8 5-2
Ethernet Connector - J9 5-4
Serial Connector - J17 5-4
Memory Mezzanine Connector 1 - J22 5-5
Memory Mezzanine Connector 2 - J15 5-7
SCSI Connector - J23 5-9
VMEbus Connectors (P1, P2) 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

Disk/Tape Controller Modules Supported D-1
Disk/Tape Controller Default Configurations D-2
IOT Command Parameters D-3

APPENDIX E Related Documentation

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

FIGURES

Figure 1-1. MVME172LX Switch, Header, Connector, and LED Locations 1-6
Figure 1-2. DB-25 DTE-to-RJ-45 Adapter 1-22
Figure 1-3. DB-25 DCE-to-RJ-45 Adapter 1-22
Figure 1-4. Typical RJ-45 Serial Cable 1-23
Figure 2-1. MVME172LX/Firmware System Startup 2-3
Figure 4-1. MVME172LX Block Diagram 4-4

TABLES

Table 1-1. Startup Overview 1-1
Table 1-2. MVME172LX Jumper Settings 1-5
Table 1-3. EPROM/Flash Mapping — 256K x 8 EPROMs 1-13
Table 1-4. EPROM/Flash Mapping — 512K x 8 EPROMs (default) 1-13
Table 1-5. EPROM/Flash Mapping — 1M x 8 EPROMs 1-14
Table 1-6. EPROM/Flash Mapping — 1M x 8 EPROMs, Onboard Flash Disabled 1-14
Table 1-7. Memory Mezzanine Stacking Options 1-16
Table 2-1. MVME172LX Front Panel Controls 2-1
Table 2-2. Software-Readable Jumpers 2-6
Table 3-1. Memory Offsets with 172Bug 3-4
Table 3-2. Debugger Commands 3-6
Table 3-3. ENV Command Parameters 3-11
Table 4-1. MVME172LX Features 4-1
Table 4-2. Local Bus Arbitration Priority 4-15
Table 4-3. J2 Pin Assignments 4-16
Table 5-1. Remote Reset Connector J2 Pin Assignments 5-2

Table 5-2. IndustryPack Interconnect Signals....5-3
Table 5-3. DB15 Ethernet Connector Pin Assignments....5-4
Table 5-4. Serial Connector Pin Assignments....5-4
Table 5-5. Mezzanine Connector J22 Pin Assignments....5-5
Table 5-6. Mezzanine Connector J15 Pin Assignments....5-7
Table 5-7. SCSI Connector J23 Pin Assignments....5-10
Table 5-8. VMEbus Connector P1 Pin Assignments....5-11
Table 5-9. VMEbus Connector P2 Pin Assignment....5-12
Table A-1. MVME172LX Specifications....A-1
Table B-1. Troubleshooting MVME172LX 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 MVME172LX VME Embedded Controller.

Getting Started

This section supplies an overview of startup procedures applicable to the MVME172LX. 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 Caution and Warning notes, 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.
Carry out any necessary jumper configuration on the MVME172LX board.	*Preparing the Board* on page 1-4.
Ensure that memory mezzanines and IP modules are properly installed on the MVME172LX board.	*Memory Mezzanine Options* on page 1-16; *IP Installation* on page 1-17.
Install the MVME172LX board in a chassis.	*MVME172LX Installation* on page 1-18.
Connect a display terminal.	*Serial Connections* on page 1-21.

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 2-2.
	Troubleshooting; Solving Startup Problems on page B-1.
Note that the firmware initializes and tests the board.	*Applying Power* on page 2-2.
	You may also wish to obtain the *172Bug Firmware User's Manual*, listed in the *Related Documentation* appendix.
Initialize the system clock.	*Using 172Bug*, Debugger Commands on page 3-6.
Examine and/or change environmental parameters.	*Using 172Bug*, 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 MVME172LX 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 MVME172LX 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 MVME172LX 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 MVME172LX, you may need to carry out certain hardware modifications before installing the module.

The MVME172LX provides software control over most options: by setting bits in control registers after installing the module in a system, you can modify its configuration. (The MVME172LX registers are described in Chapter 3 under *ENV – Set Environment*, and/or in the *MVME172 VME Embedded Controller Programmer's Reference Guide* as listed in "Related Documentation" in Appendix E.)

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

MVME172LX Configuration

Figure 1-1 illustrates the placement of the jumper headers, connectors, and various other components on the MVME172LX. Manually configurable jumper headers on the MVME172LX are listed in the following table (with default settings enclosed in brackets).

Table 1-2. MVME172LX Jumper Settings

Jumper	Function	Settings	
J1	VMEbus system controller selection	No jumper [1-2] 2-3	Not system controller. System controller. Automatic system controller.
J11	IP bus clock selection	[1-2] 2-3	IP bus clock = 8MHz. IP bus clock = local bus clock (30MHz / 32MHz).
J12	SCSI terminator status	No jumper [1-2]	Onboard SCSI bus terminators disabled. Onboard SCSI bus terminators enabled.
J14	SRAM backup power source selection	No jumper [1-3, 2-4] 3-5, 4-6 1-3, 4-6 3-5, 2-4	Backup power disabled (storage only). Primary +5V STBY; secondary +5V STBY. Primary onboard battery; secondary onboard batt. Primary +5V STBY; secondary onboard battery. Primary onboard battery; secondary +5V STBY.
J16	Flash write protection	No jumper [1-2]	Flash memory is write-protected. Flash memory is writable.
J18	IP bus strobe control	[No jumper] 1-2	No strobe signal from IP2 ASIC to IP bus. IP2 strobe present on IP bus.
J19	IP DMA snoop control	[No jumper] 1-2	Snooping inhibited. Snooping enabled.
J20	EPROM/Flash configuration	3-4, 9-11, 10-12 [5-6, 8-10, 9-11] 7-9, 8-10 1-2, 7-9, 8-10	256K x 8 EPROMs 512K x 8 EPROMs 1M x 8 EPROMs 1M x 8 EPROMs, on-board Flash disabled
J21	General-Purpose Readable Jumper configuration	9-10 empty [9-10]	Firmware defaults (in EPROM) selected. Flash selected. Other headers are user-definable. Factory configuration: all jumpers installed except 9-10.

Notes Items in brackets are factory default settings.

J13 is a two-pin header to access the THERM1 and THERM0 pins on the M68060. These pins are connected to an internal thermal resistor that can be used to obtain information about the average temperature of the die. J13 is not configurable.

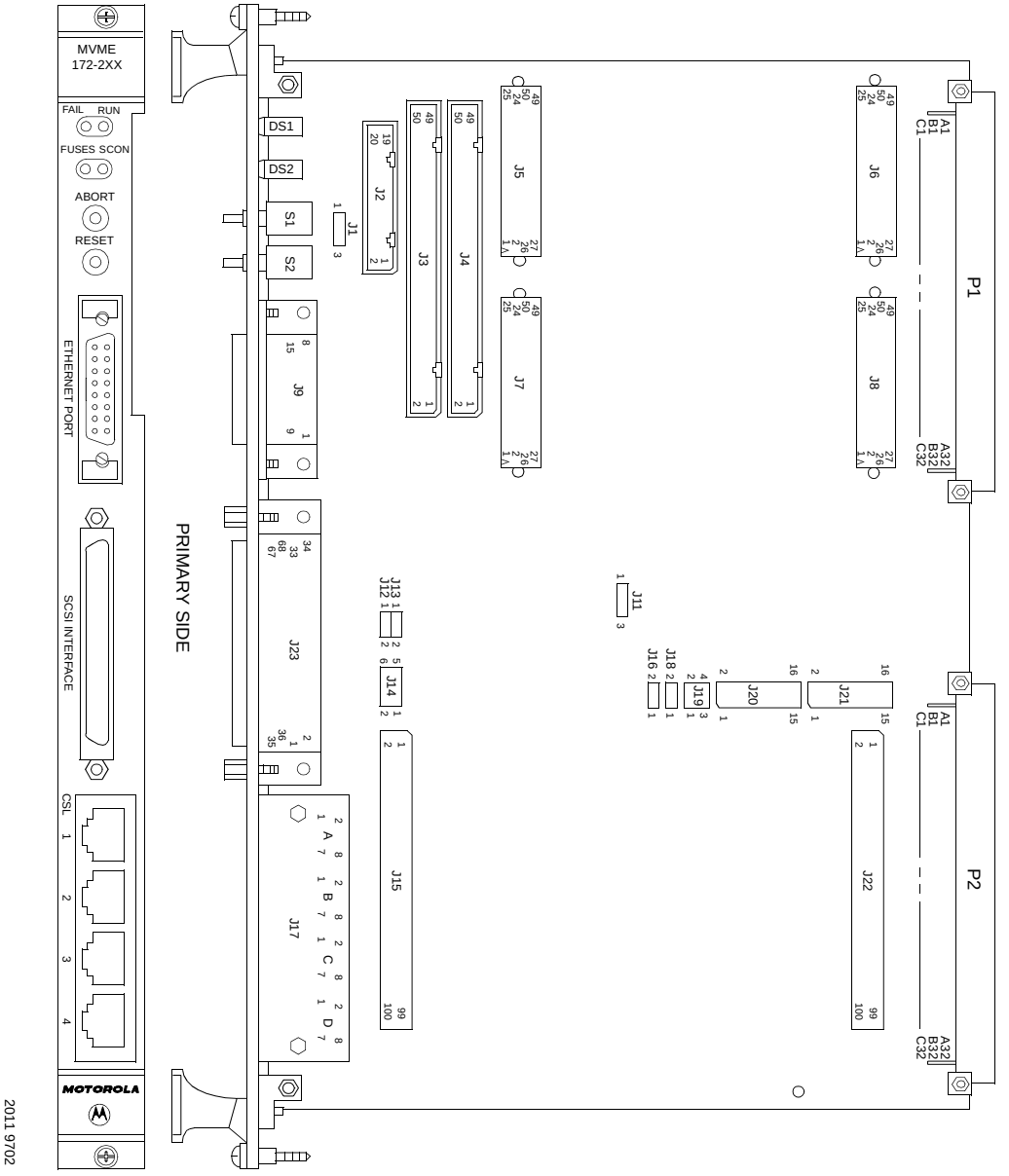
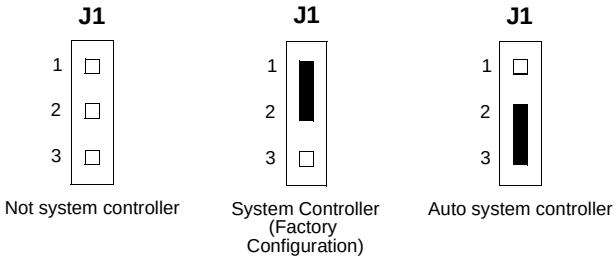


Figure 1-1. MVME172LX Switch, Header, Connector, and LED Locations

System Controller Select Header (J1)

The MVME172LX is factory-configured as a VMEbus system controller (i.e., a jumper is installed across pins 1 and 2 of header J1). Remove the J1 jumper if the MVME172LX is not to be the system controller. Note that when the MVME172LX is functioning as system controller, the **SCON** LED is turned on.

Note On MVME172s without the optional VMEbus interface (i.e., with no VMEchip2 ASIC), the jumper may be installed or removed without affecting normal operation.



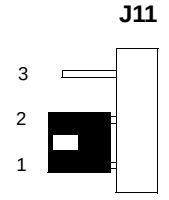
IP Bus Clock Header (J11)

J11 selects the speed of the IP bus clock. The IP bus clock speed may be either 8MHz or the speed of the local bus clock (30MHz for the MC68060, 32MHz for the MC68LC060). The default factory configuration has a jumper installed on J11 pins 1 and 2, denoting an 8MHz clock.

