Getting Started With Your VME-MC6000 and the NI-VXI Software for AIX

320398-01   February 1993

Limited Warranty

The National Instruments MXIbus boards are warranted against defects in materials and workmanship for a period of one year from the date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

Trademarks

NI-VXI^{TM} is a trademark of National Instruments Corporation.

Product and company names listed are trademarks or trade names of their respective companies.

FCC/DOC Radio Frequency Interference Compliance

This equipment generates and uses radio frequency energy and, if not installed and used in strict accordance with the instructions in this manual, may cause interference to radio and television reception. This equipment has been tested and found to comply with the following two regulatory agencies:

- The limits for a Class A computing device, in accordance with the specifications in Part 15 of U.S. Federal Communications Commission (FCC) Rules. Operation is subject to the following two conditions:

    - This device may not cause harmful interference.

    - This device must accept any interference received, including interference that may cause undesired operation.

- The limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications (DOC).

These regulations are designed to provide reasonable protection against interference from the equipment to radio and television reception in commercial areas.

There is no guarantee that interference will not occur in a particular installation. However, the chances of interference are much less if the equipment is used according to this instruction manual.

If the equipment does cause interference to radio or television reception, which can be determined by turning the equipment on and off, one or more of the following suggestions may reduce or eliminate the problem.

- Operate the equipment and the receiver on different branches of your AC electrical system.

- Move the equipment away from the receiver with which it is interfering.

- Relocate the equipment with respect to the receiver.

- Reorient the receiver's antenna.

- Be sure that the equipment is plugged into a grounded outlet and that the grounding has not been defeated with a cheater plug.

Notice to user: Changes or modifications not expressly approved by National Instruments could void the user's authority to operate the equipment under the FCC Rules.

If necessary, consult National Instruments or an experienced radio/television technician for additional suggestions. The following booklet prepared by the FCC may also be helpful: *How to Identify and Resolve Radio-TV Interference Problems.* This booklet is available from the U.S. Government Printing Office, Washington, DC 20402, Stock Number 004-000-00345-4.

Contents

Appendix A
Specifications..A-1

Appendix B
Customer Communication..B-1

Glossary...Glossary-1

Figures

Tables

About This Manual

This manual contains instructions for installing and configuring the National Instruments VME-MC6000 interface kit. This manual is meant to be used with the *NI-VXI C Software Reference Manual for VME* (part number 320389-01) and the *NI-VXI Text Utilities Reference Manual* (part number 320321-01).

Organization of This Manual

Getting Started with Your VME-MC6000 and the NI-VXI Software for AIX is organized as follows:

- Chapter 1, *Introduction*, describes the VME-MC6000 interface kit, contains a picture of the VME-MC6000 interface kit, and lists the contents and optional equipment of your kit.

- Chapter 2, *Hardware Configuration and Installation*, contains the instructions to configure and install the VME-MC6000 interface kit.

- Chapter 3, *NI-VXI Software Installation and Configuration*, lists all the programs and files located on the NI-VXI distribution diskette, contains instructions for installing and configuring the hardware and the NI-VXI software for AIX for VME, and contains a description of the `vxitedit` configuration editor.

- Appendix A, *Specifications*, lists various module specifications of the MC-MXI and VME-MXI, such as physical dimensions and power requirements.

- Appendix B, *Customer Communication*, contains forms you can use to request help from National Instruments or to comment on our products and manuals.

- The *Glossary* contains an alphabetical list and description of terms used in this manual, including abbreviations, acronyms, metric prefixes, and symbols.

Conventions Used in This Manual

Throughout this manual, the following conventions are used to distinguish elements of text:

italic	Italic text denotes emphasis, a cross reference, menu items, or an introduction to a key concept.
`monospace`	Text in this font denotes text or characters that are to be literally input from the keyboard, the proper names of directories, programs, commands, variables, filenames, and extensions, and statements and comments taken from program code.

Abbreviations, acronyms, metric prefixes, mnemonics, symbols, and terms are listed in the *Glossary*.

How to Use the Documentation Set

Begin by reading this manual to guide you through the installation and configuration of the hardware and software. The software configuration requires that you first complete the installation and configuration of the hardware.

When you are familiar with the material in this manual, you can begin to use the *NI-VXI C Software Reference Manual for VME*. Chapter 1, *Introduction to NI-VXI for VME*, and Chapter 2, *Introduction to the NI-VXI Functions for VME,* present the concepts of using the NI-VXI software for programming VME systems and prepare you for detailed explanations of the NI-VXI functions. Study the descriptions of each function given in Chapters 3 through 9 to fully understand the purpose and syntax of each function. Refer to the *NI-VXI Text Utilities Reference Manual* to learn more about the `victext` and `vxitedit` programs.

Related Documentation

The following documents contain information that you may find helpful as you read this manual:

- *IEEE Standard for a Versatile Backplane Bus: VMEbus*, ANSI/IEEE Standard 1014-1987

- *NI-VXI Software Reference Manual for C* (for VXI systems, part number 320307-01)

- *VME-MXI User Manual* (part number 320330-01)

- VXI-1, *VXIbus System Specification*, Rev. 1.4, VXIbus Consortium (available from National Instruments Corporation, part number 350083-01)

- VXI-6, *VXIbus Mainframe Extender Specification*, Rev. 1.0, VXIbus Consortium (available from National Instruments Corporation, part number 340258-01)

Customer Communication

National Instruments wants to receive your comments on our products and manuals. We are interested in the applications you develop using our products, and we want to help if you have problems with them. To make it easy for you to contact us, this manual contains comment and configuration forms for you to complete. These forms are in Appendix B, *Customer Communication*, at the end of this manual.

Chapter 1
Introduction

This chapter describes the VME-MC6000 interface kit, contains a picture of the VME-MC6000 interface kit, and lists the contents of your kit and optional equipment.

The VME-MC6000 interface kit links any RISC System/6000 computer system (hereafter referred to as the RS/6000) directly to the VMEbus. An RS/6000 equipped with a VME-MC6000 can control the VMEbus directly. The VME-MC6000 makes the RS/6000 perform as though it were an embedded CPU plugged directly into the VME backplane, or as though it has internal VME slots for plug-in VME boards. The driver software for the VME boards must, of course, be ported to run on the RS/6000 with the VME-MC6000 kit.

Figure 1-1 shows the VME-MC6000 interface kit.
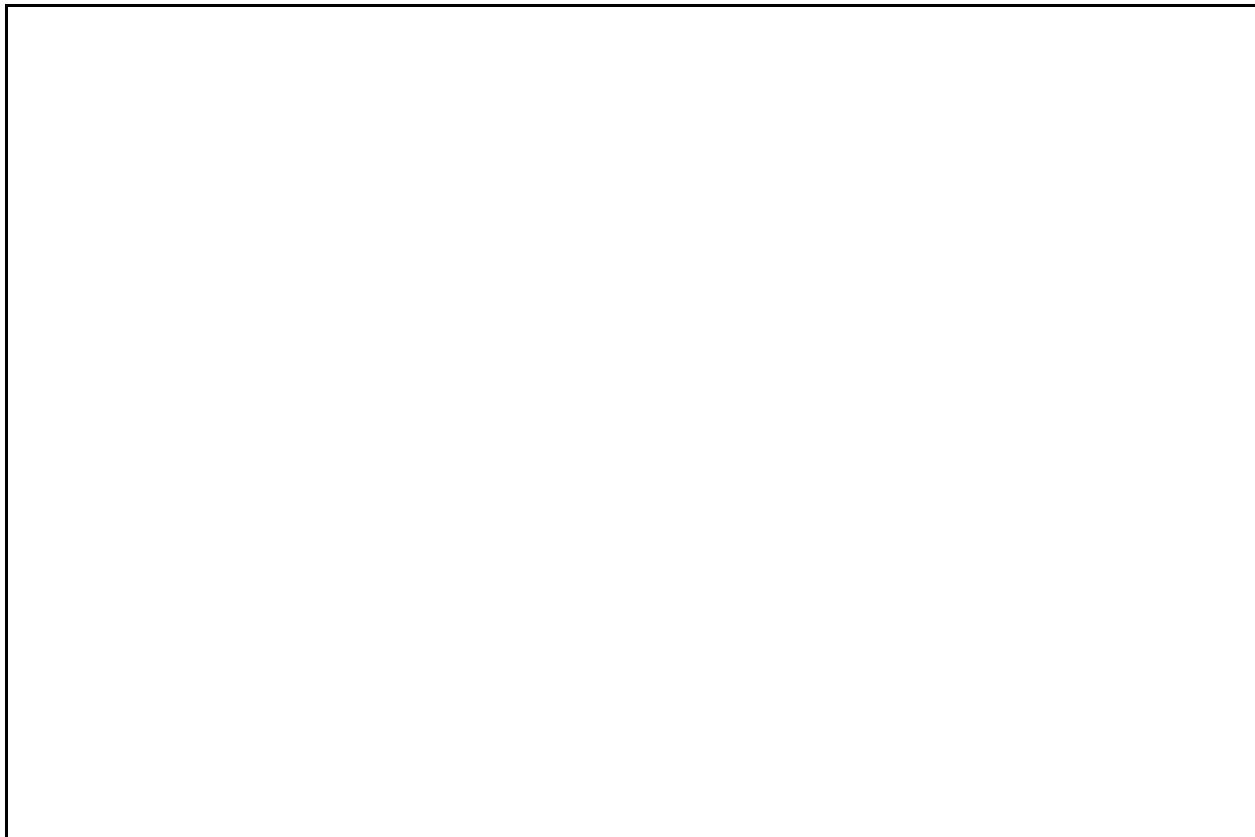
Figure 1-1. VME-MC6000 Interface Kit

What Your Kit Should Contain

Your kit should contain the following components.

