VME-MXI

Getting Started with Your VME-AT2000 and the NI-VXI™ Software for MS-DOS

August 1994 Edition

Part Number 320332B-01

National Instruments Corporate Headquarters
6504 Bridge Point Parkway
Austin, TX 78730-5039
(512) 794-0100
Technical support fax: (800) 328-2203
(512) 794-5678

Branch Offices:
Australia (03) 879 9422, Austria (0662) 435986, Belgium 02/757.00.20, Canada (Ontario) (519) 622-9310, Canada (Québec) (514) 694-8521, Denmark 45 76 26 00, Finland (90) 527 2321, France (1) 48 14 24 24, Germany 089/741 31 30, Italy 02/48301892, Japan (03) 3788-1921, Netherlands 03480-33466, Norway 32-848400, Spain (91) 640 0085, Sweden 08-730 49 70, Switzerland 056/20 51 51, U.K. 0635 523545

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

Trademarks

Warning Regarding Medical and Clinical Use of National Instruments Products

National Instruments products are not designed with components and testing intended to ensure a level of reliability suitable for use in treatment and diagnosis of humans. Applications of National Instruments products involving medical or clinical treatment can create a potential for accidental injury caused by product failure, or by errors on the part of the user or application designer. Any use or application of National Instruments products for or involving medical or clinical treatment must be performed by properly trained and qualified medical personnel, and all traditional medical safeguards, equipment, and procedures that are appropriate in the particular situation to prevent serious injury or death should always continue to be used when National Instruments products are being used. National Instruments products are NOT intended to be a substitute for any form of established process, procedure, or equipment used to monitor or safeguard human health and safety in medical or clinical treatment.

Contents

Figures

Tables

About This Manual

This manual contains instructions for installing and configuring the National Instruments VME-AT2000 interface kit. This manual is meant to be used with the *NI-VXI C Software Reference Manual for VME*, the *NI-VXI Software Reference Manual for BASIC*, and the *NI-VXI DOS Utilities Reference Manual.*

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. The *NI-VXI Software Reference Manual for BASIC* covers the same material but with BASIC syntax. Refer to the *NI-VXI DOS Utilities Reference Manual* to learn more about the `VIC` and `VXIEDIT` programs. The *LabWindows VXI Library Reference Manual*, part number 320318-01, explains how to install and use the optional LabWindows VXI Library for the AT-MXI.

Organization of This Manual

Getting Started with Your VME-AT2000 and the NI-VXI Software for MS-DOS is organized as follows:

- Chapter 1, *Introduction*, describes the VME-AT2000 interface kit, lists what you need to get started, and lists optional equipment and software.

- Chapter 2, *Hardware Configuration and Installation*, contains the instructions to configure and install the VME-AT2000 interface kit.

- Chapter 3, *NI-VXI Software Installation and Configuration*, lists all the programs and files located on the NI-VXI distribution diskette(s), contains step-by-step instructions for installing and configuring the NI-VXI software for MS-DOS for VME, and contains a description of the `VXIEDIT` configuration editor.
- Appendix A, *Specifications*, lists various module specifications of the AT-MXI and VME-MXI, such as physical dimensions and power requirements.
- Appendix B, *Troubleshooting*, addresses certain problems you may encounter when using the NI-VXI bus interface software on the AT-MXI platform.
- Appendix C, *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, mnemonics, and symbols.

Conventions Used in This Manual

The following conventions are used to distinguish elements of text throughout this manual:

bold Bold text denotes menus or menu items.

italic Italic text denotes emphasis, a cross reference, 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 disk drives, paths, directories, programs, subprograms, device names, filenames and extensions, and for statements and comments taken from program code.

`bold monospace` Bold text in this font denotes the messages and responses that the computer automatically prints to the screen.

bold italic Bold italic text denotes a note, caution, or warning.

<> Angle brackets enclose the name of a key on the keyboard–for example, <Esc>.