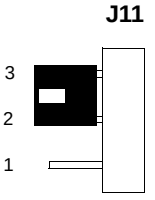
If the jumper is installed on J11 between pins 2 and 3, the IP bus clock speed matches that of the MC68060 or MC68LC060 bus clock (30/32MHz), thus allowing the IP module to run with a 30/32MHz MPU. Whether the setting is 8MHz or 30/32MHz, all IP ports operate at the same speed.

Caution

The setting of the IP32 CSR bit (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) for 30/32MHz. If the jumper and the CSR bit are not configured the same, the board may not run properly.



8MHz IP Bus Clock
(Factory configuration)

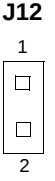


30/32MHz IP Bus Clock
(from MPU Bus Clock)

SCSI Terminator Enable Header (J12)

The MVME172LX provides terminators for the SCSI bus. The SCSI terminators are enabled/disabled by a jumper on header J12. The SCSI terminators may be configured as follows:

Onboard SCSI Bus Terminator Enabled
(Factory Configuration)



Onboard SCSI Bus Terminator Disabled

Note If the MVME172LX is to be used at one end of the SCSI bus, the SCSI bus terminators must be enabled.

Thermal Sensing (J13)

J13 is a two-pin header to access the M68060 THERM1 and THERM0 pins. These pins are connected to an internal thermal resistor that can be used to provide information about the average temperature of the die. J13 is not configurable.

SRAM Backup Power Source Select Header (J14)

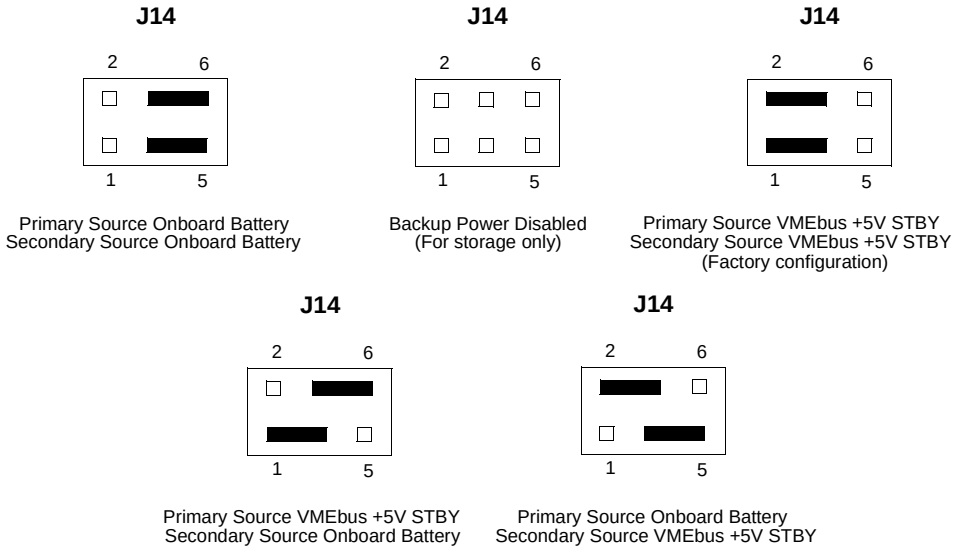
Header J14 determines the source for onboard static RAM backup power on the MVME172LX.

The following backup power configurations are available for onboard SRAM through header J14. In the factory configuration, the VMEbus +5V standby voltage serves as primary and secondary power source (the onboard battery is disconnected).

Note For MVME172LXs without the optional VMEbus interface (i.e., without the VMEchip2 ASIC), you must select the onboard battery as the backup power source.

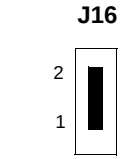
Caution

Removing all jumpers may temporarily disable the SRAM. Do not remove all jumpers from J14, except for storage.



Flash Write Protect Header (J16)

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



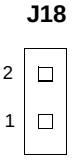
Flash Write Protect
(Factory configuration)

IP Bus Strobe Select Header (J18)

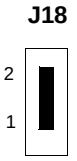
Some IP bus implementations make use of the Strobe∗ signal (pin 46) as an input to the IP modules from the IP2 chip. Other IP interfaces require that the strobe be disconnected.

With a jumper installed between J18 pins 1 and 2, a programmable frequency source is connected to the Strobe∗ signal on the IP bus. Refer to the IP2 chip programming section in the *MVME172 Embedded Controller Programmer's Reference Guide* for additional information.

If the jumper is removed from J18, the strobe line is available for a sideband type of messaging between IP modules. The Strobe∗ signal is not connected to any active devices on the board, but it may be connected to a pull-up resistor.



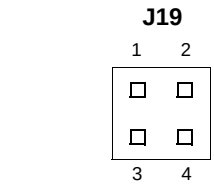
IP Strobe disconnected
(Factory configuration)



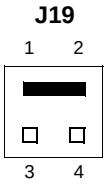
IP Strobe connected

IP DMA Snoop Jumper (J19)

J19 defines the state of the snoop control bus when an IP DMA controller is local bus master. J19 pins 1 and 2 enable/disable the Snoop Control signal on the MC68060 processor (pins 3-4 have no function on MVME172LX boards).



Snoop Inhibited (factory configuration)



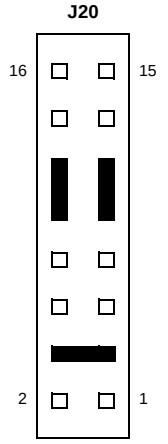
Snoop Enabled

For the MVME172LX, shorting pins 1-2 enables snooping. Leaving pins 1-2 disconnected (the factory configuration) inhibits snooping. With snooping enabled, the snoop signal to the MC68060 processor is driven low during IP DMA operations.

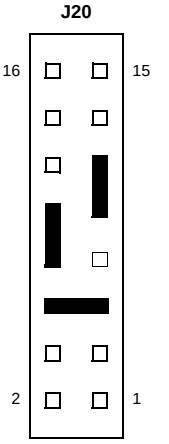
EPROM/Flash Configuration Header (J20)

The MVME172LX can be ordered with 2MB of Flash memory and two EPROM sockets ready for the installation of EPROMs, which may be ordered separately. The EPROM locations are standard JEDEC 32-pin DIP sockets that support three jumper-selectable densities (256 Kbit x 8; 512 Kbit x 8 — the factory default; 1 Mbit x 8) and which in addition permit disabling of the Flash memory.

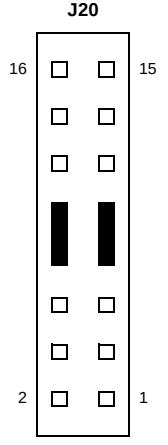
J20 supplies eight jumper headers for configuration of the EPROM sockets.



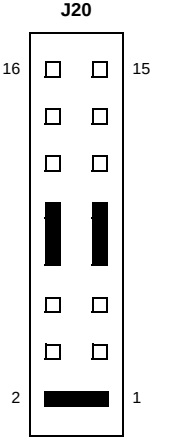
CONFIGURATION 1: 256K x 8 EPROMs



CONFIGURATION 2: 512K x 8 EPROMs (default)



CONFIGURATION 3: 1M x 8 EPROMs



CONFIGURATION 4: 1M x 8 EPROMs
ONBOARD FLASH DISABLED

2190 9804

The next four tables show the address range for each EPROM socket in all four configurations. GPI4 (J21 pins 9-10) is a control bit in the MC2chip ASIC that allows reset code to be fetched from Flash memory or from EPROMs.

Table 1-3. EPROM/Flash Mapping — 256K x 8 EPROMs

GPI4		Address Range	Device Accessed
Removed	1	$FF800000 - $FF83FFFF	EPROM A (XU1)
		$FF840000 - $FF87FFFF	EPROM B (XU2)
		$FFA00000 - $FFBFFFFF	Onboard Flash
Installed	0	$FF800000 - $FF9FFFFF	Onboard Flash
		$FFA00000 - $FFA3FFFF	EPROM A (XU1)
		$FFA40000 - $FFA7FFFF	EPROM B (XU2)

Table 1-4. EPROM/Flash Mapping — 512K x 8 EPROMs (default)

GPI4		Address Range	Device Accessed
Removed	1	$FF800000 - $FF87FFFF	EPROM A (XU1)
		$FF880000 - $FF8FFFFF	EPROM B (XU2)
		$FFA00000 - $FFBFFFFF	Onboard Flash
Installed	0	$FF800000 - $FF9FFFFF	Onboard Flash
		$FFA00000 - $FFA7FFFF	EPROM A (XU1)
		$FFA80000 - $FFAFFFFF	EPROM B (XU2)

Table 1-5. EPROM/Flash Mapping — 1M x 8 EPROMs

GPI4		Address Range	Device Accessed
Removed	1	$FF800000 - $FF8FFFFF	EPROM A (XU1)
		$FF900000 - $FF9FFFFF	EPROM B (XU2)
		$FFA00000 - $FFBFFFFF	Onboard Flash
Installed	0	$FF800000 - $FF9FFFFF	Onboard Flash
		$FFA00000 - $FFAFFFFF	EPROM A (XU1)
		$FFB00000 - $FFBFFFFF	EPROM B (XU2)

Table 1-6. EPROM/Flash Mapping — 1M x 8 EPROMs, Onboard Flash Disabled

GPI4		Address Range	Device Accessed
Removed	1	$FF800000 - $FF8FFFFF	EPROM A (XU1)
		$FF900000 - $FF9FFFFF	EPROM B (XU2)
		Not used	Onboard Flash
Installed	0	Not used	Onboard Flash
		$FF800000 - $FF8FFFFF	EPROM A (XU1)
		$FF900000 - $FF9FFFFF	EPROM B (XU2)

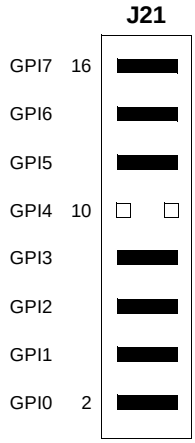
General-Purpose Readable Jumpers Header (J21)

Header J21 provides eight software-readable jumpers. These jumpers can be read as a register (at address $FFF4202D) in the MC2chip LCSR. Bit 0 is associated with header pins 1-2; bit 7 is associated with pins 15-16. The bit values are read as a **0** when the jumper is installed, and as a **1** when the jumper is removed. The MVME172LX is shipped from the factory with J21 set to all **0**s (jumpers on all pins) except for GPI4, as diagrammed below.

If the MVME172BUG firmware is installed, three jumpers are user-definable (i.e., pins 11-12, 13-14, 15-16). If the MVME172BUG firmware is not installed, seven jumpers are user-definable (i.e., pins 1-2, 3-4, 5-6, 7-8, 11-12, 13-14, 15-16).

Note Pins 9-10 (GPI4) are reserved to select either the Flash memory map (jumper installed) or the EPROM memory map (jumper removed). They are not user-definable. The address ranges for the various EPROM/Flash configurations appear in the preceding section of this chapter.

The MVME172LX is shipped from the factory with J21 set to all zeros (jumpers on all pins) except for GPI4.