Component	Part Number
MC-MXI Interface Board	181075-01
Standard VME-MXI Interface Module or Enhanced VME-MXI Interface Module with INTX	180960-01 180960-02
NI-VXI Distribution Disk for MC-MXI and AIX	422821-52
2 m Type M1 MXIbus Cable *	180758-02
Getting Started with Your VME-MC6000 and the NI-VXI Software for AIX	320398-01
NI-VXI C Software Reference Manual for VME	320389-01
NI-VXI Text Utilities Reference Manual	320321-01

* The 2 m Type M1 MXIbus cable is included in your kit unless you specified otherwise in your order. You may have ordered your kit without this cable so that you could order a different type or length of MXIbus cable. Refer to the *Optional Equipment* section on the following page.

Make sure each of these items is in your kit. If any item is missing, contact National Instruments.

Optional Equipment

Equipment	Part Number
Type M1 MXIbus Cables Straight Point-to-Point Connectors: – 1 m – 2 m – 4 m – 8 m – 20 m	 180758-01 180758-02 180758-04 180758-08 180758-20
Type M2 MXIbus Cables Straight Point-to-Right Angle Daisy-Chain Connectors: – 1 m – 2 m – 4 m – 8 m – 20 m	 180760-01 180760-02 180760-04 180760-08 180760-20
Type M3 MXIbus Cables Right Angle Point-to-Right Angle Daisy-Chain Connectors: – 1 m – 2 m – 4 m – 8 m – 20 m	 180761-01 180761-02 180761-04 180761-08 180761-20
MXIbus Terminating Pac (External)	180780-01
VXI-MXI Standard VXIbus Mainframe Extender Kit	776310-01
VXI-MXI Enhanced VXIbus Mainframe Extender Kit	776310-02
VME-MXI Standard VMEbus Chassis Extender Kit	776434-01
VME-MXI Enhanced VMEbus Chassis Extender Kit	776434-02

Chapter 2
Hardware Configuration and Installation

This chapter contains the instructions to configure and install the VME-MC6000 interface kit. The instructions are given in the order that you should perform them. A summary of the steps is as follows:

1. Unpack the MC-MXI and VME-MXI hardware.

2. Install the MC-MXI hardware.

3. Configure the VME-MXI hardware.

4. Install the VME-MXI hardware.

5. Connect the MXIbus cable.

Step 1. Unpack the MC-MXI and VME-MXI

Follow these steps when unpacking your MC-MXI board and VME-MXI module:

1. Before attempting to configure or install the MC-MXI and VME-MXI, inspect the shipping container and its contents for damage. If damage appears to have been caused in shipment, file a claim with the carrier. Retain the packing material for possible inspection and/or for reshipment.

2. Verify that the pieces contained in the package you received match the kit parts list. *Do not* remove the boards from their plastic bags at this point.

3. Your MC-MXI board and VME-MXI module are shipped packaged in antistatic plastic bags to prevent electrostatic damage. Some of the circuitry on the MC-MXI and VME-MXI use CMOS technology and can be damaged by electrostatic discharge. Before removing the boards from their antistatic bags, touch the bags to a metal part of your computer chassis.

4. As you remove the MC-MXI and VME-MXI from their bags, be sure to handle them only by their edges. Avoid touching any of the IC components or connectors. Inspect them for loose components or any other sign of damage. Notify National Instruments if either board appears damaged in any way. *Do not* install equipment that appears to be damaged.

Step 2. Install the MC-MXI Hardware

Before attempting to install the MC-MXI, notice that some MXIbus cable connector hoods are slightly wider than most standard connector hoods and might interfere with other cables installed in adjacent Micro Channel slots. Normally, this will only be a problem if the cable connector hoods for the adjacent slots are also oversized. When choosing a slot in which to install the MC-MXI, verify that the MXIbus cable connector will not interfere with cables and connectors in other slots. If necessary, reposition the boards in the system to prevent cabling conflicts. It may also help to install the MC-MXI in one of the end slots so that you will only have to contend with the cable connectors of one other board.

If you cannot configure the MC-MXI to co-exist in an existing Micro Channel system by repositioning the boards, you can use one of the MXIbus cable options with a straight-point connector hood on the cable end that attaches to the MC-MXI. The straight-point connector hood is narrower than the MXIbus dual-connector arrangement and provides an easier fit for many system configurations. However, this approach requires that the MC-MXI be the first device in the MXIbus daisy-chain because a cable with a straight-point connector end cannot accept another MXIbus cable to propagate the bus. Remember that the first device in the MXIbus daisy-chain must also be configured as the MXIbus System Controller.

The following instructions are general installation instructions. Consult the user or technical reference manual of your computer for specific instructions and warnings.

1. Plug in your RS/6000 computer before installing the MC-MXI. The plug grounds the computer and protects it from electrical damage while you are setting up.

   Warning: To protect both yourself and the computer from electrical hazards, the computer should remain off until you are finished installing the board.

2. Remove the top cover or access port to the Micro Channel I/O bus.

3. Select a 32-bit full length expansion slot. These slots have a longer card edge expansion connector than those found on 16-bit slots. Most Micro Channel computers have an assortment of both 16-bit and 32-bit expansion slots. A 32-bit board, such as the MC-MXI, cannot be installed in a 16-bit expansion slot.

4. Locate the metal bracket that covers the cut-out in the back panel of the Micro Channel chassis for the slot you have selected. Loosen the bracket retaining thumbscrew and remove the bracket by lifting it out of the top of the slot.

5. Before picking up the MC-MXI, touch the metal part of the power supply case inside the computer to discharge any static electricity that might be on your clothes or body.

6. Line up the MC-MXI with the MXIbus connector near the cut-out on the back panel and the other card edge lined up with the respective slot guide. Slowly push down on the middle of the MC-MXI until its connector is resting on the expansion slot receptacle. Using slow, evenly distributed pressure, press the MC-MXI straight down until it seats in the expansion slot.

7. Tighten the bracket retaining thumbscrew to secure the MC-MXI to the back panel rail.

8. Check the installation.

9. Replace the computer cover and restore power to the RS/6000 computer.

Step 3. Configure the VME-MXI Hardware

Figure 2-1 shows the location and factory default settings of the configuration switches and jumpers for a VME-MXI without the INTX daughter card option. The remainder of this chapter describes only those options that are user-configurable. Do not change the default settings of other jumpers and switches on the VME-MXI board unless you will be installing more than one VME-MXI in the same chassis. If this is the case, refer to the *VME-MXI User Manual* that came with your additional VME-MXI interface for more information.
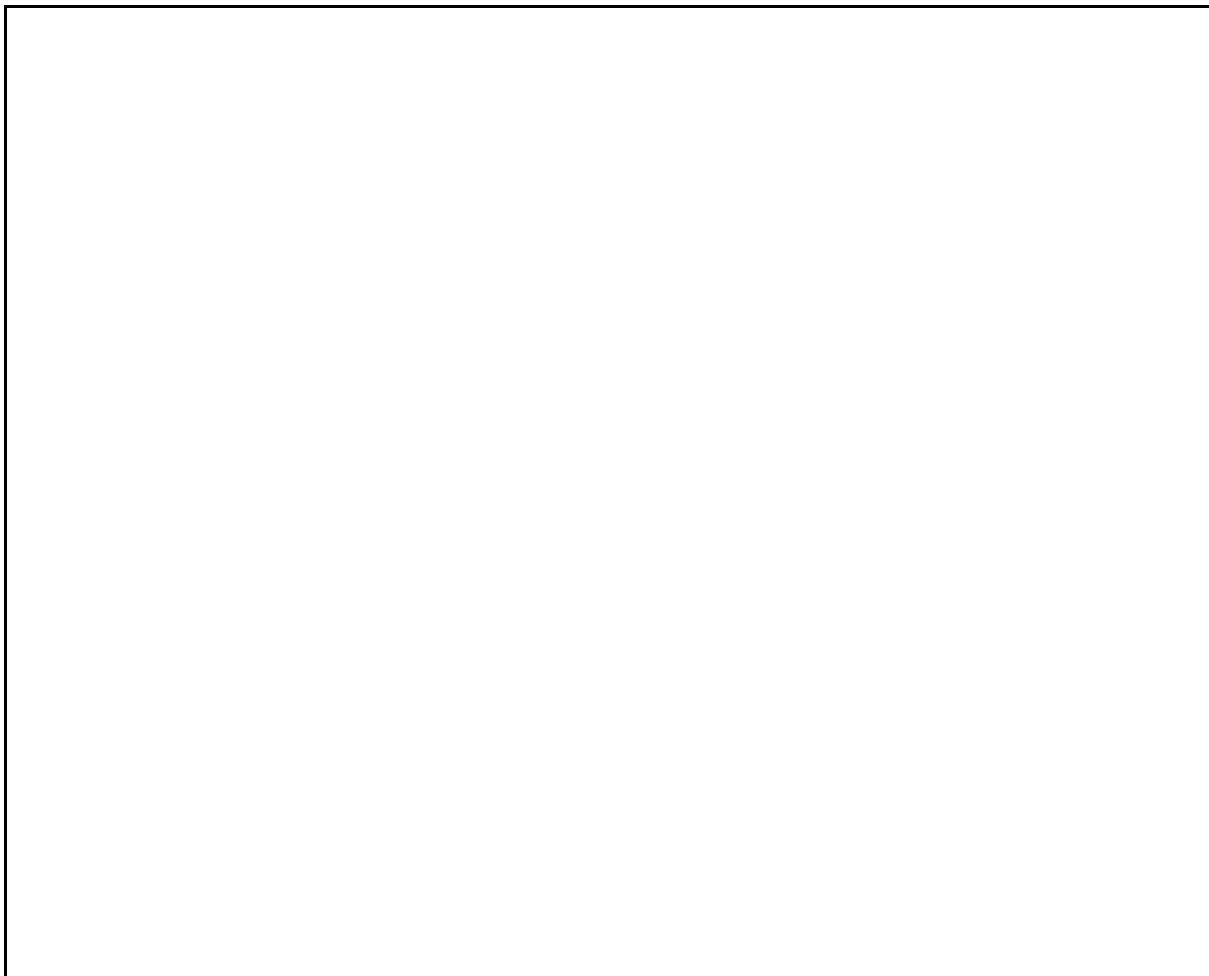
Figure 2-1. VME-MXI Parts Locator Diagram