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

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
- *Multisystem Extension Interface Bus Specification*, Version 1.2
- *NI-VXI Software Reference Manual for C* (for VXI systems), National Instruments Corporation
- *VME-MXI User Manual*, Instruments Corporation
- VXI-1, *VXIbus System Specification*, Rev. 1.4, VXIbus Consortium
- VXI-6, *VXIbus Mainframe Extender Specification*, Rev. 1.0, VXIbus Consortium

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 C, *Customer Communication*, at the end of this manual.

Chapter 1
Introduction

This chapter describes the VME-AT2000 interface kit, lists what you need to get started, and lists optional equipment and software.

The VME-AT2000 is an interface kit that links any IBM Personal Computer AT or compatible computer (hereafter referred to as the PC AT) directly to the VMEbus. A PC AT equipped with a VME-AT2000 can function as a VME master and/or slave device. The VME-AT2000 makes the PC AT appear as though it were an embedded CPU plugged directly into the VME backplane, or as though it has internal VME slots for plug-in boards. The driver software for the VME boards must, of course, be ported to run on the PC AT with the VME-AT2000 kit.

What You Need to Get Started

❑ MS-DOS version 5.0 (or equivalent) or higher installed on your computer

❑ AT-MXI interface board

❑ Standard VME-MXI interface module

or

Enhanced VME-MXI interface module with INTX option

❑ NI-VXI distribution disks for the AT-MXI and MS-DOS

You received both 3.5 in. disks and 5.25 in. disks in your kit.

❑ 2 m Type M1 MXIbus cable

Note: ***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.***

Optional Equipment

Contact National Instruments for ordering information for any of the following optional equipment.

- VXI-MXI standard VXIbus mainframe extender kit
- VXI-MXI enhanced VXIbus mainframe extender kit
- VME-MXI standard VXIbus chassis extender kit
- VME-MXI enhanced VXIbus chassis extender kit
- MXIbus terminating pac (External)
- Type M1 MXIbus cables (straight point-to-point connectors)
- Type M2 MXIbus cables (straight point-to-right angle daisy-chain connectors)
- Type M3 MXIbus cables (right angle point-to-right angle daisy-chain connectors)

The Type M1, M2, and M3 MXIbus cables are available in the following lengths:

- 1 m
- 2 m
- 4 m
- 8 m
- 20 m

Optional Software

If you want to use LabWindows with your VME-AT2000 system, you can order the LabWindows for DOS, VXI Development System. It contains the following components:

- LabWindows for DOS Full Develoment System
- LabWindows for DOS VXI Libraries
- LabWindows for DOS VXI Instrument Library

If you want to use LabVIEW with your VME-AT2000 system, you must order NI-VXI Windows Software for the AT-MXI and the LabVIEW for Windows VXI Development System. It contains the following components:

- LabVIEW for Windows Full Development System
- LabVIEW for Windows VXI Library
- LabVIEW for Windows/Sun VXI Instrument Library

Chapter 2
Hardware Configuration and Installation

This chapter contains the instructions to configure and install the VME-AT2000 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 AT-MXI and VME-MXI hardware.
2. Configure the AT-MXI hardware.
3. Install the AT-MXI hardware.
4. Configure the VME-MXI hardware.
5. Install the VME-MXI hardware.
6. Connect the MXIbus cable.

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

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

1. Before attempting to configure or install the AT-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 AT-MXI board and VME-MXI module are shipped packaged in antistatic plastic bags to prevent electrostatic damage. Some of the circuitry on the AT-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 AT-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. Configure the AT-MXI Hardware

Figure 2-1 shows the location of the AT-MXI configuration jumpers and switches.

Art not available in PDF version of document.

Figure 2-1. AT-MXI Parts Locator Diagram

Switch and Jumper Settings

You can configure three options on the AT-MXI board:

- Base I/O address
- Interrupt levels
- DMA channels

Table 2-1 shows the factory settings and optional configurations for the switches and jumpers on the AT-MXI.