	172BUG Installed (default)	User Code Installed
GPI7	USER-DEFINABLE	USER-DEFINABLE
GPI6	USER-DEFINABLE	USER-DEFINABLE
GPI5	USER-DEFINABLE	USER-DEFINABLE
GPI4	IN=FLASH; OUT=EPROM	IN=FLASH; OUT=EPROM
GPI3	REFER TO DEBUG MANUAL	REFER TO DEBUG MANUAL
GPI2	REFER TO DEBUG MANUAL	REFER TO DEBUG MANUAL
GPI1	REFER TO DEBUG MANUAL	REFER TO DEBUG MANUAL
GPI0	REFER TO DEBUG MANUAL	REFER TO DEBUG MANUAL

EPROMs Selected (factory configuration)

Memory Mezzanine Options

Two 100-pin connectors (J15 and J22) are provided on the MVME172LX to accommodate optional memory mezzanine boards. Two memory mezzanine options are available for the MVME172LX:

- 4, 8, 16MB parity DRAM
- 4, 8, 16, 32, 64MB ECC DRAM

The mezzanine boards may either be used individually or be combined in a stack (not more than two deep). The following connector options govern stacking arrangements:

- The 4, 8, and 16MB parity DRAM board has connectors on the bottom only; it must be either the only mezzanine or the upper mezzanine.
- All ECC DRAM boards are available with two connector options:
  - Connectors on both the top and bottom
  - Connectors on the bottom only; must be either the only mezzanine or the upper mezzanine

When the mezzanines are stacked, the following combinations are possible:

Table 1-7. Memory Mezzanine Stacking Options

Upper	None	None	Parity DRAM	ECC DRAM
Lower	Parity DRAM	ECC DRAM	ECC DRAM	ECC DRAM

Installation Instructions

This section covers:

- Installation of IndustryPacks (IPs) on the MVME172LX
- Installation of the MVME172LX in a VME chassis
- System considerations relevant to the installation. Ensure that EPROM devices are installed as needed. Ensure that all header jumpers are configured as desired.

IP Installation on the MVME172LX

Up to two IP modules may be installed on the MVME172LX. Install the IPs on the MVME172LX as follows:

1. Each IP module has two 50-pin connectors that plug into two corresponding 50-pin connectors on the MVME172LX: J5/J6, J7/J8. See Figure 2-1 for the MVME172LX connector locations.
   - Orient the IP module(s) so that the tapered connector shells mate properly. Plug IP_a into connectors J5 and J6; plug IP_b into J7 and J8. If a double-sized IP is used, plug IP_ab into J5, J6, J7, and J8.
2. Two additional 50-pin connectors (J3 and J4) are provided behind the MVME172LX front panel for external cabling connections to the IP modules. There is a one-to-one correspondence between the signals on the cabling connectors and the signals on the associated IP connectors (i.e., J4 has the same IP_a signals as J5; J3 has the same IP_b signals as J7).
   - Connect user-supplied 50-pin cables to J3 and J4 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 slot in the MVME172LX front panel and attach them to the appropriate external equipment, depending on the nature of the particular IP(s).

MVME172LX Installation

With EPROMs and IP modules installed and headers properly configured, proceed as follows to install the MVME172LX in the 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 MVME172LX.
   - If you intend to use the MVME172LX 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 MVME172LX as system controller, it can occupy any unused double-height card slot.
4. Slide the MVME172LX 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 MVME172LX in the chassis with the screws provided, making good contact with the transverse mounting rails to minimize RF emissions.

6. On the chassis backplane, remove the INTERRUPT ACKNOWLEDGE (IACK) and BUS GRANT (BG) jumpers from the header for the card slot occupied by the MVME172LX.
7. Connect the appropriate cable(s) to the MVME172LX panel connectors for the EIA-232-D serial ports, SCSI port, and LAN Ethernet port.
   – Note that some cables are not provided with the MVME172LX and must be made or purchased by the user. (Motorola recommends shielded cable for all peripheral connections to minimize radiation.)
8. Connect the peripheral(s) to the cable(s).
9. Install any other required VMEmodules in the system.
10. Replace the chassis cover.
11. Connect the power cable to the AC power source and turn the equipment power ON.

System Considerations

The MVME172LX draws power from both the P1 and the P2 connectors on the VMEbus backplane. 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 MVME172LX may not operate properly without its main board connected to VMEbus backplane connectors P1 and P2.

Whether the MVME172LX operates as a VMEbus master or as a VMEbus slave, it is configured for 32 bits of address and 32 bits of data (A32/D32). However, it handles A16 or A24 devices in the address ranges indicated in Chapter 3. D8 and/or D16 devices in the system must be handled by the MC68060/MC68LC060 software. Refer to the memory maps in the *MVME172 VME Embedded Controller Programmer's Reference Guide.*

The MVME172LX 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 MVME172Bug firmware. This may be changed via software to any other base address. Refer to the *MVME172 VME Embedded Controller Programmer's Reference Guide* for more information.

If the MVME172LX 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 MVME172LX waits forever for the VMEbus cycle to complete. This will cause the system to lock up. There is only one situation in which the system might lack this global bus timeout: when the MVME172LX is not the system controller and there is no global bus timeout elsewhere in the system.

Multiple MVME172LXs may be installed in a single VME chassis. In general, hardware multiprocessor features are supported.

Note If you are installing multiple MVME172LXs in an MVME945 chassis, do not install an MVME172LX 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 includes four bits that function as location monitors to allow one MVME172LX processor to broadcast a signal to any other MVME172LX processors. All eight registers 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:

- LAN AUI
- SCSI terminator
- Remote reset connector
- IndustryPack 5V
- ±12V

The **FUSES** LED illuminates to indicate that all fuses are functioning correctly. If a fuse opens, power must be removed for several minutes to allow the fuse to return to a closed or shorted condition.

Serial Connections

The MVME172LX uses two Zilog Z85230 serial port controllers to implement the four serial communications interfaces. Each interface supports CTS, DCD, RTS, and DTR control signals as well as the TXD and RXD transmit/receive data signals. Because the serial clocks are omitted in the MVME172LX implementation, serial communications are strictly asynchronous. The Z85230 is interfaced as DTE (data terminal equipment) with EIA-232-D signal levels. The serial ports are routed to four RJ-45 connectors on the front panel.

For additional information on the EIA-232-D interface, refer to the *EIA-232-D Standard*.

Connection diagrams for the four serial ports on the MVME172LX are provided in the following figures. These ports are connected to external devices through cables connected to the front panel.

Figure 1-2 diagrams the pin assignments required in a cable to adapt a DB-25 DTE device to the RJ-45 connectors.

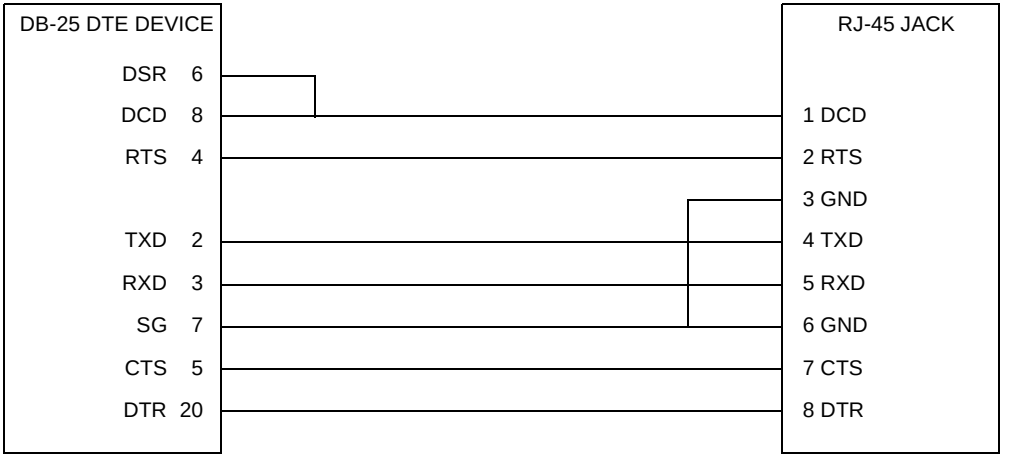


Figure 1-2. DB-25 DTE-to-RJ-45 Adapter

Figure 1-3 diagrams the pin assignments required in a cable to adapt a DB-25 DCE device to an RJ-45 connector.

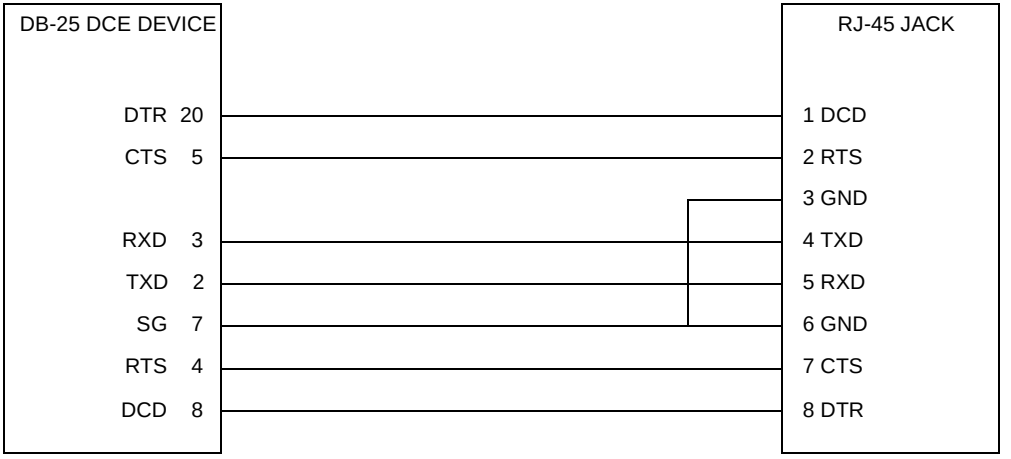


Figure 1-3. DB-25 DCE-to-RJ-45 Adapter

Figure 1-4 diagrams the pin assignments required in a typical 8-conductor serial cable having RJ-45 connectors at both ends. Note that all wires are crossed.

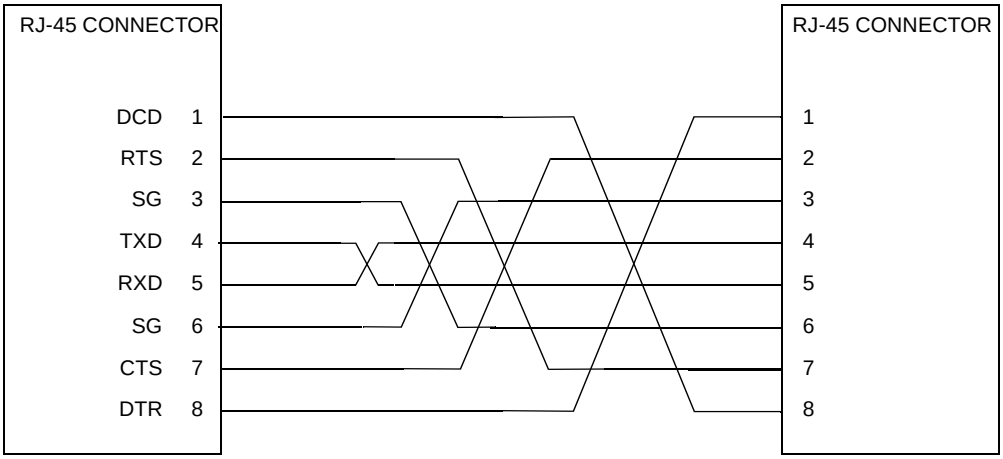


Figure 1-4. Typical RJ-45 Serial Cable

Startup and Operation 2

Introduction

This chapter provides information on powering up the MVME172LX 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 *MVME172 Embedded Controller Programmer's Reference Guide*.

Front Panel Switches and Indicators

There are two switches (**ABORT** and **RESET**) and four LED (light-emitting diode) status indicators (**FAIL**, **RUN**, **SCON** and **FUSES**) located on the MVME172LX front panel.

Table 2-1. MVME172LX 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 MVME172LX Flash memory. The interrupter connected to the ABT 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 MVME172LX 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*.

Table 2-1. MVME172LX Front Panel Controls

Control/Indicator	Function
FAIL LED (red)	Board failure. Lights when the BRDFAIL* signal line is active. Part of DS1.
RUN LED (green/amber)	CPU activity. Lights when the local bus TIP* signal line is active. This indicates that one of the local bus masters is executing a local bus cycle. Part of DS1.
SCON LED (green)	System controller. Lights when the MVME172LX is functioning as VMEbus system controller. Part of DS2.
FUSES LED (green)	Fuse OK. Indicates that +5Vdc, +12Vdc, and –12Vdc power is available to the LAN and SCSI interfaces and IP connectors. Part of DS2.