Figure 2-2 shows the location and factory default settings of the configuration switches and jumpers for a VME-MXI with the INTX daughter card option. The options described in the rest of this chapter apply to VME-MXI modules with or without the INTX option. For more information about configuring modules with the INTX option, refer to the *VME-MXI User Manual*.
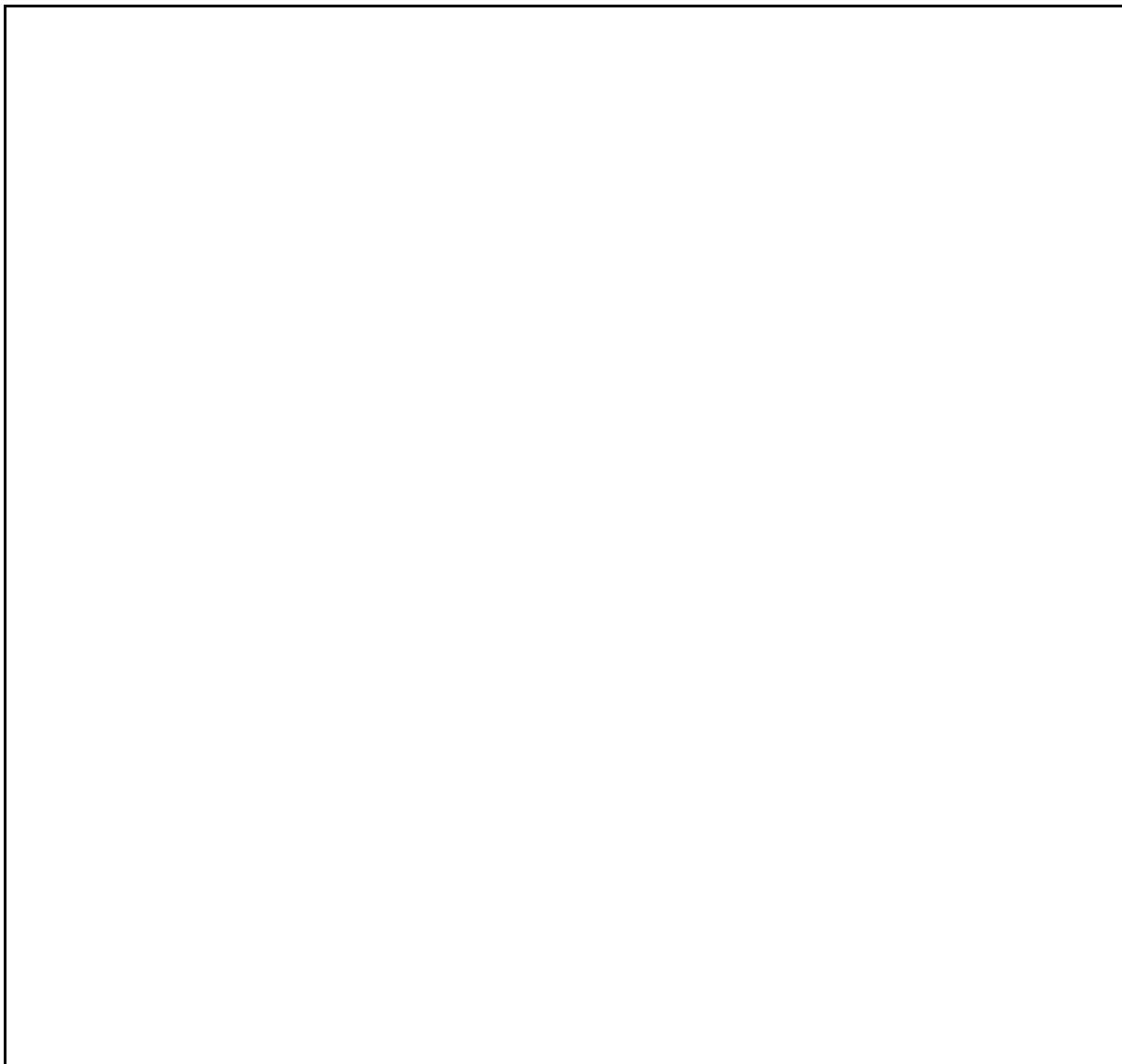
Figure 2-2. VME-MXI with INTX Parts Locator Diagram

Front Panel Features

The VME-MXI has the following front panel features.

- Three front panel LEDs
    - *FAILED* LED indicates that the VMEbus SYSFAIL line is asserted.
    - *VME ACCESS* LED indicates when the VME-MXI is accessed from the VMEbus.
    - *MXI ACCESS* LED indicates when the VME-MXI is accessed from the MXIbus.
- MXIbus connector
- System reset pushbutton
- INTX connector (if your VME-MXI includes the INTX daughter card option)

VMEbus System Controller

The VME-MXI is shipped from the factory configured to be installed in Slot 1 of the VMEbus chassis as the VMEbus System Controller. If another device is already in Slot 1, you must decide which device will be the VMEbus System Controller and reconfigure the other device as non-VMEbus System Controller.

When the VME-MXI is the VMEbus System Controller, it has the VMEbus Data Transfer Bus Arbiter capability (PRI ARBITER) and it drives the 16-MHz VMEbus system clock. The VMEbus Data Transfer Bus Arbiter circuitry accepts bus requests on all four VMEbus request levels, prioritizes the requests, and grants the bus to the highest priority requester. The VMEbus system clock is driven by an onboard 16-MHz oscillator with a 50% ±5% duty cycle.

If you want to install the VME-MXI into a VXIbus mainframe, you can install it in any slot except Slot 0. The VME-MXI has VXIbus configuration registers, which makes it compatible with the VXIbus specification. However, the VME-MXI does not have the CLK10 and MODID circuitry required by a Slot 0 VXIbus device. Therefore, configure the VME-MXI as a non-VMEbus System Controller and install it in any other slot.

Figure 2-3(a) shows the default configuration setting for the VME-MXI installed as the VMEbus System Controller. To configure the VME-MXI as a non-VMEbus System Controller, change slide switch S5 as depicted in Figure 2-3(b).
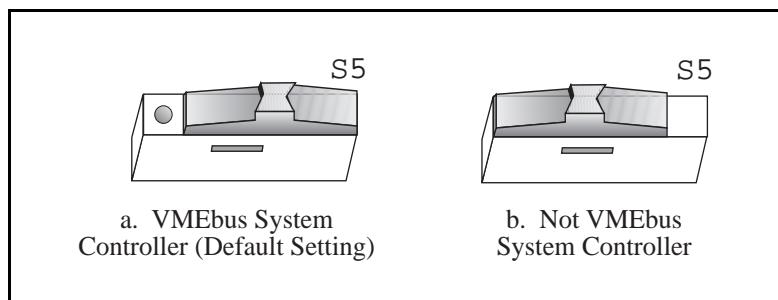
Figure 2-3. VMEbus System Controller Selection

VME Address

The MC-MXI and the VME-MXI use VME short (A16) address space for their configuration, communication, and status registers. Because both boards are not only VMEbus-compatible but are also compatible with the VXIbus specification, each board has an associated logical address. The logical address assigns a unique 64-byte block of address space in the upper 16 kilobytes of A16 space to each board. The starting address of a device's address space can be calculated using the following formula:

$$\text{starting address (A16 space)} = (logical\ address * 40\text{h}) + \text{C000h}$$

The MC-MXI is responsible for configuring the system upon power-up or reset and is referred to as the System Resource Manager. Because the System Resource Manager is defined to have a logical address of 0, the MC-MXI will use A16 address space from C000h through C03Fh. The logical address of the MC-MXI is software configurable, but should not be changed when used with a VME system.

The VME-MXI is factory-configured with a logical address of 1, meaning that it occupies A16 space in the range of C040h through C07Fh. The logical address of the VME-MXI is configurable via an onboard switch, but should not be changed unless you are using multiple VME-MXIs in your system. If this is the case, refer to the *VME-MXI User Manual* that came with your additional VME-MXIs for more information.

Ensure that no other VMEbus devices in your system occupy address space used by the MC-MXI or the VME-MXI. If they do, change the other VME devices so that they occupy unique portions of address space. In addition, if you are installing the VME-MXI in a VXIbus system, make sure that no other VXIbus devices are statically configured at either Logical Address 0 or 1.

VMEbus Request Level Selection

The VME-MXI uses one of the four VMEbus request levels to request use of the VME Data Transfer Bus (DTB). The VME-MXI requests use of the DTB whenever an external MXIbus device, such as an RS/6000 with an MC-MXI interface, attempts a transfer that maps into the VMEbus chassis.