Table 2-1. AT-MXI Factory Default Settings and Optional Configurations

AT-MXI	Default	Optional
Base I/O Address (hex)	340	100 to 3E0, increments of 20 hex
Master DMA Channel	6	0, 1, 2, 3, 5, 6, 7, and Not Used
Slave DMA Channel	3	0, 1, 2, 3, 5, 6, 7, and Not Used
Board Interrupt Level	12	3, 4, 5, 6, 7, 9, 10, 11, 12, 14, and 15
MXIbus Interrupt Level	10	3, 4, 5, 6, 7, 9, 10, 11, 12, 14, 15, and Not Used

The factory-configured settings of the base I/O address, the interrupt levels, and the Direct Memory Access (DMA) channels are suitable for most computer systems. The following sections describe under what conditions you would be required to change the configuration switches and/or jumpers on the AT-MXI and how to make these changes.

Base I/O Address Selection

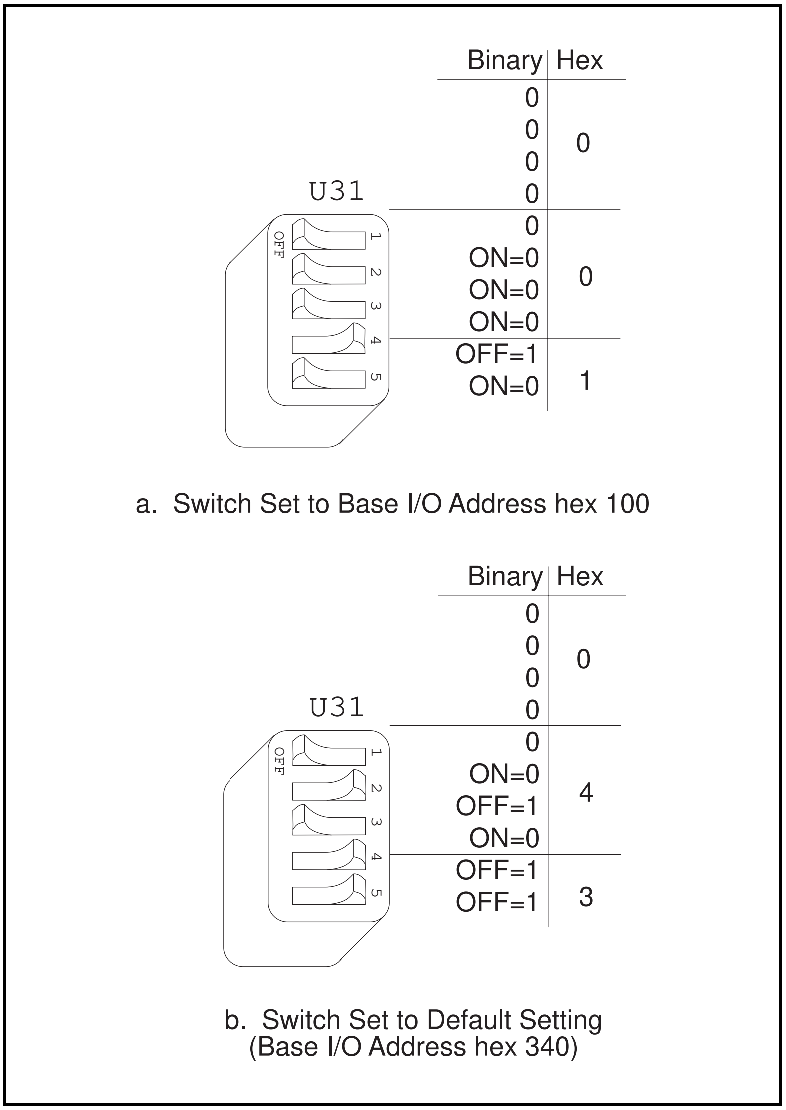
The base I/O address of the AT-MXI is the starting address of the AT-MXI configuration registers in PC AT I/O space. The base I/O address is determined by the position of the five switches at location U31, as shown in Figure 2-1. The switches are set at the factory for a base I/O address of 340 hex. Because the AT-MXI requires 32 bytes of consecutive I/O space for its internal registers, the factory configuration uses the I/O address space in the range of 340 to 35F hex.