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 MVME172LX's MPU, hardware, and firmware along with the rest of the system.

The Flash-resident firmware initializes the devices on the MVME172LX board in preparation for booting the operating system. The firmware is shipped from the factory with an appropriate set of defaults. In most cases there is no need to modify the firmware configuration before you boot the operating system. For specifics in this regard, refer to Chapter 3 and to the user documentation for the MVME172Bug 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 MVME172LX hardware. The figure below charts the flow of the basic initialization sequence that takes place during system startup.

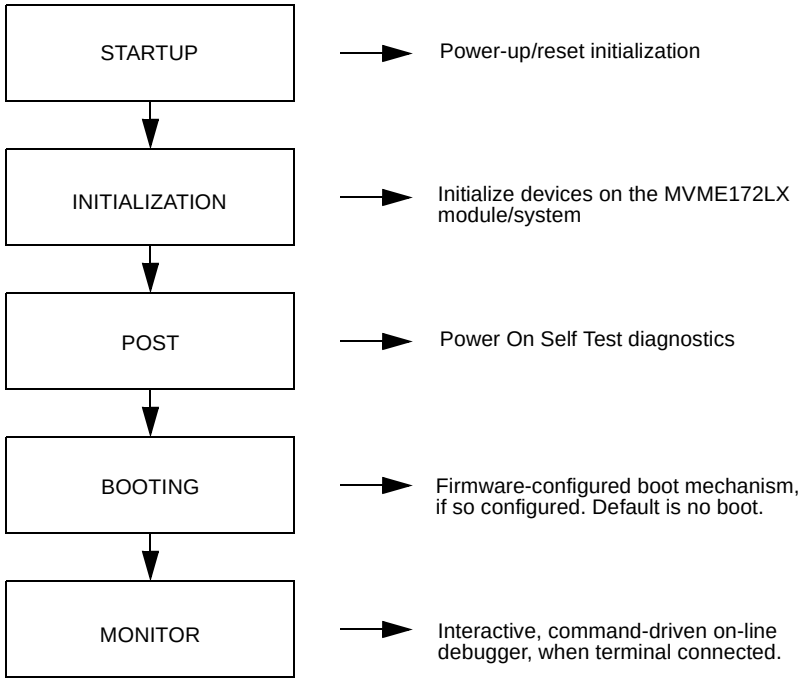


Figure 2-1. MVME172LX/Firmware System Startup

2

Pre-Startup Checklist

Before you power up the MVME172LX system, be sure that the following conditions exist:

1. Jumpers and/or configuration switches on the MVME172LX VME Embedded Controller and associated equipment are set as required for your particular application.
2. The MVME172LX board is installed and cabled up as appropriate for your particular chassis or system, as outlined in Chapter 1.
3. The terminal that you plan to use as the system console is connected to the console port (serial port 1) on the MVME172LX module.
4. The terminal is set up as follows:
   - Eight bits per character
   - One stop bit per character
   - Parity disabled (no parity protection)
   - Baud rate 9600 baud (the default baud rate of many serial ports at power-up)
5. Any other device that you wish to use, such as a host computer system and/or peripheral equipment, is cabled to the appropriate connectors.

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

Bringing up the Board

The MVME172LX comes with 172Bug firmware installed. For the firmware to operate properly with the board, you must follow the steps below.

Caution

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

1. Turn all equipment power OFF. Refer to *MVME172LX Configuration* on page 1-4 and configure jumpers on headers as necessary for your particular application.

   a. The jumper block for header J21 contains eight bits, which all affect 172Bug operation. They are read as a register (at location $FFF4202D) on the MC2chip ASIC. (The *MVME172 VME Embedded Controller Programmer's Reference Guide* contains additional information on the MC2chip.)

      The bit values are read as a **0** when the corresponding jumper is installed, or as a **1** when the jumper is removed.

      The default configuration for J21 on the MVME172LX has seven jumpers installed (no jumper between pins 9-10).

      The 172Bug firmware reserves/defines the four lower order bits (GPI3 to GPI0). Table 2-2 describes the bit assignments on J21.

   b. Configure header J1 by installing/removing a jumper between pins 1 and 2. A jumper installed/removed enables/disables the system controller function on the MVME172LX.

2. Refer to the setup procedure for your particular chassis or system for details concerning the installation of the MVME172LX.

Table 2-2. Software-Readable Jumpers

J21 Bit No.	Pins	Function
Bit #0 (GPI0)	1-2	When set to 1 (high), this bit instructs the debugger to use local static RAM for its work page (i.e., variables, stack, vector tables, etc.).
Bit #1 (GPI1)	3-4	When set to 1 (high), this bit instructs the debugger to use the default setup/operation parameters in ROM 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/ROM defaults.
Bit #2 (GPI2)	5-6	Reserved for future use.
Bit #3 (GPI3)	7-8	Reserved for future use.
Bit #4 (GPI4)	9-10	When set to 0 (low), this bit informs the debugger that it is executing out of Flash memory. When set to 1 (high), it informs the debugger that it is executing out of the PROM.
Bit #5 (GPI5)	11-12	Open to your application.
Bit #6 (GPI6)	13-14	Open to your application.
Bit #7 (GPI7)	15-16	Open to your application.

3. Connect the terminal that is to be used as the 172Bug system console to the default debug EIA-232-D port at serial port 1 on the front panel of the MVME172LX. Refer to *Serial Connections* in Chapter 1 for other connection options. Set the terminal up as follows:
   - Eight bits per character
   - One stop bit per character
   - Parity disabled (no parity)
   - Baud rate 9600 baud (default baud rate of MVME172LX ports at powerup)

   After power-up, you can reconfigure the baud rate of the debug port by using the 172Bug Port Format (**PF**) command.

Note In order for high-baud-rate serial communication between 172Bug and the terminal to work, the terminal must do some form of handshaking. If the terminal being used does not do hardware handshaking via the CTS line, then it must do XON/XOFF handshaking. If you get garbled messages and missing characters, then you should check the terminal to make sure XON/XOFF handshaking is enabled.

4. If you want to connect devices (such as a host computer system and/or a serial printer) to the other EIA-232-D port connectors, connect the appropriate cables and configure the port(s) as detailed in step 3 above. After power-up, you can reconfigure these port(s) by programming the MVME172LX Z85230 Serial Communications Controllers (SCCs), or by using the 172Bug **PF** command.

5. The EPROM/Flash header J20 must be set to configuration 3, with jumpers between J20 pins 5-6, 8-10, and 9-11. This sets it up for 512K x 8 EPROMs.

6. Power up the system. 172Bug executes some self-checks and displays the debugger prompt `172-Bug>` if the firmware is in Board mode.

   However, if the **ENV** command has put 172Bug in System mode, the system performs a self-test and tries to autoboot. Refer to the **ENV** and **MENU** commands (Table 4-3).

   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.

7. Before using the MVME172LX after the initial installation, set the date and time using the following command line structure:

   `172-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, 1999:

   `172-Bug>` **SET 0907991037**

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

Autoboot

Autoboot is a software routine that is contained in the 172Bug Flash/PROM 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 upon the system console:

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

A delay follows this message so that you can abort the Autoboot process if you wish. Then the actual I/O begins: the program pointed to within the volume ID of the media specified is loaded into RAM and control passed to it. If, however, during this time you want to gain control without Autoboot, 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 MVME172LX. 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 MVME172LX 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, 172Bug occupies an EPROM installed in socket XU2. This leaves one socket (XU1) and the Flash memory 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 MVME172LX 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 contained in the 172Bug Flash/PROM that provides a mechanism for booting an operating system using a network (local Ethernet interface) as the boot device. The Network Auto Boot routine automatically scans for controllers and devices in a specified sequence until a valid bootable device containing boot media is found or until the list is exhausted. If a valid bootable device is found, a boot from that device is started. The controller scanning sequence goes from the lowest controller Logical Unit Number (LUN) detected to the highest LUN detected. (Refer to Appendix C for default LUNs.)

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

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

After this message, there is a delay to let you abort the Auto Boot process if you wish. Then the actual I/O is begun: the program pointed to within the volume ID of the media specified is loaded into RAM and control passed to it. If, however, during this time you want to gain control without Network Boot, 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 MVME172LX initiates a system reset. Resets can also be asserted through the **RESET** switch on the MVME172LX front panel, or by software.

For details on resetting the MVME172LX board through software, refer to the *MVME172 Embedded Controller Programmer's Reference Guide*.

Both "cold" and "warm" reset modes are available. By default, 172Bug is in "cold" mode. During *cold* resets, a total system initialization takes place, as if the MVME172LX 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 172Bug variables and tables are preserved, as well as the target state registers and breakpoints.

Note that when the MVME172LX comes up in a cold reset, 172Bug runs in Board mode. Using the Environment (**ENV**) or **MENU** commands can make 172Bug 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 172Bug environment is ever lost (vector table is destroyed, stack corrupted, etc.).

Abort

Aborts are invoked by pressing and releasing the **ABORT** switch on the MVME172LX 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 is returned 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 prior to its completion (for example, during the display of a large block of memory). Break allows you to terminate the command.

Diagnostic Facilities

The 172Bug package includes a set of hardware diagnostics for testing and troubleshooting the MVME172LX. 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 `172-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.

172Bug Firmware 3

Introduction

The 172Bug firmware is the layer of software just above the hardware. The firmware supplies the appropriate initialization for devices on the MVME172LX board upon power-up or reset.

This chapter describes the basics of 172Bug 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 172Bug commands appears at the end of the chapter.

For complete user information about 172Bug, refer to the *Debugging Package for Motorola 68K CISC CPUs User's Manual* and to the *MVME172Bug Diagnostics User's Manual*, listed under *Related Documentation*.

172Bug 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. This member of the M68000 Firmware family is implemented on the MVME172LX MC68060- or MC68LC060-based Embedded Controller, and is known as the MVME172BUG, or 172Bug. It includes diagnostics for testing and configuring IndustryPack modules.

172Bug is a powerful evaluation and debugging tool for systems built around MVME172LX CISC-based microcomputers. Facilities are available for loading and executing user programs under complete operator control for system evaluation. The 172Bug firmware provides a high degree of functionality, user friendliness, portability, and ease of maintenance.

172Bug 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 172Bug routines that handle I/O, data conversion, and string functions available to user programs.

172Bug consists of three parts:

- A command-driven user-interactive *software debugger*, described in this chapter. It is referred to here as "the debugger" or "172Bug".
- A command-driven *diagnostic package* for the MVME172LX hardware, referred to here as "the diagnostics".
- A *user interface* or *debug/diagnostics monitor* that accepts commands from the system console terminal.

When using 172Bug, you operate out of either the *debugger directory* or the *diagnostic directory*.

- If you are in the debugger directory, the debugger prompt `172-Bug>` is displayed and you have all of the debugger commands at your disposal.
- If you are in the diagnostic directory, the diagnostic prompt `172-Diag>` is displayed and you have all of the diagnostic commands at your disposal as well as all of the debugger commands.

Because 172Bug is command-driven, it performs its various operations in response to user commands entered at the keyboard. When you enter a command, 172Bug 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 172Bug, 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 172Bug very similar. Some effort has also been made to make the interactive commands more consistent. For example, delimiters between commands and arguments may be commas or spaces interchangeably.

172Bug Implementation

MVME172Bug 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, 172Bug is contained in a single 27C040 DIP EPROM installed in socket XU2, providing 512KB (128K longwords) of storage. Optionally, the 172Bug can be loaded and executed in a single Flash memory chip. The executable code is checksummed at every power-on or reset firmware entry, and the result (which includes a pre-calculated checksum contained in the memory devices) is verified against the expected checksum.

Memory Requirements

The program portion of 172Bug is approximately 512KB of code, consisting of download, debugger, and diagnostic packages and contained entirely in Flash memory or PROM.