The VME-MXI is factory configured to use VMEbus request level 3. This is the highest priority request level and is suitable for most VMEbus systems. However, you can change the VME-MXI to use any of the other three request levels (0, 1, or 2) by changing the jumper configuration of the three pin arrays at locations W2, W3, W4, W5, and W6. You may want to change request levels to change the priority of the VME-MXI request signal. For more information, refer to the VMEbus specification.

To change the VMEbus request level of the VME-MXI, rearrange the jumpers on the pin arrays as shown in Figure 2-4.
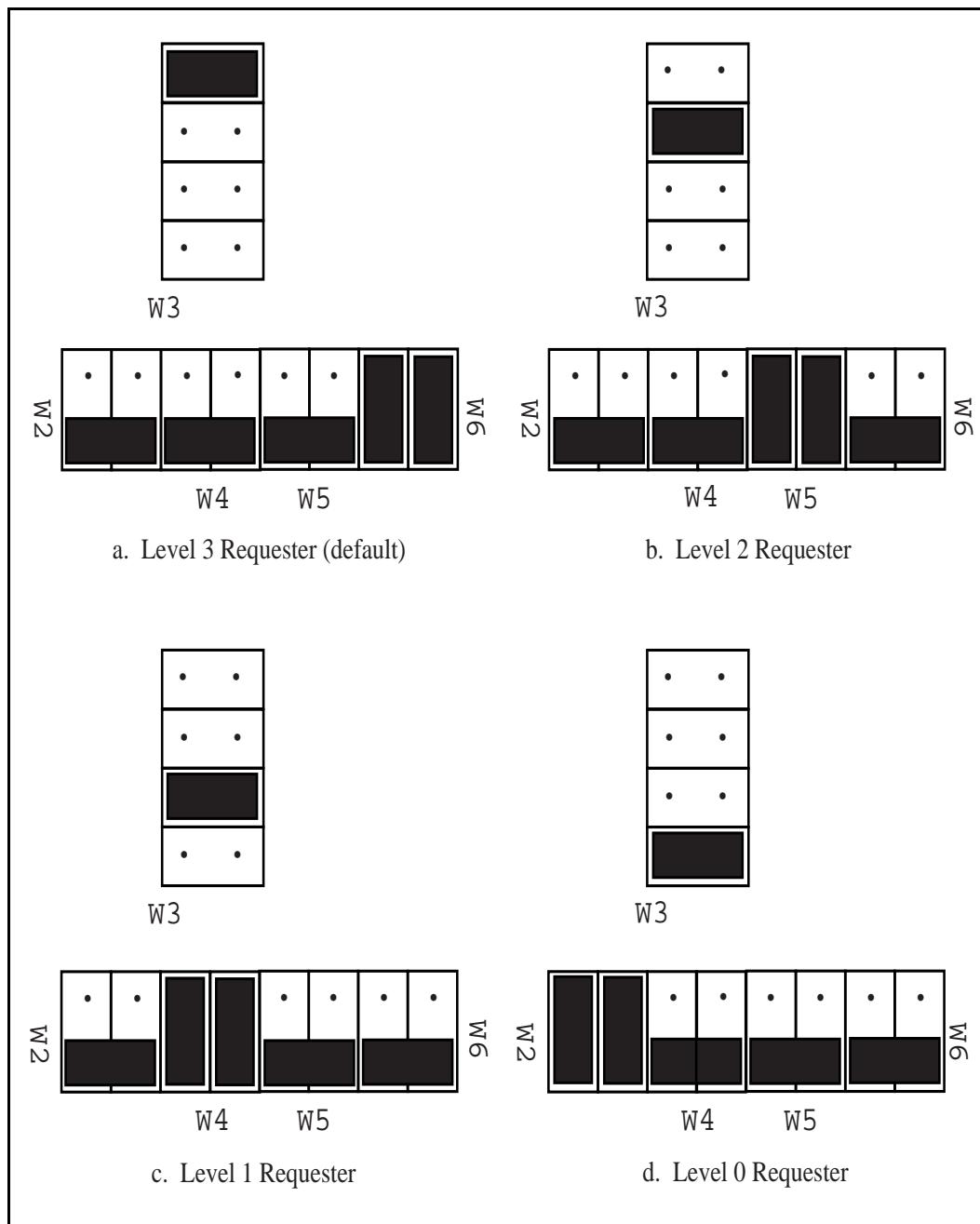
Figure 2-4. VME-MXI VMEbus Requester Jumper Settings

VMEbus Timeout Value

The VME-MXI contains the VMEbus Bus Timeout Unit (BTO) circuitry for the VMEbus. The BTO monitors the current bus cycle and asserts the bus error (BERR*) signal if either data strobe (DS1* or DS0*) remains active for a given amount of time.

The VMEbus system should have one, and only one, device that functions as the BTO Monitor on the VMEbus. The VME-MXI is factory configured to be the VMEbus System Controller (installed in Slot 1) and to provide the VMEbus BTO function. It is recommended that you use the VME-MXI as the System Controller. If, however, you use another device as the VMEbus System Controller, it is recommended that you still retain the VME-MXI as the BTO Monitor, if possible. In this case, disable the BTO function on the System Controller and leave it enabled on the VME-MXI. If it is not possible to disable the BTO function on the System Controller, you can disable the BTO function on the VME-MXI instead. However, be sure that the BTO timeout value on your System Controller is sufficient to allow transfers to complete across the MXIbus. This is especially important for multiple-chassis systems or for systems with more than one VME-MXI.

You can either disable the VMEbus BTO value or set it to 100, 200, or 400 μsec by rearranging the jumper selection at location W8, as shown in Figure 2-5.
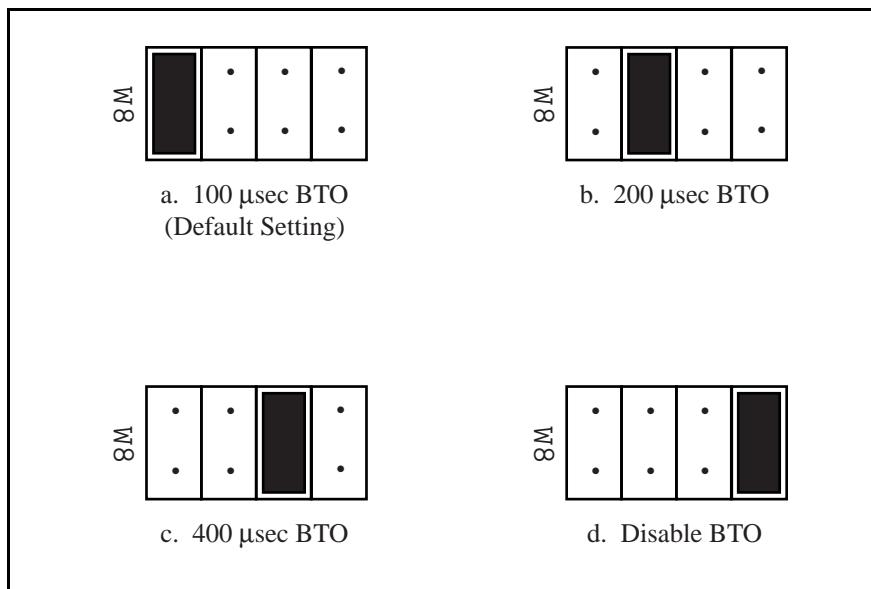
Figure 2-5. VMEbus Timeout Value Selection

Step 4. Install the VME-MXI Hardware

Given below are general installation instructions for the VME-MXI. Consult the user manual or technical reference manual of your VMEbus chassis for specific instructions and warnings.

1. Plug in your VMEbus chassis before installing the VME-MXI. The plug grounds the chassis and protects it from electrical damage while you are installing boards.

   Warning: To protect both yourself and the chassis from electrical hazards, the chassis should remain off until you are finished installing the board.

2. Remove or open any doors or covers blocking access to the chassis slots.

3. Select the slot in the chassis that is correct for your system (see warning below) and insert the VME-MXI in the slot by aligning the top and bottom of the card with the card edge guides inside the chassis. Slowly push the VME-MXI straight into the slot until its plug connectors are resting on the backplane's receptacle connectors. Using slow evenly distributed pressure, press the VME-MXI straight in until it seats in the expansion slot. The front panel of the VME-MXI should be even with the front panel of the chassis.

   Warning: The VME-MXI can be installed into your VMEbus chassis either in Slot 1 when configured as the VMEbus System Controller, or it can be installed in any other slot except Slot 1 when configured as non-System Controller. Installing your VME-MXI into a slot that does not correspond with the jumper settings may result in damage to the VME-MXI, the VMEbus backplane, or both.

4. Tighten the retaining screws on the top and bottom edges of the front panel.

5. Check the installation.

6. Connect the cables as described in the following section before restoring power.

7. Replace or close any doors or covers to the chassis.

Step 5. Connect the MXIbus Cable

There are two basic types of MXIbus cables. MXIbus cables can have either a single connector on each end or a single connector on one cable end and a double connector on the other end. Your VME-MC6000 kit comes standard with a cable with single connectors on each end.

Nonpolarized Cables

The cable with a single connector on each cable end is nonpolarized and can be installed with either end connected to either device. Be sure to tighten the screw locks to ensure proper pin connection.

Polarized Cables

If you are using a MXIbus cable with a single connector on one cable end and a double connector on the other end, it is a polarized cable that must be installed correctly for the system to function properly. Connect the end with the *single* connector to the MC-MXI and the end of the cable with the *double* connector to the VME-MXI. Be sure to tighten the screw locks to ensure proper pin connection.

When the MXIbus cable is properly connected, turn on the VME chassis and the RS/6000 computer. After all devices are powered on and running properly, you may run the the Resource Manager software utility to configure the system.

Chapter 3
NI-VXI Software Installation and Configuration

This chapter lists all the programs and files located on the NI-VXI distribution diskette, contains the instructions for installing the NI-VXI software, and describes how to use the `vxitedit` configuration editor to configure the NI-VXI software for VME.

The NI-VXI software package is a comprehensive set of utility programs and driver software routines that you can use to configure your MXIbus-to-VME link and program your VME system. The NI-VXI software package was actually developed for the VXI-MC6000 kit for VXI systems. Because VXI is a superset of VME, and because the VME-MXI interface board for the VMEbus is compatible with the VXI-MXI interface board for the VXIbus, the NI-VXI software package contains comprehensive tools for VME systems. Some of the utility programs and driver software routines in the NI-VXI software package apply only to VXI systems and are not used in VME systems. Refer to the remainder of this chapter and to the *NI-VXI C Software Reference Manual for VME* to determine which of the NI-VXI software package utility programs and driver functions are used for VME, and which apply only to VXI.

The NI-VXI Software Package

The following files make up the NI-VXI software.

Main Programs and Files

The following are the main programs and files of the NI-VXI software package.

- `vxiinit` is the MXIbus initialization program. This program initializes the MC-MXI board for operation. Run this program after each reboot before using the NI-VXI software.

- `resman` is the National Instruments multiframe Resource Manager program that automatically configures the VME-MXI interfaces installed in your VME chassis. Run this program after `vxiinit`.

- `victext` is the text-based interactive control program that executes functions you enter from the keyboard. It helps you learn the functions, program your VME (and/or VXI) devices, and develop and debug your application program. This program is described in detail in the *NI-VXI Text Utilities Reference Manual*.

- `vxitedit` is the text-based VXI Resource Editor program you use to configure the system. Use the Non-VXI Device Editor in this program to identify details about VME devices installed in your system. You must use this editor to instruct your system about the addresses your VME devices occupy. The Resource Manager can then use this configuration information to automatically *open* the hardware windows so that your RS/6000 can access the VMEbus. This program also displays the system configuration information generated by `resman` after it configures the link to the VMEbus. In VXI systems, you also use this program to edit the manufacturer name and ID numbers, and edit the model names of VXI devices. This program is described in detail in the *NI-VXI Text Utilities Reference Manual*.

- `nivxidd` is the NI-VXI device driver for AIX.

- `cvxi.c` is the NI-VXI C language interface for the NI-VXI device driver.

- `nivxi_add` and `nivxi_delete` are the shell scripts for installing and removing the NI-VXI device driver, respectively.

- `nivxi.stanza` is a file that describes configuration information about the MC-MXI.

- `cfgnivxi` is the configuration file used by the system while installing the driver.

- `readme.doc` contains the latest updates and corrections to the manual when appropriate.

Additional Programs and Files

The `/tbl` directory includes the following table files.

- `mfnameid.tbl` contains the data base of manufacturer names and their ID numbers. This file is primarily used in VXI systems because VXI devices have manufacturer IDs, and is not used in VME systems unless you manually enter information about each VME device in your system through the use of the Non-VXI Device Editor.

- `model.tbl` contains the global data base of the model names, manufacturer names, and the model codes associations. This file is used only in VXI systems and is not used in VME systems unless you manually enter information about each VME device in your system through the use of the Non-VXI Device Editor.

- `device.tbl` contains the global data base of the device names, manufacturer names, model names, and frame and slot associations for the system. This file is used only in VXI systems and is not used in VME systems unless you manually enter information about each VME device in your system through the use of the Non-VXI Device Editor.

- `nonvxi.tbl` contains the global data base for all non-VXI (VME) devices in the system. This file is used for VME systems and describes the address space and interrupt requirements of the VME devices in the system. `resman` uses this file to determine how to open the windows to the VMEbus. Use the Non-VXI Device Editor to configure the address space requirements for your VME devices so that `resman` can configure the hardware windows properly.

- `intcfg.tbl` contains the system interrupt configuration information. In VXI systems, interrupts can be assigned dynamically. In VME systems, however, interrupts are statically configured. This file also contains interrupt mapping information that can be used in a VME system but only in a multimainframe situation.

- `trigcfg.tbl` contains the system trigger configuration editor information. This file is used only in VXI systems and is not used in VME systems.

- `utilbus.tbl` contains the utility bus configuration editor information. This file is used in a VME system but only in a multimainframe situation.

- `vxibus.cfg` contains the bus configuration information.

- `vxila.cfg` contains the logical address configuration information.

- `vxidev.cfg` contains the device configuration information.

The `/include` directory contains include files for the C language interface.

- `nivxi.h` is the main header file containing the C prototypes for the NI-VXI functions.

- `datasize.h` contains data size specifications.

- `busacc.h` contains parameter and return values for the bus access functions.

- `devinfo.h` contains parameter and return values for the device information and system configuration functions.

- `vxiint.h` contains parameter and return values for the interrupt and signal functions.

- `sysint.h` contains parameter and return values for the system interrupt functions.

- `trig.h` contains parameter and return values for the trigger functions. This file is useful in VXI systems but is not applicable for VME systems.

- `ws.h` contains parameter and return values for the Commander and Servant Word Serial functions. This file is useful in VXI systems but is not applicable for VME systems.

The `/hlp` directory contains various help (`.hlp`) files used by `victext` and `vxitedit`.

The `/example` directory contains example programs showing you how to use the NI-VXI software.

Installing the Software

Follow these procedures to install the NI-VXI software.

1. Log on as super-user (`root`).

2. Create a working directory (for example, `/usr/nivxi`) by entering the following command.

    ```
    mkdir /usr/nivxi      <Enter>
    ```

3. Change to this working directory.

    ```
    cd /usr/nivxi         <Enter>
    ```

4. Insert the NI-VXI distribution diskette and copy the files to this directory.

    ```
    tar xvf /dev/fd0      <Enter>
    ```

5. Compile the file `cvxi.c`. You can add `cvxi.o` to an existing library or include `cvxi.o` during the link phase of each compile operation.

6. If you want to run the NI-VXI utilities from any directory other than `/usr/nivxi`, set the environmental variable `NIVXIPATH` to the main directory containing the `/hlp` and `/tbl` subdirectories.

    For example, if your main directory for the NI-VXI files is `/usr/nivxi`, use

    ```
    set  NIVXIPATH=/usr/nivxi
    ```

    The NI-VXI utilities can then read the files from the `/usr/nivxi/tbl` and `/usr/nivxi/hlp` directories.

Loading the NI-VXI AIX Driver

The NI-VXI AIX driver is a loadable driver. You do *not* need to go through the process of linking the driver with the kernel's object files, rebuilding the kernel, and restarting the system as you would have to do with a non-loadable driver. Use the `nivxi_add` script file to load the NI-VXI driver.

Run the install script `nivxi_add` by typing the following command.

```
./nivxi_add x         <Enter>
```

where *x* is the slot number of your MC-MXI board.

If the hardware and the software are installed correctly, the following message appears on the screen.

```
nivxi0 available
```

Unloading the Driver (Optional)

You can also unload the NI-VXI AIX driver from your system. To remove the NI-VXI driver, run the `nivxi_delete` script.

>   `./nivxi_delete nivxi0`   <Enter>

If the driver was removed successfully, a message similar to the following appears on the screen.

>   `nivxi0 deleted`
>
>   `odmdelete: 4 objects deleted`
>   `odmdelete: 1 objects deleted`

Configuring the MC-MXI Hardware

When the driver is installed, the Configuration Manager for AIX for RISC System/6000 automatically configures the hardware by assigning values for the base I/O address, slave memory size and offset, and interrupt and arbitration levels of the interface board according to the `nivxi.stanza` file. If there are any conflicts with the configuration information, the Configuration Manager flags the appropriate error messages. The following section describes the configuration information related to Shared Memory Access in the `nivxi.stanza` file.

Shared Memory Access

It is possible to use RS/6000 memory as a shared memory resource so that the local RS/6000 masters as well as external MXIbus devices such as a VXIbus master in an external mainframe can have direct access to a segment of RS/6000 memory.

If you intend to use a VXIbus shared memory protocol that uses Micro Channel memory, you must select a shared memory base address (Micro Channel address). The shared memory base must be an even multiple of the shared memory size. The range for the shared memory base when the MC-MXI is an A16/A24 device is 0 to 15 megabytes, in increments of 1 megabyte. The range for shared memory base when the MC-MXI is an A16/A32 device is 0 to 3840 megabytes, in increments of 256 megabytes.

Configuring the NI-VXI Software

The NI-VXI software has factory default configurations that match the factory default configurations of the hardware interface. The MC-MXI default configuration is Resource Manager (Logical Address 0), Message-Based device, and MXIbus System Controller. Although the NI-VXI software is designed for VXI systems, you can normally use the default configuration information for your MC-MXI installed in a VME system.

Run the `vxitedit` device configuration utility if you need to make any changes to the default software configuration. If you do not use `vxitedit` to make changes, the default

characteristics of the software will remain in effect. You can also run `vxitedit` if you just want to examine the software configurations.

Note: You must run `vxitedit` and use the Non-VXI Device Editor to configure the VME addresses, interrupt requirements, and other details about the VME devices you want to access from your RS/6000. If you do not use `vxitedit`, the Resource Manager cannot integrate your VME devices into the system.

Run the `vxitedit` program by typing *vxitedit*.

The following sections describe how to modify and/or view the configuration information for the MC-MXI board. To configure VME devices into your VME system, refer to the *VME (Non-VXI) Device Configuration* section later in this chapter.

For further instructions on the use of the `vxitedit` program, refer to the *NI-VXI Text Utilities Reference Manual*.