The 172Bug executes from address $FF800000 whether in Flash or EPROM. If you remove the jumper at J21 pins 9 and 10, the address spaces of the Flash and EPROM are swapped. For MVME172-2*xx* series boards (MVME172LX), the factory ship configuration is with jumper J21 pins 9-10 removed (172Bug operating out of EPROM).

The 172Bug initial stack completely changes 8KB of SRAM memory at address offset $C000 from the SRAM base address, at power-up or reset.

Table 3-1. Memory Offsets with 172Bug

Type of Memory Present	Default DRAM Base Address	Default SRAM Base Address
Single DRAM mezzanine	$00000000	$FFE00000 (onboard SRAM)
Single SRAM mezzanine	N/A	$00000000
DRAM mezzanine stacked with SRAM mezzanine	$00000000	$E1000000
Two DRAM mezzanines stacked	$00000000	$FFE00000 (onboard SRAM)

DRAM can be ECC or parity type. DRAM mezzanines are mapped in contiguously starting at zero ($00000000), largest first. With two mezzanines of the same size but different types, parity DRAM is mapped to the selected base address and the ECC mezzanine will follow. If both mezzanines are ECC type, the bottom one is first.

The 172Bug requires 2KB of NVRAM for storage of board configuration, communication, and booting parameters. This storage area begins at $FFFC16F8 and ends at $FFFC1EF7.

172Bug 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 172Bug stack and static variable space and the rest is reserved as user space. Whenever the MVME172LX is reset, the target PC is initialized to the address corresponding to the beginning of the user space, and the target stack pointers are initialized to addresses within the user space, with the target Interrupt Stack Pointer (ISP) set to the top of the user space.

Using 172Bug

172Bug is command-driven; it performs its various operations in response to commands that you enter at the keyboard. When the `172-Bug>` prompt appears on the terminal screen, the debugger is ready to accept debugger commands. When the `172-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 172Bug debugger commands are summarized in the following table. The commands are described in detail in the *Debugging Package for Motorola 68K CISC CPUs User's Manual.*

Table 3-2. Debugger Commands

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

Table 3-2. Debugger Commands (Continued)

Command	Description
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
NIOT	Network I/O Teach
NPING	Network Ping
OF	Offset Registers Display/Modify

Table 3-2. Debugger Commands (Continued)

Command	Description
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, 172Bug, to modify certain parameters contained in the MVME172LX's Non-Volatile RAM (NVRAM), also known as Battery Backed-Up RAM (BBRAM).

- The Board Information Block in NVRAM contains various elements concerning operating parameters of the hardware. Use the 172Bug command **CNFG** to change those parameters.
- Use the 172Bug command **ENV** to change configurable 172Bug 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 172Bug debugger, along with the parameters that can be configured with the **ENV** command.