Configuration Editor

If you want to view the default hardware interface configuration for your hardware link to VME or make changes to it, select the Configuration Editor from the main menu by entering the option number or typing in the editor name. The three configuration options available under the Configuration Editor are Logical Address Configuration, Bus Configuration, and Device Configuration. The Device Configuration option is not used. Do not modify the `vxidev.cfg` file, as it contains some system-dependent timer information used by the `vxiinit` initialization program. For help on individual editors, type *help* while in the corresponding editor.

The following sections describe the various configuration options available under the Logical Address and Bus Configuration Editors.

Logical Address Configuration

Table 3-1 describes the characteristics of the options available under the Logical Address Configuration menu. Normally for VME systems, you do not have to modify the default configuration information.

Table 3-1. Logical Address Configuration Characteristics

Characteristic	Description	Default Value
Logical Address	An 8-bit number that uniquely identifies the MC-MXI. If it is set to 0, it is the Resource Manager.	
	Range = 0 to 254	0
Device Type	Indicates the classification of the MC-MXI.	
	0 = Memory 1 = Extended 2 = Message-Based 3 = Register-Based	2

(continues)

Table 3-1. Logical Address Configuration Characteristics (continued)

Characteristic	Description	Default Value
Address Space	Indicates the addressing mode(s) of the MC-MXI's operational registers. 0 = A16/A24 1 = A16/A32 2 = Reserved 3 = A16 only If the default value is changed, change the Shared Memory Base and Size (the width field) appropriately in the `nivxi.stanza` file.	3
Shared Memory Size	Indicates the size of VME Shared RAM. The MC-MXI local RAM can be slave-accessed in either A24 or A32 space in D08, D16, and D32 data transfer modes. As an A16-only device, this field is not used. When the MC-MXI is an A16/A24 device, the possible values are as follows. 1M, 2M, 4M, 8M When the MC-MXI is an A16/A32 device, the possible values are as follows. 1M, 2M, 4M, 8M, 16M, 32M, 64M, 128M, 256M, 512M, 1024M, 2048M Make sure that the Shared Memory Size is the same as the Shared Memory Size (the width field) in the `nivxi.stanza` file.	 4M 4M
Resource Manager Delay	Time in seconds to wait after power-up or reset before configuring the VME hardware interface and checking for any VXIbus devices residing in the system. Range = 0 to 0x7fffffff sec	 5 sec

(continues)

Table 3-1. Logical Address Configuration Characteristics (continued)

Characteristic	Description	Default Value
Servant Area Size	Servant area size is supplied when the MC-MXI receives the *Read Servant Area* command from the Resource Manager. The Servant area size is an 8-bit value (0 through 255) that indicates the MC-MXI Servant area. The Servant area begins at the logical address following the MC-MXI's logical address, and includes *N* contiguous logical addresses, where *N* is the value of the Servant area size. Not used in VME systems. Range = 0 to 255	0
Protocol Register	Copy of the Protocol register, indicating which protocols the device supports. (Refer to the *VXIbus System Specification*.) Not used in VME systems.	0x0ff0
Response for Servant-Side Word Serial *Read Protocol* query	The information that is supplied when the MC-MXI receives the *Read Protocol* query from the Resource Manager. (Refer to the *VXIbus System Specification*.) Not used in VME systems.	0x8448
Number of Handlers	The number of interrupt handlers that the application requires the MC-MXI to support (minimum of 1). Range = 0 to 7	1
Number of Interrupters	The number of interrupters that the application requires the MC-MXI to support. Range = 0 to 7	0

Bus Configuration

Table 3-2 describes the characteristics of the options available under the Bus Configuration menu.

Table 3-2. Bus Configuration Characteristics

Characteristic	Description	Default Value
MXIbus System Controller	A MXIbus link must have a single device that is responsible for MXIbus interrupt and bus arbitration as well as bus timeouts. This device is referred to as the MXIbus System Controller and is always the first device in the MXIbus daisy-chain. The MC-MXI is always designated as the MXIbus System Controller when used in this kit and should *not* be changed. 0 = Not MXIbus System Controller 1 = MXIbus System Controller	1
MXIbus Fair Request	MXIbus requesters give all MXIbus devices equal opportunity to become bus masters and transfer data, regardless of their position within the MXIbus daisy-chain. MXIbus devices that are not fair requesters have a fixed priority scheme characterized by their relative position within the MXIbus daisy-chain. Devices closest to the MXIbus System Controller have the highest priority. The default configuration programs the MC-MXI to be an unfair requester and should be changed only if additional MXIbus devices are added to the system. Refer to the user manual included with your additional MXIbus devices for more information. 0 = Not a MXIbus Fair Requester 1 = MXIbus Fair Requester	0

VME (Non-VXI) Device Configuration

To add or modify the address space and interrupt requirements for VME devices in your system, select the Non-VXI Device Editor from the `vxitedit` main menu by entering the option number or typing in the editor name. The NI-VXI software needs information about the attributes of the VME devices so that the Resource Manager can integrate them into the system. For help with the Non-VXI Device Editor, type *help*. Refer also to the *NI-VXI Text Utilities Reference Manual* for more information about this editor.

When you use `vxitedit` to configure your non-VXI (VME) devices, be sure to specify a unique pseudo-logical address for each VME device. The pseudo-logical address can be any value between 256 and 511 inclusive. We recommend that you start with 256 for your first device, and then give each subsequent device the next higher number. Assigning pseudo-logical addresses to your VME devices ensures that the Resource Manager program recognizes your VME devices when it configures your interface hardware.

You also need to enter device names, manufacturer names, model names, and other identification information for your VME devices. The `Frame` field should contain the logical address of the VME-MXI in the chassis. The default logical address is 1. Be sure to configure the `InSystem` parameter to instruct the Resource Manager that this particular VME device is indeed installed in your system.

Keep in mind that the Resource Manager will use the configuration information for only those devices that you indicated `InSystem`. When the Resource Manager executes, it displays information regarding the configuration of the interface hardware and your VME devices by referencing their assigned pseudo-logical addresses. After the Resource Manager has executed and configured your interface hardware, your application can retrieve the Resource Manager information about the VME devices in your system that you entered using the Non-VXI Device Editor. To retrieve this information, your application program will use the pseudo-logical address value that you assigned to your VME device(s) as a parameter in the NI-VXI function calls.

You can enter configuration information for each VME device in your system either individually as unique VME devices, or collectively by considering the overall address space requirements of your VME devices as one logical VME device. By using `vxitedit` to configure the address space requirements, the Resource Manager can automatically enable the hardware windows through which your computer can access the corresponding VME addresses where your VME devices reside. You can also run `vxitedit` at any time to examine the software configurations.

Type *list* to examine all the entries. Type *display* for a particular entry to examine all the fields of the particular entry. To edit any existing entry, type *modify* and give the index of the entry and the field to modify. You will be prompted for the new field value. After you modify the entry, `vxitedit` sorts the list in alphabetical order. Make sure to set or clear the `InSystem` field of an entry, depending on whether that device is actually present in your system.

An easy way to create a new entry is to first duplicate a similar entry by typing *duplicate* and giving the index of the existing entry. After you duplicate an entry, modify its

pseudological address and the other address and interrupt requirements. Optionally, you can type *add* and enter all the values for a new entry.

You can delete any entry in the list by typing *delete* and giving the index of the entry. You can save the information you have modified by typing *save*, or type *abort* to ignore the modifications.

As an example, assume you have a VME memory device in your system that requires 64 kilobytes of memory in A24 space. Assign it a pseudo-logical address between 256 and 511. Set its Frame field to the logical address of the VME-MXI in the chassis (the default logical address for the VME-MXI in your kit is 1). Next, assign it a device name, manufacturer name, and model name. Finally, set its A24 memory base and size fields according to its memory requirements, and set the InSystem field to 1.

Run the Resource Manager to automatically configure the system. Verify that the Resource Manager display reflects all the changes you have made to accommodate the VME devices in your system.

Exiting vxitedit and Reinitializing the Hardware

To exit vxitedit, type the number *11* when the main menu is displayed or type *exit*. If you have changed any information, you are prompted to save your changes before exiting the menu. Saving the configuration information updates the configuration files vxila.cfg and vxibus.cfg. The vxidev.cfg file, which contains some system-dependent timer information used by the vxiinit program, should not be modified. Run vxiinit to reinitialize the hardware according to the new settings.

If you changed any of the software configuration settings from the default settings, record the new settings on the *VME-MC6000 Hardware and Software Configuration Form* in Appendix B.

Developing Your Application Program

After you have verified that the driver software has been installed successfully, you can proceed with development of your VXI application software.

Use the VXI Text Interactive Control (victext) program to communicate with VME devices through commands you enter at the keyboard. This feature helps you learn how to communicate with devices, troubleshoot problems, and develop your application.

The victext command set includes the same capability of NI-VXI function calls in addition to auxiliary commands that are unique to victext. You can use this utility to interactively read and write to VME addresses from the keyboard and to display data received from VME address locations on the screen. After each function executes, victext displays the device's response and the status.

The victext utility is designed to help you learn how to use the NI-VXI functions to interact with the VMEbus and program devices. Once you develop a sequence of steps that works successfully for your system, you can easily incorporate the sequence into an application program using the appropriate language and syntax.

The file `cvxi.c` interfaces between your C language application program and the operating system entry points to the VXI driver. The functions available in `cvxi.c` are described in detail in the *NI-VXI C Software Reference Manual for VME*. To use these functions, you must compile `cvxi.c` and include the resulting object file during the link phase of each application compile operation.