CNFG - Configure Board Information Block

Use this command to display and configure the Board Information Block which is resident within the NVRAM. The board information block contains various elements that correspond to specific operational parameters of the MVME172LX board. The board structure for the MVME172LX is as follows:

```
172-Bug>cnfg
Board (PWA) Serial Number = "     "
Board Identifier = "     "
Artwork (PWA) Identifier = "     "
MPU Clock Speed = "     "
Ethernet Address = 08003E200000
Local SCSI Identifier = "     "
Parity Memory Mezzanine Artwork (PWA) Identifier = "     "
Parity Memory Mezzanine (PWA) Serial Number = "     "
Static Memory Mezzanine Artwork (PWA) Identifier = "     "
Static Memory Mezzanine (PWA) Serial Number = "     "
```

```
ECC Memory Mezzanine #1 Artwork (PWA) Identifier = "     "
ECC Memory Mezzanine #1 (PWA) Serial Number = "     "
ECC Memory Mezzanine #2 Artwork (PWA) Identifier = "     "
ECC Memory Mezzanine #2 (PWA) Serial Number = "     "
Serial Port 2 Personality Artwork (PWA) Identifier = "     "
Serial Port 2 Personality Module (PWA) Serial Number = "    "
IndustryPack A Board Identifier = "     "
IndustryPack A (PWA) Serial Number = "     "
IndustryPack A Artwork (PWA) Identifier = "     "
IndustryPack B Board Identifier = "     "
IndustryPack B (PWA) Serial Number = "     "
IndustryPack B Artwork (PWA) Identifier = "     "
IndustryPack C Board Identifier = "     "
IndustryPack C (PWA) Serial Number = "     "
IndustryPack C Artwork (PWA) Identifier = "     "
IndustryPack D Board Identifier = "     "
IndustryPack D (PWA) Serial Number = "     "
IndustryPack D Artwork (PWA) Identifier = "     "
172-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 zeroes if the length is not met.

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

Refer to the *MVME172 VME 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 172Bug 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 MVME172LX that affect these parameters appears in your *MVME172 VME Embedded Controller Programmer's Reference Guide*.

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

Configuring the 172Bug 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-172 environment, each MVME172LX 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 MVME172LX. 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 MVME172LX GCSR GPCSR0 as accessed through VMEbus A16 space; it assumes the MVME172LX GRPAD (group address) and BDAD (board address within group) switches are set to "on". This byte-wide value is initialized to $FF by MVME172LX hardware after a System or Power-On reset. In a multi-172 environment, where the work pages of several Bugs reside in the memory of the primary (first) MVME172, 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 MVME172 series modules. You should have a working knowledge of the VMEchip2 as given in the *MVME172 VME 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 MVME172LX. 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 the 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	04	Defines the ROM access time. The default is $04, which sets an access time of five clock cycles of the local bus.
Flash Access Time Code	03	Defines the Flash access time. The default is $03, which sets an access time of four clock cycles of the local bus.
MCC Vector Base VMEC2 Vector Base #1 VMEC2 Vector Base #2	05 06 07	Base interrupt vector for the component specified. Default: MC2chip = $05, VMEchip2 Vector 1 = $06, VMEchip2 Vector 2 = $07.
VMEC2 GCSR Group Base Address	D2	Specifies group address ($FFFF*XX*00) in Short I/O for this board. Default = $D2.
VMEC2 GCSR Board Base Address	00	Specifies base address ($FFFFD2*XX*) in Short I/O for this board. Default = $00.
VMEbus Global Time Out Code	01	Controls VMEbus timeout when the MVME172LX 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 MVME172LXs.

The *MVME172 VME Embedded Controller Programmer's Reference Guide* describes the base addresses and the IP register settings. Refer to that manual for information on setting base addresses and register bits.

Note The IP2 ASIC on the MVME172LX supports up to four IndustryPack (IP) interfaces, designated IP_a through IP_d. The MVME172LX itself accommodates two IPs: IP_a and IP_b. In the following discussion, the segments applicable to IP_c and IP_d are not used in the MVME172LX.

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 MVME172LX VME embedded controller on a block diagram level. The *Description of Features* provides an overview of the MVME172LX, 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 MVME172LX blocks, including programmable registers in the ASICs and peripheral chips, can be found in the *MVME172 Embedded Controller Programmer's Reference Guide* (part number VME172A/PG). Refer to it for a functional description of the MVME172LX in greater depth.

Summary of Features

The following table summarizes the features of the MVME172LX VME embedded controller.

Table 4-1. MVME172LX Features

Feature	Description
Microprocessor	60MHz MC68060 or 64MHZ MC68LC060 processor
Form factor	6U VMEbus
Memory	4/8/16/32/64MB interleaved, ECC-protected DRAM or 4/8/16MB parity-protected DRAM
	128KB SRAM with battery backup
Flash memory	One Intel 28F016SA 2MB 8-bit Flash device
EPROMs	Two 32-pin JEDEC DIP PROM sockets with 256Kb/512Kb/1MB x8 selectable densities
Real-time clock	8KB NVRAM with RTC, battery backup, and watchdog function (SGS-Thomson M48T58)

Table 4-1. MVME172LX Features (Continued)

Feature	Description
Switches	Reset (**RST**) and Abort (**ABT**)
Status LEDs	Four: Board Fail (**FAIL**), **RUN**, System Controller (**SCON**), Fuses (**FUSES**)
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	Four EIA-232-D serial ports via RJ45 connectors on front panel
Ethernet I/O	Optional Ethernet transceiver interface via DB15 connector on front panel
IP interface	Two IndustryPack interface channels via 3M connectors behind front panel
SCSI I/O	Optional SCSI interface with DMA via 68-pin front panel connector
VMEbus interface	VMEbus system controller functions
	VMEbus-to-local-bus interface (A24/A32, D8/D16/D32/block transfer [D8/D16/D32/D64])
	Local-bus-to-VMEbus interface (A16/A24/A32, D8/D16/D32)
	VMEbus interrupter
	VMEbus interrupt handler
	Global Control/Status Register (GCSR) for interprocessor communications
	DMA for fast local memory/VMEbus transfers (A16/A24/A32, D16/D32/D64)

The MVME172LX is based on the MC68060/MC68LC060 microprocessor. Various versions of the MVME172LX have 4, 8, or 16MB of parity-protected DRAM or 4, 8, 16, 32, or 64MB of ECC-protected DRAM; 128KB of SRAM (with battery backup); time-of-day clock (with battery backup); an optional LAN Ethernet transceiver interface; four serial ports with EIA-232-D interface; six tick timers with watchdog timer(s); two EPROM sockets; 2MB Flash memory (one Flash device); two 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).

Input/Output (I/O) signals are routed through industry-standard connectors on the MVME172LX front panel; no adapter boards or transition modules are required. I/O connections include an optional 68-pin SCSI connector, an optional DB-15 Ethernet connector, and four 8-pin RJ-45 serial connectors on the front panel. In addition, the panel has cutouts for routing of flat cables to the optional IndustryPack modules.

The following ASICS are used on the MVME172LX:

- **VMEchip2.** (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, 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.

- **MC2chip.** Provides four tick timers, the interface to the LAN chip, SCSI chip, serial port chip, BBRAM, EPROM/Flash, parity-DRAM and SRAM.

- **MCECC memory controller.** Provides the programmable interface for the ECC-protected 16MB DRAM mezzanine board.

- **IndustryPack Interface Controller (IP2).** The IP2 provides control and status information for up to two single-wide IPs or one double-wide IP that can be plugged into the MVME172LX main board.

Block Diagram

The block diagram in Figure 4-1 on page 4-4 illustrates the MVME172LX's overall architecture.

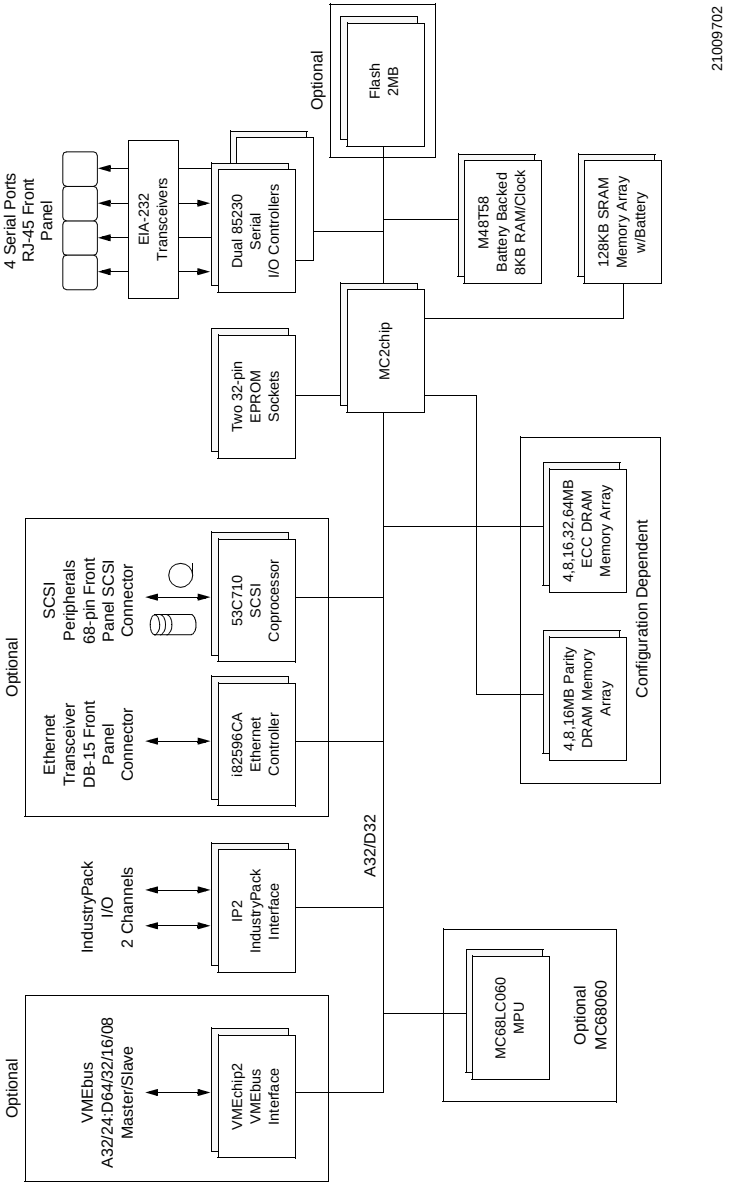


Figure 4-1. MVME172LX Block Diagram

Functional Description

This section contains a functional description of the major blocks on the MVME172LX.

Data Bus Structure

The local bus on the MVME172LX is a 32-bit synchronous bus that is based on the MC68060 bus, and which supports burst transfers and snooping. The various local bus master and slave devices use the local bus to communicate. The local bus is arbitrated by priority type; the priority of the local bus masters from highest to lowest is: 82596CA LAN, 53C710 SCSI, VMEbus, and MPU. As a general rule, any master can access any slave; not all combinations pass the common sense test, however. Refer to the *MVME172 VME Embedded Controller Programmer's Reference Guide* and to the user's guide for each device to determine its port size, data bus connection, and any restrictions that apply when accessing the device.

Microprocessor

The MVME172LX may be ordered with an MC68060 or MC68LC060 microprocessor.

The MC68060 has on-chip instruction and data caches and a floating point processor. (The floating point coprocessor is the major difference between the MC68060 and MC68LC060.) Refer to the M68060 user's manual for more information.

MC68*xx*060 Cache

The MVME172LX local bus masters (VMEchip2, MC68060/MC68LC060, 53C710 SCSI controller, and 82596CA Ethernet controller) have programmable control of the snoop/caching mode. The MVME172LX local bus slaves which support MC68060/MC68LC060 bus snooping are defined in the "Local Bus Memory Map" in the *MVME172 VME Embedded Controller Programmer's Reference Guide.* The Industry Pack DMA has jumpers to control the state of the snoop control signals.

Note The snoop capabilities of the MC68xx060 differ from those of the MC68xx040. Software must take these differences into consideration.

No-VMEbus-Interface Option

The MVME172LX may be operated as an embedded controller without the VMEbus interface. To support this feature, certain logic in the VMEchip2 has been duplicated in the MC2chip. This logic is inhibited in the MC2chip when the VMEchip2 is present. The enables for these functions are controlled by software and MC2chip hardware initialization.

Memory Options

The following memory options are available on the different versions of MVME172LX boards.

DRAM Options

The MVME172LX offers the following DRAM options: either 4MB, 8MB, or 16MB shared DRAM with programmable parity on a mezzanine module, or 4MB, 8MB, 16MB, 32MB, and 64MB ECC DRAM on a mezzanine board. The DRAM architecture for non-ECC memory is non-interleaved for 4MB or 8MB and interleaved for 16MB. Parity protection is enabled with interrupts or bus exception when a parity error is detected. DRAM performance is specified in the section on the DRAM Memory Controller in the MC2chip Programming Model in the *MVME172 VME Embedded Controller Programmer's Reference Guide.*

The DRAM map decoder may be programmed to accommodate different base address(es) and sizes of mezzanine boards. The onboard DRAM is disabled by a local bus reset and must be programmed before the DRAM may be accessed. Refer to the MC2chip and MCECC descriptions in the *MVME172 VME Embedded Controller Programmer's Reference Guide* for detailed programming information.

Most DRAM devices require a certain number of access cycles before the DRAMs are fully operational. Normally this requirement is met by the onboard refresh circuitry and normal DRAM initialization. However, software should insure a minimum of 10 initialization cycles are performed to each bank of RAM.

SRAM Options

The MVME172LX provides 128KB of 32-bit-wide onboard static RAM in a single non-interleaved architecture with onboard battery backup. The SRAM arrays are not parity protected.

The battery backup function for the onboard SRAM and the mezzanine 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 MVME172LX provides jumpers (on J14) that allow 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 MVME172LX. The SRAM may malfunction if inputs to the EM1275 are left unconnected.

The SRAM is controlled by the MC2chip, and the access time is programmable. Refer to the MC2chip description in the *MVME172 VME 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 MVME172LX 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 172Bug **PS** command (refer to *Debugger Commands* in Chapter 3) or its equivalent application-specific command. When restoring the board to service, execute the 172Bug **SET** command (**set *mmddyyhhmm***) after installation to restart the oscillator and initialize the clock.

The MVME172LX 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 J14 before installing the board. Refer to *SRAM Backup Power Source Select Header (J14)* on page 1-9 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 MVME172LX may be ordered with 2MB of Flash memory and two EPROM sockets ready for the installation of the EPROMs, which may be ordered separately. Flash memory is a single Intel 28F016SA device organized in a 2Mb x 8 configuration. The EPROM locations are standard JEDEC 32-pin DIP sockets that accommodate three jumper-selectable densities (256 Kb x 8; 512 Kb x 8; 1 Mb x8). A jumper setting (GPI4, pins 9-10 on J21), allows reset code to be fetched either from Flash memory (GPI4 installed) or from EPROMs (GPI4 removed).

Battery Backed Up RAM and Clock

An M48T58 RAM and clock chip is used on the MVME172LX. 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 automatically made. No interrupts are generated by the clock. Although the M48T58 is an 8-bit device, the interface provided by the MC2chip supports 8-, 16-, and 32-bit accesses to the M48T58. Refer to the MC2chip in the *MVME172 VME Embedded Controller Programmer's Reference Guide* and to the M48T58 data sheet for detailed programming 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 *MVME172 VME 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. The Abort switch interrupt is integrated into the MC2chip ASIC at location $FFF42043. The GPI inputs are integrated into the MC2chip ASIC at location $FFF4202C bits 23-16.