The `/example` subdirectory contains various example programs along with a makefile that show how to use various functions in the NI-VXI software and how to develop application programs using these functions.

Refer to the `readme.doc` file for futher development information.

Appendix A
Specifications

This appendix lists various module specifications of the MC-MXI and VME-MXI, such as physical dimensions and power requirements.

MC-MXI

The following pages list the specifications for the MC-MXI module.

Capability Codes

MXIbus

Capability Code	Description
MA32	Master Mode A32, A24 and A16 addressing
MBLT	Master Mode block transfers
SA32	Slave Mode A32, A24, and A16 addressing
SBLT	Slave Mode block transfers
MD32	Master Mode D32, D16, and D08 data sizes
SD32	Slave Mode D32, D16, and D08 data sizes
SC	Optional MXIbus System Controller
FAIR	Can be a fair MXIbus requester
LOCK	Can lock the MXIbus for indivisible transfers
TERM	Can terminate the MXIbus

Micro Channel

Capability Code	Description
MCMD32	Master D32, D16, and D08 data sizes
MCSD32	Slave D32, D16, and D08 data sizes
MCFAIR	Can be a fair Micro Channel requester
AM	Can function as an MC alternate master
LOCK	Can lock the Micro Channel for indivisible transfers
DMA32	Supports D32, D16 or D08 DMA transfers
INT	Can be a Micro Channel Interrupter

Electrical

Source	Typical	Direct Current (max)
+5 VDC	3.0 A	3.5 A

Environmental

Component Temperature   0° to 70° C (32° to 158° F) operating
                               -55° to 150° C (-67° to 302° F) storage

Emissions   FCC Class A

Relative Humidity   0% to 95% noncondensing; operating
                               0% to 100% noncondensing; storage

Safety   Not applicable

Shock and Vibration   Not applicable

Physical

Board Size   Standard full-length Micro Channel board
                               (11.5 in. by 3.475 in.)

Connectors   Single fully implemented MXIbus connector

Slot Requirements   Single 32-bit Micro Channel slot

Reliability

MTBF   Contact Factory

Requirements

Memory space required   64 kilobytes or 32,832 kilobytes

I/O space required   4 kilobytes

Timing

Master Mode

Transfer Type	Transfer Rate
Write	530 nsec
Read	430 nsec
Block Write	290 nsec
Block Read	190 nsec

Slave Mode

Transfer Type	Transfer Rate
Write	340 nsec
Read	440 nsec
Block Write	310 nsec
Block Read	360 nsec

Other

Daisy-Chain Delay   120 nsec
(Passing GIN to GOUT or GOUT generation from System Controller)

VME-MXI

The following pages list the specifications for the VME-MXI module.

Capability Codes

VMEbus

Capability Code	Description
MA32, MA24, MA16	Master Mode A32, A24, and A16 addressing
SA32, SA24, SA16	Slave Mode A32, A24, and A16 addressing
MD32, MD16, MD08(EO)	Master Mode D32, D16, and D08 data sizes
SD32, SD16, SD08(EO)	Slave Mode D32, D16, and D08 data sizes
MBLOCK	Master Mode block transfers
SBLOCK	Slave Mode block transfers
MRMW	Master Mode Read/Modify/Write
SRMW	Slave Mode Read/Modify/Write
PRI	Prioritized arbitration
ROR	Release on Request bus requester
IH	Interrupt Handler
IR	Interrupt Requester
ROAK	Release on Acknowledge interrupter
BTO	Bus Timeout
SC	Optional VMEbus System Controller
IACK	IACK daisy-chain driver

MXIbus

Capability Code	Description
MA32, MA24, MA16	Master Mode A32, A24, and A16 addressing
SA32, SA24, SA16	Slave Mode A32, A24, and A16 addressing
MD32, MD16, MD08(EO)	Master Mode D32, D16, and D08 data sizes
SD32, SD16, SD08(EO)	Slave Mode D32, D16, and D08 data sizes
MBLOCK	Master Mode block transfers
SBLOCK	Slave Mode block transfers
SC	Optional MXIbus System Controller
FAIR	Optional MXIbus fair requester
TERM	Can accept MXIbus termination resistors
IH	Interrupt Handler
IR	Interrupt Requester

Electrical

Source	Typical	Maximum
+5 VDC	5.25 A	7.0 A

Environmental

Component temperature	0° to 70° C operating; -40° to 85° C storage
Relative humidity	10% to 90% noncondensing operating; 0% to 95% noncondensing storage
Emissions	FCC Class A
Safety	Not applicable
Shock and Vibration	Not applicable

Physical

Board size   233.35 mm by 160 mm
                                                              (9.187 in. by 6.299 in.)

Connectors Single fully implemented MXIbus connector
                                                              Single INTX connector (on boards equipped
                                                              with optional INTX daughter card)

Slot Requirements   Single slot

VXI Keying Class   Class 1 TTL

Fully compatible with VMEbus specification

Reliability

MTBF   Contact Factory

Requirements

A16 Space   64 bytes

Timing

<div style="text-align:center">Master Mode</div>

Transfer Type	Transfer Rate
Write	675.5 nsec
Read	625.5 nsec
Block Write	320 nsec
Block Read	270 nsec

<div style="text-align:center">Slave Mode</div>

Transfer Type	Transfer Rate
Write	381 nsec
Read	381 nsec
Block Write	238 nsec
Block Read	238 nsec

Other

Daisy-Chain Delay   120 nsec max
(Passing GIN to GOUT or GOUT generation from System Controller)

Appendix B
Customer Communication

For your convenience, this appendix contains a form to help you gather the information necessary to help us solve technical problems you might have as well as a form you can use to comment on the product documentation. Filling out a copy of the *Technical Support Form* before contacting National Instruments helps us help you better and faster.

Technical Support

Because it helps us solve your problems more easily, we encourage you to fax us a completed copy of the *Technical Support Form* and the configuration forms from the user manuals of the National Instruments hardware or software you are using, even if you wish to contact us later by telephone. You can contact us by fax at any time at the following number:

 (512) 794-5678

You can use the following numbers between the hours of 8:00 a.m. and 5:30 p.m. (central time) to call the National Instruments applications engineering department:

 (512) 794-0100
 (800) 433-3488 (toll-free in the U.S. and Canada)

For best service, please complete the *Technical Support Form* and any applicable configuration forms, record any error messages, and be available at your computer when calling for technical support.

Documentation Comments

You can use the *Documentation Comment Form* to express your opinion of our documents. Please mail or fax it according to the instructions printed at the bottom of the form.

Technical Support Form

Photocopy this form and update it each time you make changes to your software or hardware, and use the completed copy of this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

If you are using any National Instruments hardware or software products related to this problem, include the configuration forms from their user manuals. Use additional pages if necessary.

Name _____

Company _____

Address _____

_____

Fax   (___)_____   Phone   (___)_____

Computer brand _____ Model _____ Processor _____

Operating system _____

Speed _____MHz   RAM _____M   Display adapter _____

Mouse _____yes _____no   Other adapters installed _____

Hard disk capacity _____M   Brand _____

Instruments used _____

National Instruments hardware product model _____   Revision _____

Configuration _____

National Instruments software product _____   Version _____

Configuration _____

The problem is _____

_____

_____

_____

_____

List any error messages _____

_____

_____

_____

_____

The following steps will reproduce the problem _____

_____

_____

_____

VME-MC6000 Hardware and Software Configuration Form

Record the settings and revisions of your hardware and software on the line located to the right of each item. Complete this form each time you revise your software or hardware configuration, and use this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

National Instruments Products

- NI-VXI Software Revision Number   _____
  (Disk Label: *NI-VXI Distribution Diskette for MC-MXI for AIX*)

- Operating System Version   _____

- MC-MXI Logical Address   _____

- MC-MXI Device Type   _____

- MC-MXI Address Space   _____

- MC-MXI Shared Memory Size   _____

- MC-MXI Resource Manager Delay   _____

- MC-MXI Servant Area Size   _____

- MC-MXI Protocol Register   _____

- MC-MXI Response for Servant-Side
  Word Serial *Read Protocol* Query   _____

- Number of MC-MXI Handlers   _____

- Number of MC-MXI Interrupters   _____

- MC-MXI is MXIbus System Controller?   _____

- MXIbus (Fair) Controller?   _____

- MC-MXI Base I/O Address   _____

- MC-MXI Shared Memory MC Base   _____

- MC-MXI Interrupt Levels   _____

- MC-MXI Arbitration Levels   _____

- VME-MXI Installed in Slot 1?   _____

- VME-MXI is VMEbus System Controller?   _____

- VME-MXI Address   _____

- VME-MXI VMEbus Request Level   _____

- VMEbus BTO Timeout Value   _____

- VME-MXI Hardware Revision   _____

- MC-MXI Hardware Revision   _____

Other Products

- Computer Make and Model   _____

- Microprocessor   _____

- Clock Frequency (Bus and Microprocessor) _____

- Type of Video Board Installed   _____

- Programming Language Version   _____

- Other Boards in System   _____

- Base I/O Address of Other Boards   _____

- Arbitration Levels of Other Boards   _____

- Interrupt Level of Other Boards   _____

- VMEbus Chassis Make and Model   _____

- Other VME/VXIbus Devices in System   _____

- Static Logical Addresses of Other Devices   _____

Documentation Comment Form

National Instruments encourages you to comment on the documentation supplied with our products. This information helps us provide quality products to meet your needs.

Title: **Getting Started with Your VME-MC6000 and the NI-VXI™ Software for AIX**

Edition Date: **February 1993**

Part Number: **320398-01**

Please comment on the completeness, clarity, and organization of the manual.