I/O Interfaces

The MVME172LX provides onboard I/O for many system applications. The I/O functions include serial ports and optional interfaces for IP modules, LAN Ethernet transceivers, and SCSI mass storage devices.

Serial Communications Interface

The MVME172LX uses two Zilog Z85230 serial port controllers to implement the four serial communications interfaces. Each interface supports CTS, DCD, RTS, and DTR control signals, as well as the TXD and RXD transmit/receive data signals. Because the serial clocks are omitted in the MVME172LX implementation, serial communications are strictly asynchronous. The MVME172LX hardware supports 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 MC2chip. Refer to the Z85230 data sheet listed in this chapter, and to the MC2chip Programming Model in the *MVME172 VME Embedded Controller Programmer's Reference Guide*, for information.

The Z85230s are interfaced as DTE (data terminal equipment) with EIA-232-D signal levels. The four serial ports are routed to four RJ-45 connectors on the MVME172LX front panel.

IP Interfaces

Up to two IP modules may be installed on the MVME172LX as an option. The interface between the IPs and MVME172LX is the IndustryPack Interface Controller (IP2) ASIC. Access to the IPs is provided by two 3M connectors located behind the MVME172LX front panel. Refer to the chapter on the IP2 in the *MVME172 VME Embedded Controller Programmer's Reference Guide* for detailed features of the IP interface.

Ethernet Interface

The MVME172LX uses the Intel 82596CA LAN coprocessor to implement the 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 MVME172LX that has 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 MVME172LX 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 configuration area of the BBRAM. 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 MVME172 debugger has the capability to retrieve or set the Ethernet address.

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

The Ethernet transceiver interface is located on the MVME172LX main board, and the industry standard connector is located on its front panel

Support functions for the 82596CA are provided by the MC2chip. Refer to the 82596CA user's guide and to the MC2chip description in the *MVME172 VME Embedded Controller Programmer's Reference Guide* for detailed programming information.

SCSI Interface

The MVME172LX supports 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 SCSI interface is implemented using the NCR 53C710 SCSI I/O controller.

Support functions for the 53C710 are provided by the MC2chip. Refer to the NCR 53C710 user's guide and to the MC2chip in the *MVME172 VME Embedded Controller Programmer's Reference Guide* for detailed programming information.

SCSI Termination

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

The MVME172LX main board provides terminators for the SCSI bus. The SCSI terminators are enabled/disabled by a jumper on header J12. If the SCSI bus ends at the MVME172LX, a jumper must be installed between J12 pins 1 and 2.

The **FUSES** LED (part of DS2) on the MVME172LX front panel monitors the SCSI bus TERMPWR signal in addition to LAN power and IndustryPack power; the **FUSES** LED illuminates when all fuses are operational. The fuses are solid-state devices that reset when the short is removed.

Because any device on the SCSI bus can provide TERMPWR, the **FUSES** LED does not directly indicate the condition of the fuse.

Local Resources

The MVME172LX 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

Four 32-bit programmable tick timers with 1µs resolution are provided in the MC2chip and two 32-bit programmable tick timers are provided in the optional VMEchip2. The tick timers may be programmed to generate periodic interrupts to the processor. Refer to the VMEchip2 and MC2chip in the *MVME172 VME Embedded Controller Programmer's Reference Guide* for detailed programming information.

Watchdog Timer

A watchdog timer is provided in both the MC2chip and the optional VMEchip2. The timers operate independently but in parallel. When the watchdog timers are enabled, they must be reset by software within the programmed time or they will time out. The watchdog timers may be programmed to generate a SYSRESET signal, local reset signal, or board fail signal if they time out. Refer to the VMEchip2 and the MC2chip in the *MVME172 VME Embedded Controller Programmer's Reference Guide* for detailed programming information.

The watchdog timer logic is duplicated in the VMEchip2 and MC2chip ASICs. Because the watchdog timer function in the VMEchip2 is a superset of that function in the MC2chip (system reset function), the timer in the VMEchip2 is used in all cases except for the version of the MVME172LX which does not include the VMEbus interface ("No VMEbus Interface" option).

Software-Programmable Hardware Interrupts

Eight software-programmable hardware interrupts are provided by the VMEchip2. These interrupts allow software to create a hardware interrupt. Refer to the VMEchip2 in the *MVME172 VME Embedded Controller Programmer's Reference Guide* for detailed programming information.

Local Bus Timeout

The MVME172LX provides timeout functions in the VMEchip2 and the MC2chip 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 infinite. 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 the MC2chip in the *MVME172 VME Embedded Controller Programmer's Reference Guide* for detailed programming information.

The MC2chip also provides local bus timeout logic for MVME172LXs without the optional VMEbus interface (i.e., without the VMEchip2).

The access timer logic is duplicated in the VMEchip2 and MC2chip ASICs. Because the local bus timer in the VMEchip2 can detect an offboard access and the MC2chip local bus timer cannot, the timer in the VMEchip2 is used in all cases except for the version of the MVME172LX which does not include the VMEbus interface ("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
MC68060/MC68LC060	4	Lowest

Connectors

The MVME172LX 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 are not used.

The MVME172LX has a 20-pin connector J2 mounted behind the front panel. When the MVME172LX 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.

Table 4-3. J2 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

The serial ports on the MVME172LX are connected to four 8-pin RJ-45 female connectors (J17) on the front panel. The two IPs connect to the MVME172LX by two pairs of 50-pin connectors. Two 50-pin connectors behind the front panel are for external connections to IP signals. The memory mezzanine board is plugged into two 100-pin connectors. The Ethernet LAN connector (J9) is a 15-pin socket connector mounted on the front panel. The SCSI connector (J23), is a 68-pin socket connector mounted on the front panel.

Pin Assignments

5

Connector Pin Assignments

This chapter summarizes the pin assignments for the following groups of interconnect signals on the MVME172LX:

Connector	Location	Table
Remote Reset connector	J2	Table 5-1
IndustryPack A and B connectors	J4/5/6, J3/7/8	Table 5-2
Ethernet port, DB15	J9	Table 5-3
Serial ports, RJ45	J17	Table 5-4
Memory Mezzanine connector 1	J22	Table 5-5
Memory Mezzanine connector 2	J15	Table 5-6
SCSI connector	J23	Table 5-7
VMEbus connector P1	P1	Table 5-8
VMEbus connector, 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 MVME172LX (available through your Motorola sales office).

Remote Reset Connector - J2

The MVME172LX has a 20-pin connector (J2) mounted behind the front panel. When the MVME172LX board is enclosed in a chassis and the front panel is not visible, this connector enables you to extend the reset, abort and LED functions to the control panel of the system, where they remain accessible.

Table 5-1. Remote Reset Connector J2 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

IndustryPack A and B Connectors - J4/5/6, J3/7/8

Up to two IndustryPack (IP) modules may be installed on the MVME172LX. For each IP module, there are two 50-pin plug connectors on the board: J5/6 for module A, J7/8 for module B. For external cabling to the IP modules, two 50-pin IDC connectors (J4 for module A, J3 for module B) are provided behind the MVME172LX front panel. The pin assignments are the same for both types of connector.

Table 5-2. 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

Ethernet Connector - J9

The MVME172LX's Ethernet interface is implemented with a DB15 connector located on the front panel of the board. The pin assignments for this connector are listed in the following table.

Table 5-3. DB15 Ethernet Connector Pin Assignments

1	GND	C+	2
3	T+	GND	4
5	R+	GND	6
7	No Connection	GND	8
9	C–	T–	10
11	GND	R–	12
13	+12V	GND	14
15	No Connection		

Serial Connector - J17

A four-segment RJ45 connector located on the front panel of the MVME172 provides the interface to the four asynchronous serial ports. The pin assignments for each segment in the connector are as follows.

Table 5-4. Serial Connector Pin Assignments

1	DCD*n*
2	RTS*n*
3	Ground
4	TXD*n*
5	RXD*n*
6	Ground
7	CTS*n*
8	DTR*n*

Memory Mezzanine Connector 1 - J22

Connector J22 is a standard double-row 100-pin socket connector mounted on the MVME172LX Embedded Controller PWB. It connects to a corresponding 100-pin plug connector on the ECC DRAM mezzanine board and (together with J15) carries the DRAM address, data, and control signals to and from the mezzanine board. The following table lists the pin assignments for J22.

Table 5-5. Mezzanine Connector J22 Pin Assignments

1	GND	LD1	2
3	LD0	LD3	4
5	LD2	GND	6
7	LD4	LD5	8
9	GND	LD7	10
11	LD6	LD9	12
13	LD8	GND	14
15	LD10	LD11	16
17	GND	LD13	18
19	LD12	LD15	20
21	LD14	GND	22
23	LD16	LD17	24
25	GND	LD19	26
27	LD18	LD21	28
29	LD20	GND	30
31	LD22	LD23	32
33	GND	LD25	34
35	LD24	LD27	36
37	LD26	GND	38
39	LD28	LD29	40
41	GND	LD30	42
43	DRAM_PD0	LD31	44

Table 5-5. Mezzanine Connector J22 Pin Assignments (Continued)

45	DRAM_PD1	GND	46
47	DRAM_PD2	MEZ0	48
49	DRAM_PD3	MEZ1	50
51	GND	MEZ2	52
53	RDRAM_0	RDRAM_5	54
55	RDRAM_1	RDRAM_6	56
57	RDRAM_2	GND	58
59	RDRAM_3	RDRAM_7	60
61	RDRAM_4	RDRAM_8	62
63	GND	RDRAM_9	64
65	RRCAS0*	RDRAMRAS*	66
67	RRCAS1*	GND	68
69	RRCAS2*	DRAMOE0*	70
71	RRCAS3*	DRAMOE1*	72
73	GND	DRAMOE2*	74
75	DRAMWELL*	DRAMOE3*	76
77	SRAMWELL*	GND	78
79	DRAMWELM*	DRAMWEUM*	80
81	SRAMWELM*	SRAMWEUM*	82
83	GND	DRAMWEUU*	84
85	SRAMCS0*	SRAMWEUU*	86
87	SRAMCS1*	GND	88
89	SRAMCS2*	SRAM_PA2	90
91	SRAMCS3*	SRAM_PA3	92
93	SRAMOE*	Reserved	94
95	GND	+5V STBY	96
97	+5V	+5V	98
99	+5V	+5V	100

Memory Mezzanine Connector 2 - J15

Connector J15 is a standard double-row 100-pin socket connector mounted on the MVME172LX Embedded Controller PWB. It connects to a corresponding 100-pin plug connector on the ECC DRAM mezzanine board and (together with J22) carries the DRAM address, data, and control signals to and from the mezzanine board. The following table lists the pin assignments for J15.

Table 5-6. Mezzanine Connector J15 Pin Assignments

1	GND	MPUCLK	2
3	GND	GND	4
5	PURESET*	LTS*	6
7	LBRESET*	GND	8
9	LRD	LTIP*	10
11	GND	SELECT	12
13	LLOCK*	LTA*	14
15	LSIZ0	GND	16
17	LSIZ1	LTEA*	18
19	GND	SRAMDIS*	20
21	LTM0	MIACKIN*	22
23	LTM1	GND	24
25	LTM2	LST0	26
27	GND	LST1	28
29	LSC0	MEZZIPL0*	30
31	LSC1	GND	32
33	LTT0	MEZZIPL1*	34
35	GND	MEZZIPL2*	36
37	LTT1	MEZZBR*	38
39	LMI*	GND	40
41	LOCKOK	MEZZBG*	42
43	GND	LBB*	44

Table 5-6. Mezzanine Connector J15 Pin Assignments (Continued)

45	PEIRQ*	SRAMSIZ0	46
47	Reserved	GND	48
49	LA0	SRAMSIZ1	50
51	GND	LA1	52
53	LA2	LA3	54
55	LA4	GND	56
57	LA6	LA5	58
59	GND	LA7	60
61	LA8	LA9	62
63	LA10	GND	64
65	LA12	LA11	66
67	GND	LA13	68
69	LA14	LA15	70
71	LA16	GND	72
73	LA18	LA17	74
75	GND	LA19	76
77	LA20	LA21	78
79	LA22	GND	80
81	LA24	LA23	82
83	GND	LA25	84
85	LA26	LA27	86
87	LA28	GND	88
89	LA30	LA31	90
91	Reserved	LA31	92
93	+12V	–12V	94
95	+12V	–12V	96
97	GND	GND	98
99	+5V	+5V	100

SCSI Connector - J23

Connector J23 is a standard 68-pin SCSI connector mounted on the front panel of the MVME172LX Embedded Controller. Table 5-7 lists the pin assignments for J23.

VMEbus Connectors (P1, P2)

Two three-row 96-pin DIN type connectors, P1 and P2, supply the interface between the base board and the VMEbus. P1 provides power and VME signals for 24-bit addressing and 16-bit data. Its pin assignments are set by the IEEE P1014-1987 VMEbus Specification. P2 Row B supplies the base board with power, with the upper 8 VMEbus address lines, and with an additional 16 VMEbus data lines. P2 rows A and C are not used in the MVME172LX implementation. The pin assignments for P1and P2 are listed in Table 5-8 and Table 5-9 respectively.

Table 5-7. SCSI Connector J23 Pin Assignments

1	GND	GND	2
3	GND	GND	4
5	GND	GND	6
7	GND	GND	8
9	GND	GND	10
11	GND	GND	12
13	GND	GND	14
15	GND	GND	16
17	+5.0V TERMPWR	+5.0V TERMPWR	18
19	No Connection	GND	20
21	GND	GND	22
23	GND	GND	24
25	GND	GND	26
27	GND	GND	28
29	GND	GND	30
31	GND	GND	32
33	GND	GND	34
35	DB*(12)	DB*(13)	36
37	DB*(14)	DB*(15)	38
39	DPH*	DB*(0)	40
41	DB*(1)	DB*(2)	42
43	DB*(3)	DB*(4)	44
45	DB*(5)	DB*(6)	46
47	DB*(7)	DBP*	48
49	GND	GND	50
51	+5.0V TERMPWR	+5.0V TERMPWR	52
53	No Connection	GND	54
55	ATN*	GND	56
57	BSY*	ACK*	58
59	RST*	MSG*	60
61	SEL*	DC*	62
63	REQ*	IO*	64
65	DB*(8)	DB*(9)	66
67	DB*(10)	DB*(11)	68

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	Not Used	+12V	31
32	+5V	+5V	+5V	32

Table 5-9. VMEbus Connector P2 Pin Assignment

	ROW A	ROW B	ROW C	
1	No Connection	+5V	No Connection	1
2	No Connection	GND	No Connection	2
3	No Connection	Not Used	No Connection	3
4	No Connection	VA24	No Connection	4
5	No Connection	VA25	No Connection	5
6	No Connection	VA26	No Connection	6
7	No Connection	VA27	No Connection	7
8	No Connection	VA28	No Connection	8
9	No Connection	VA29	No Connection	9
10	No Connection	VA30	No Connection	10
11	No Connection	VA31	No Connection	11
12	No Connection	GND	No Connection	12
13	No Connection	+5V	No Connection	13
14	No Connection	VD16	No Connection	14
15	No Connection	VD17	No Connection	15
16	No Connection	VD18	No Connection	16
17	No Connection	VD19	No Connection	17
18	No Connection	VD20	No Connection	18
19	No Connection	VD21	No Connection	19
20	No Connection	VD22	No Connection	20
21	No Connection	VD23	No Connection	21
22	No Connection	GND	No Connection	22
23	No Connection	VD24	No Connection	23
24	No Connection	VD25	No Connection	24
25	No Connection	VD26	No Connection	25
26	No Connection	VD27	No Connection	26
27	No Connection	VD28	No Connection	27
28	No Connection	VD29	No Connection	28
29	No Connection	VD30	No Connection	29
30	No Connection	VD31	No Connection	30
31	No Connection	GND	No Connection	31
32	No Connection	+5V	No Connection	32

Board Specifications

The following table lists the general specifications for the MVME172LX VME embedded controller. The subsequent sections detail cooling requirements and EMC regulatory compliance.

A complete functional description of the MVME172LX boards appears in Chapter 4. Specifications for the optional IndustryPack modules can be found in the documentation for those modules.

Table A-1. MVME172LX Specifications

Characteristics		Specifications
Power requirements (with EPROMs; 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 MVME172LX 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 MVME172LX may subject it to high temperatures.

The MVME172LX 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 MVME172LX. 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 MVME172-223 was developed in an MVME945 12-slot VME chassis. This board was loaded with one GreenSpring IP-Dual P/T module (position a) and one GreenSpring IP-488 module (position b). 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 MVME172LX 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 MVME172LX 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 MVME172LX 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 MVME172LX.
3. Use low-power IP modules only.

EMC Regulatory Compliance

The MVME172LX was tested in an EMC-compliant chassis and meets the requirements for Class B equipment. Compliance was achieved under the following conditions:

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

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

The MVME172LX is a board-level product and meant to be used in standard VME applications. As such, it is the responsibility of the OEM to meet the regulatory guidelines as determined by its application.

All external I/O connectors are shielded to aid in meeting EMC emissions standards. MVME172LX boards are tested in an MCG chassis for EMC compliance.

Solving Startup Problems

In the event of difficulty with your MVME172LX 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 MVME172LX Boards

Condition	Possible Problem	Try This:
I. Nothing works, no display on the terminal.	A. If the **RUN** (or **FUSES**) 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 MVME172LX 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 `172-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.	
IV. Debug prompt `172-Bug>` appears at powerup, but the board does not autoboot.	A. The initial debugger environment parameters may be set incorrectly.	1. Start the onboard calendar clock and timer. Type: **set *mmddyyhhmm* <CR>** where the characters indicate the month, day, year, hour, and minute. The date and time will be displayed. **Caution** — Performing the next step (**env;d**) will change some parameters that may affect your system's operation.
	B. There may be some fault in the board hardware.	

(continues>)

Table B-1. Troubleshooting MVME172LX 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: `172-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 MVME172LX 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 172Bug firmware supports 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 the 172Bug. 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
MVME172	$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

Disk/Tape Controller Modules Supported

The 172Bug firmware supports the following VMEbus disk/tape controller modules.

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 TRAP #15 calls .DSKRD or .DSKWR. If you use another controller of the same type, 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*	--	--	--
MVME328 - SCSI Controller	$06	$FFFF9000	$07	$FFFF9800
MVME328 - SCSI Controller	$16	$FFFF4800	$17	$FFFF5800
MVME328 - SCSI Controller	$18	$FFFF7000	$19	$FFFF7800

*If an MVME172LX with a SCSI port is used, then the MVME172LX has CLUN 0.

Disk/Tape Controller 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

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		
17	$FFFF5800	40 48 50 58 60 68 70	Same as above, but these are available only if the daughter card for the second SCSI channel is present.
18	$FFFF7000		
19	$FFFF7800		

IOT Command Parameters

The following table lists the proper **IOT** command parameters for floppy disks used with boards such as the MVME328 and MVME172LX.

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 notation 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
MVME172 VME Embedded Controller Programmer's Reference Guide	VME172A/PG
MVME172Bug Diagnostics User's Manual	V172DIAA/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

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 MC68060 Microprocessor User's 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 M68060UM M68040UM
82596CA Local Area Network Coprocessor Data Sheet 82596CA Local Area Network Coprocessor User's Manual 28F016SA Flash Memory Data Sheet Intel Corporation Web: http://developer.intel.com/design	290218 296853 209435
SYM 53C710 (was NCR 53C710) SCSI I/O Processor Data Manual SYM 53C710 (was NCR 53C710) SCSI I/O Processor Programmer's Guide Symbios Logic Inc. 1731 Technology Drive, Suite 600 San Jose, CA 95110 NCR Managed Services Center — Telephone: 1-800-262-7782 Web: http://www.symbios.com/techsupport	NCR53C710DM NCR53C710PG
M48T58(B) TIMEKEEPER™ and 8K x 8 Zeropower™ RAM Data Sheet SGS-Thomson Microelectronics Group Marketing Headquarters (or nearest Sales Office) 1000 East Bell Road Phoenix, Arizona 85022 Telephone: (602) 867-6100 Web: http://www.st.com/stonline/books	M48T58

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

172Bug
- firmware 3-9
- implementation 3-3
- overview 3-1
- stack space 3-4

27C040 EPROM 3-3
53C710 SCSI controller 4-12
82596CA LAN coprocessor 4-12

A

abort process 2-12
address ranges, EPROM 1-13
address/data configurations 1-19
addressing modes 1-19
altitude (operating) A-1
ambient air temperature A-2
arbitration priority 4-15
arguments, firmware command 3-5
autoboot process 2-8

B

backplane jumpers 1-19
batteries 4-8
BBRAM (battery-backed-up RAM)
- and clock 3-9, 4-10

BG (bus grant) signal 1-19
block diagram 4-3
board
- architecture 4-1
- configuration 1-4
- dimensions A-1
- features 4-1

board *continued*
- installation 1-18
- preparation 1-1

booting the system 2-8, 2-9, 2-10
BREAK key 2-12
break process 2-12
bus grant (BG) signal 1-19

C

C programming language 3-3
cable connections 1-19
cables, I/O ports A-4
chassis grounding A-4
Clear To Send (CTS) signal 2-7
CLUN (controller LUN) C-1, D-2
command identifier, firmware 3-5
command line 3-5
commands, firmware
- CNFG (Configure Board Information Block) 3-9
- ENV (Set Bug/Operating System Environment) 3-11

conductive chassis rails A-4
configurable items, MVME172 board 1-4
Configure Board Information Block (CNFG) firmware command 3-9
configuring
- 172Bug parameters 3-11
- default disk/tape controller D-2
- hardware 2-5
- IndustryPack modules 3-19
- IP base addresses 3-19
- VMEbus interface 3-16

connector pin assignments 5-1
connectors 4-15
console port 2-6
control/status registers 1-20
controller data D-1
controller LUN (CLUN) C-1, D-2
cooling requirements A-2
CSR bit IP32 1-8
CTS (Clear To Send) signal 2-7

D

data bus structure, MVME172 4-5
data sheets E-2
data terminal equipment (DTE) 4-11
date and time, setting 2-7, B-2
debugger
    commands 3-6
    firmware (172Bug) 3-9
    prompt 3-5
default baud rate 2-6
device LUN (DLUN) C-1, D-2
diagnostic facilities 2-13
dimensions, base board A-1
direct access device D-2
directories, switching 2-13
disk/tape controller data D-1
disk/tape controller default configurations D-2
DLUN (device LUN) C-1, D-2
DRAM (dynamic RAM)
    base address 1-19
    options 4-6
DTE (data terminal equipment) 4-11

E

ECC DRAM 1-16
EIA-232-D ports 2-6
EMC emissions standards A-4
EMC regulatory compliance A-4
ENV command parameters 3-11
environmental parameters 3-9
EPROM 3-3
EPROM address ranges 1-13
EPROM and Flash memory 4-9
EPROM sockets 1-11, 4-9
EPROM/Flash mapping
    1M x 8 EPROMs 1-14
    1M x 8 EPROMs, on-board Flash disabled 1-14
    256K x 8 EPROMs 1-13
    512K x 8 EPROMs 1-13
EPROM/Flash selection 1-15
ESD (electrostatic discharge), precautions against 1-3
Ethernet
    interface 4-12, C-1
    signals 5-4
    station address 4-12
extended addressing 1-19

F

features, hardware 4-1
firmware 2-2
firmware directories 3-5
firmware documentation E-1
Flash memory 1-11, 3-3, 4-9
Flash write-protect header (J23) 1-10
flexible diskette controller D-2
floppy disk command parameters D-3
forced air cooling A-2
front panel switches and indicators 2-1
functional description 4-1

G

GCSR (Global Control and Status Registers) 1-20
GCSR GPCSR0 address 3-14
general-purpose readable jumpers (J21) 1-14
global bus timeout 1-20
Global Control and Status Registers (GCSR) 1-20
grounding A-4
grounding strap 1-3

H

handshaking 2-7
hardware
- features 4-1
- initialization 2-3
- interrupts 4-14
- preparation 1-1

high-temperature operation A-3
humidity, relative A-1

I

I/O interfaces 4-10
IACK (interrupt acknowledge) signal 1-19
indicators 2-1
IndustryPack (IP) interfaces 4-11
IndustryPack modules
- base addresses of 3-19
- configuration 3-21
- configuring 3-19
- general control register 3-20
- installation 1-17
- memory size 3-20

IndustryPack signals 5-2
initial conditions 2-4
installation considerations 1-20
installation instructions 1-17
interconnect signals 5-1
interrupt acknowledge signal (IACK) 1-19
interrupt control registers 3-21
Interrupt Stack Pointer (ISP) 3-4
interrupts 4-14
IOT command parameters for supported
  - floppy types D-3

IP bus clock header (J11) 1-7
IP bus strobe signal 1-10
IP installation 1-17
IP32 CSR bit 1-8
ISP (Interrupt Stack Pointer) 3-4

J

jumper headers
- J1 (system controller selection) 1-7
- J11 (IP bus clock) 1-7
- J12 (SCSI termination) 1-8
- J14 (SRAM backup power) 1-9
- J18 (IP bus strobe) 1-10
- J19 (IP DMA snoop control) 1-11
- J20 (EPROM/Flash configuration) 1-11
- J21 (software-readable headers) 2-5
- J23 (Flash write protection) 1-10

jumper headers, location of 1-4
jumpers, backplane 1-19

L

LAN interface 4-12
LEDs (light-emitting diodes) 2-1
local
- bus 4-14
- bus arbiter 4-15
- bus timeout 4-14
- resources for the processor 4-13

location monitors, processor 1-20

M

MC2chip ASIC 1-13
MC2chip registers 1-14
MC68060 or MC68LC060 MPU 4-5
MC68xx060 cache memory 4-5
memory boards 1-16
memory mezzanine stacking options 1-16
memory options 4-6
memory requirements 3-3
mezzanine boards 1-16
mezzanine signals 5-5, 5-7
microprocessor 4-5
MPU 4-5
MVME172
- as network controller C-1
- board specifications A-1
- cooling requirements A-2
- EMC regulatory compliance A-4

MVME172 *continued*
features 4-1
installation 1-18
regulatory compliance A-4
MVME172Bug documentation E-1
MVME328 SCSI Controller D-1
MVME374 network controller C-1
MVME376 network controller C-1

N

network boot process 2-10
network controller modules C-1
non-volatile RAM (NVRAM) 3-9, 3-11
no-VMEbus-interface option 4-6

O

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

P

P1 and P2 connectors 5-9
parameters, ENV command 3-11
parity DRAM 1-16
pin assignments, connector 5-1
port number(s) 3-5
power requirements A-1
powering up the board 2-1
preparing the board 1-1
processor location monitors 1-20
programmable tick timers 4-13

R

regulatory compliance A-4
regulatory guidelines A-4
related specifications E-3
relative humidity A-1
remote control/status connector 5-2
remote panel connector 4-15
reset code source 1-13
resetting the system 2-1, 2-11
restarting the system 2-11
RF emissions 1-18, A-4
ROMboot process 2-9

S

safety procedures 1-4
SCSI
controller (53C710) 4-12
interface 4-12
signals 5-9
termination 4-13
terminator configuration 1-8, 4-13
SD command 2-13
serial
communication controllers (SCCs) 2-7
communication interface 1-21, 4-11, 5-4
communication parameters 2-4
serial connections
DB25-DCE-to-RJ45 1-22
DB25-DTE-to-RJ45 1-22
serial ports 2-6
Set Bug/Operating System Environment (ENV) firmware command 3-11
setting date and time 2-7, B-2
shielded cables A-4
size of base board A-1
slave address decoders, VMEbus 3-16
snoop control 1-11
software-programmable hardware interrupts 4-14
software-readable jumpers (J21) 1-14
specifications
MVME172 A-1
related E-3
SRAM (static RAM)
backup power source select header (J14) 1-9
battery options 4-8
options 4-7
stacking mezzanine boards 1-16
startup overview 1-1
static variable space 3-4
storage temperature A-1

INDEX

switches 2-1, 4-5
switching directories 2-13
system considerations 1-19
system console setup 2-6
system controller function 2-5
system controller select header (J1) 1-7
System Fail (SYSFAIL*) signal 2-9
system reset 2-11
system startup 2-3

T

temperature
    operating A-1
    storage A-1
terminal
    configuration 2-4
    input/output control 3-5
thermal sensing header (J13) 1-8
tick timers 4-13
timeout
    global bus 1-20
    local bus 4-14
troubleshooting procedures B-1
types of reset 2-11

U

user-definable jumpers 1-15

V

vibration (operating) A-1
VMEbus interface 4-10
VMEbus signals 5-9
VMEchip2 ASIC 4-10

W

watchdog timer 4-14

X

XON/XOFF handshaking 2-7

Z

Z85230 serial communication controllers
        (SCCs) 2-7

MVME172LX VME Embedded Controller Installation and Use