_____

_____

_____

_____

_____

_____

_____

If you find errors in the manual, please record the page numbers and describe the errors.

_____

_____

_____

_____

_____

_____

_____

_____

Thank you for your help.

Name   _____

Title   _____

Company   _____

Address   _____

          _____

Phone   (_____) _____

Mail to: Technical Publications   Fax to: Technical Publications
           National Instruments Corporation   National Instruments Corporation
           6504 Bridge Point Parkway, MS 53-02   MS 53-02
           Austin, TX 78730-5039   (512) 794-5678

Glossary

Prefix	Meaning	Value
n-	nano-	10^{-9}
μ-	micro-	10^{-6}
m-	milli-	10^{-3}
k-	kilo-	10^{3}
M-	mega-	10^{6}
g-	giga-	10^{9}

Symbols

°   degrees

%   percent

±   plus or minus

A

A   amperes

A16 space   VXIbus address space equivalent to the VME 64 kilobyte *short* address space. In VXI, the upper 16 kilobytes of A16 space is allocated for use by VXI devices configuration registers. This 16 kilobyte region is referred to as VXI configuration space.

A24 space   VXIbus address space equivalent to the VME 16 megabyte *standard* address space.

A32 space   VXIbus address space equivalent to the VME 4 gigabyte *extended* address space.

address   Character code that identifies a specific location (or series of locations) in memory.

address modifier   One of six signals in the VMEbus specification used by VMEbus masters to indicate the address space in which a data transfer is to take place.

address space   A set of 2^n memory locations differentiated from other such sets in VXI/VMEbus systems by six addressing lines known as address modifiers. *n* is the number of address lines required to uniquely specify a byte location in a given space. Valid numbers for *n* are 16, 24, and 32.

	In VME/VXI, because there are six address modifiers, there are 64 possible address spaces.
address window	A portion of address space that can be accessed from the application program.
ANSI	American National Standards Institute
arbitration	A process in which a potential bus master gains control over a particular bus.

B

backplane	An assembly, typically a printed circuit board, with 96-pin connectors and signal paths that bus the connector pins. A C-size VXIbus system will have two sets of bused connectors called J1 and J2. A D-size VXIbus system will have three sets of bused connectors called J1, J2, and J3.
block-mode transfer	An uninterrupted transfer of data elements in which the master sources only the first address at the beginning of the cycle. The slave is then responsible for incrementing the address on subsequent transfers so that the next element is transferred to or from the proper storage location. In VME, the data transfer may have no more than 256 elements; MXI does not have this restriction.
BTO unit	Bus Timeout Unit; a functional module that times the duration of each data transfer and terminates the cycle if the duration is excessive. Without the termination capability of this module, a bus master attempt to access a nonexistent slave could result in an indefinitely long wait for a slave response.
bus master	A device that is capable of requesting the Data Transfer Bus (DTB) for the purpose of accessing a slave device.

C

C	Celsius
chassis extender	A device such as the VME-MXI that interfaces a VMEbus chassis to an interconnect bus. It routes bus transactions from the VMEbus to the interconnect bus or vice versa. A chassis extender has a set of registers that defines the routing mechanisms for data transfers, interrupts, and utility bus signals, and has optional VMEbus Slot 1 capability.
CLK10	A 10-MHz, ± 100-ppm, individually buffered (to each module slot), differential ECL system clock that is sourced from Slot 0 of a VXIbus backplane and distributed to Slots 1 through 12 on P2. It is distributed to each slot as a single-source, single-destination signal with a matched delay of under 8 nsec.

CMOS	Complementary Metal Oxide Semiconductor; a process used in making chips.
Commander	A Message-Based device which is also a bus master and can control one or more Servants.
on registers	A set of registers through which the system can identify a module device type, model, manufacturer, address space, and memory requirements. In order to support automatic system and memory configuration, the VXIbus specification requires that all VXIbus devices have a set of such registers.

D

daisy-chain	A method of propagating signals along a bus, in which the devices are prioritized on the basis of their position on the bus.
Data Transfer Bus	DTB; one of four buses on the VMEbus backplane. The DTB is used by a bus master to transfer binary data between itself and a slave device.
DMA	Direct Memory Access; a method by which data is transferred between devices and internal memory without intervention of the central processing unit.
DTB	See *Data Transfer Bus*.
dynamic configuration	A method of automatically assigning logical addresses to VXIbus devices at system startup or other configuration times.
dynamically configured device	A device that has its logical address assigned by the Resource Manager. A VXI device initially responds at Logical Address 255 when its MODID line is asserted. A MXIbus device responds at Logical Address 255 during a priority select cycle. The Resource Manager subsequently assigns it a new logical address, which the device responds to until powered down.

E

ECL	Emitter-Coupled Logic
embedded controller	An intelligent CPU (controller) interface plugged directly into the VXI backplane, giving it direct access to the VXIbus. It must have all of its required VXI interface capabilities built in.
EMI	Electromagnetic Interference
Extended Class device	A class of VXIbus device defined for future expansion of the VXIbus specification. These devices have a subclass register within their configuration space that defines the type of extended device.

external controller | In this configuration, a plug-in interface board in a computer is connected to the VXI mainframe via one or more VXIbus extended controllers. The computer then exerts overall control over VXIbus system operations.

F

F | Fahrenheit

fair requester | A MXIbus master that will not arbitrate for the MXIbus after releasing it until it detects the bus request signal inactive. This ensures that all requesting devices will be granted use of the bus.

FCC | Federal Communications Commission

G

GIN | Daisy-chain Grant In signal

GOUT | Daisy-chain Grant Out signal

H

hex | Hexadecimal; the numbering system with base 16, using the digits 0 to 9 and letters A to F.

Hz | hertz; cycles per second.

I

IC | Integrated Circuit

IEEE | Institute of Electrical and Electronics Engineers

IEEE 1014 | The VMEbus specification

in. | inches

I/O | input/output; the techniques, media, and devices used to achieve communication between machines and users.

interrupt | A means for a device to request service from another device.

interrupt handler | A VMEbus functional module that detects interrupt requests generated by Interrupters and responds to those requests by requesting status and identify information.

interrupt level | The relative priority at which a device can interrupt.

INTX | Interrupt Timing and Extension

K

K	1,024; kilobytes of memory

L

LED	Light Emitting Diode
logical address	An 8-bit number that uniquely identifies each VXIbus device in a system. It defines the A16 register address of a device, and indicates Commander and Servant relationships.

M

M	1,048,576; megabytes of memory
m	meters
master	A functional part of a MXI/VME/VXIbus device that initiates data transfers on the backplane. A transfer can be either a read or a write.
master-mode operation	A device is in master mode if it is performing a bus cycle which it initiated.
Message-Based device	An intelligent device that implements the defined VXIbus registers and communication protocols. These devices are able to use Word Serial Protocol to communicate with one another through communication registers.
MODID	A set of signal lines on the VXI backplane that VXI systems use to identify which modules are located in which slots in the mainframe.
MTBF	Mean Time Between Failure
MXIbus	Multisystem eXtension Interface Bus; a high-performance communication link that interconnects devices using round, flexible cables.
MXIbus System Controller	A functional module that has arbiter, daisy-chain driver, and MXIbus cycle timeout responsibility. Always the first device in the MXIbus daisy-chain.

N

NI-VXI	The National Instruments bus interface software for VME/VXIbus systems.

P

propagation	The transmission of signal through a computer system.

R

Register-Based device	A Servant-only device that supports VXIbus configuration registers. Register-Based devices are typically controlled by Message-Based devices via device-dependent register reads and writes.
resman	The name of the National Instruments Resource Manager in NI-VXI bus interface software. See *Resource Manager*.
Resource Manager	A Message-Based Commander located at Logical Address 0, which provides configuration management services such as address map configuration, Commander and Servant mappings, and self-test and diagnostic management.

S

sec	seconds
Servant	A device controlled by a Commander; there are Message-Based and Register-Based Servants.
Shared Memory Protocol	A communication protocol that uses a block of memory that is accessible to both a client and a server. The memory block operates as a message buffer for communications.
slave	A functional part of a MXI/VME/VXIbus device that detects data transfer cycles initiated by a VMEbus master and responds to the transfers when the address specifies one of the device's registers.
slave-mode operation	A device is in slave mode it if is responding to a bus cycle.
statically configured device	A device whose logical address cannot be set through software; that is, it is not dynamically configurable.
SYSFAIL	A VMEbus signal that is used by a device to indicate an internal failure. A failed device asserts this line.

T

trigger	Either TTL or ECL lines used for intermodule communication.
TTL	Transistor-Transistor Logic

V

V	volts
VDC	volts direct current
victext	Text-based VXI Interactive Control Program, a part of the NI-VXI bus interface software package. Used to program VXI devices, and develop and debug VXI application programs.
VME	Versa Module Eurocard or IEEE 1014
VMEbus System Controller	A device configured for installation in Slot 1 of a VMEbus chassis. This device is unique in the VMEbus system in that it performs the VMEbus System Controller functions, including clock sourcing and arbitration for data transfers across the backplane. Installing such a device into any other slot can damage the device, the VMEbus backplane, or both.
VXIbus	VMEbus Extensions for Instrumentation
vxiinit	A program in the NI-VXI bus interface software package that initializes the board interrupts, shared RAM, VXI register configurations, and bus configurations. This program must be run at least once after rebooting your machine to initialize the system.
vxitedit	Text-based VXI Resource Editor program, a part of the NI-VXI bus interface software package. Used to configure the system, edit the manufacturer name and ID numbers, edit the model names of VXI and non-VXI devices in the system, as well as the system interrupt configuration information, and display the system configuration information generated by the Resource Manager.