Note: ***Check to determine that this space is not already used by other I/O interfaces installed in your PC AT computer. If any equipment in your computer uses this I/O address space, you must change either the base I/O address of the AT-MXI or the I/O address space requirements of the other device. All PC AT devices must have a unique partition of the system's I/O address space. If you change the base I/O address of the AT-MXI, you must make a corresponding change to the AT-MXI configuration software as described in the*** *Configuring the NI-VXI Software* ***section of Chapter 3 of this manual.***

Each switch in U31 (1 through 5) corresponds to one of the PC AT address lines (A5 through A9). The first switch (1) corresponds to address line A5, the next switch (2) corresponds to address line A6, and so on. The five least significant bits of the address (A4 through A0) are used by the AT-MXI to select the appropriate AT-MXI register and cannot be changed; therefore, bits A4 through A0 are always zeros when determining the base I/O address.

To change the base I/O address of the AT-MXI, press the side marked *OFF* to select a binary value of 1 for the corresponding address bit. Press the *ON* side of the switch to select a binary value of 0 for the corresponding address bit. Refer to Table 2-2.

Figure 2-2 shows two possible switch settings.



a. Switch Set to Base I/O Address hex 100

b. Switch Set to Default Setting
(Base I/O Address hex 340)

Figure 2-2. Base I/O Address Switch Settings

Table 2-2 lists the 24 possible switch settings, the corresponding base I/O address, and the I/O address space used for that setting. Notice that the base address settings that correspond to an I/O address in the range from 0 to FF hex are not listed. These addresses are used by logic on the PC AT motherboard and cannot be used by I/O adapter modules.

Table 2-2. Possible Base I/O Address Settings for the AT-MXI

Switch Setting					**Base I/O Address**	**I/O Ports Used**
A9	**A8**	**A7**	**A6**	**A5**	**(hex)**	**(hex)**
0	1	0	0	0	100	100 - 11F
0	1	0	0	1	120	120 - 13F
0	1	0	1	0	140	140 - 15F
0	1	0	1	1	160	160 - 17F
0	1	1	0	0	180	180 - 19F
0	1	1	0	1	1A0	1A0 - 1BF
0	1	1	1	0	1C0	1C0 - 1DF
0	1	1	1	1	1E0	1E0 - 1FF
1	0	0	0	0	200	200 - 21F
1	0	0	0	1	220	220 - 23F
1	0	0	1	0	240	240 - 25F
1	0	0	1	1	260	260 - 27F
1	0	1	0	0	280	280 - 29F
1	0	1	0	1	2A0	2A0 - 2BF
1	0	1	1	0	2C0	2C0 - 2DF
1	0	1	1	1	2E0	2E0 - 2FF
1	1	0	0	0	300	300 - 31F
1	1	0	0	1	320	320 - 33F
1	1	0	1	0	340 (default)	340 - 35F
1	1	0	1	1	360	360 - 37F
1	1	1	0	0	380	380 - 39F
1	1	1	0	1	3A0	3A0 - 3BF
1	1	1	1	0	3C0	3C0 - 3DF
1	1	1	1	1	3E0	3E0 - 3FF

Interrupt Level Selection

The AT-MXI interface board can use either one or two of the eleven interrupt levels of the PC AT I/O channel. Setting up an interrupt level for operation involves two steps. First you select the interrupt level by arranging the jumpers on an array of pins. Next you enable the interrupt level in the system software.

Note: ***You must enable the interrupt levels in the system software before they can function. If you do not enable the interrupt level, that level will not be driven and can be used by other devices, regardless of the positions of the jumpers.***

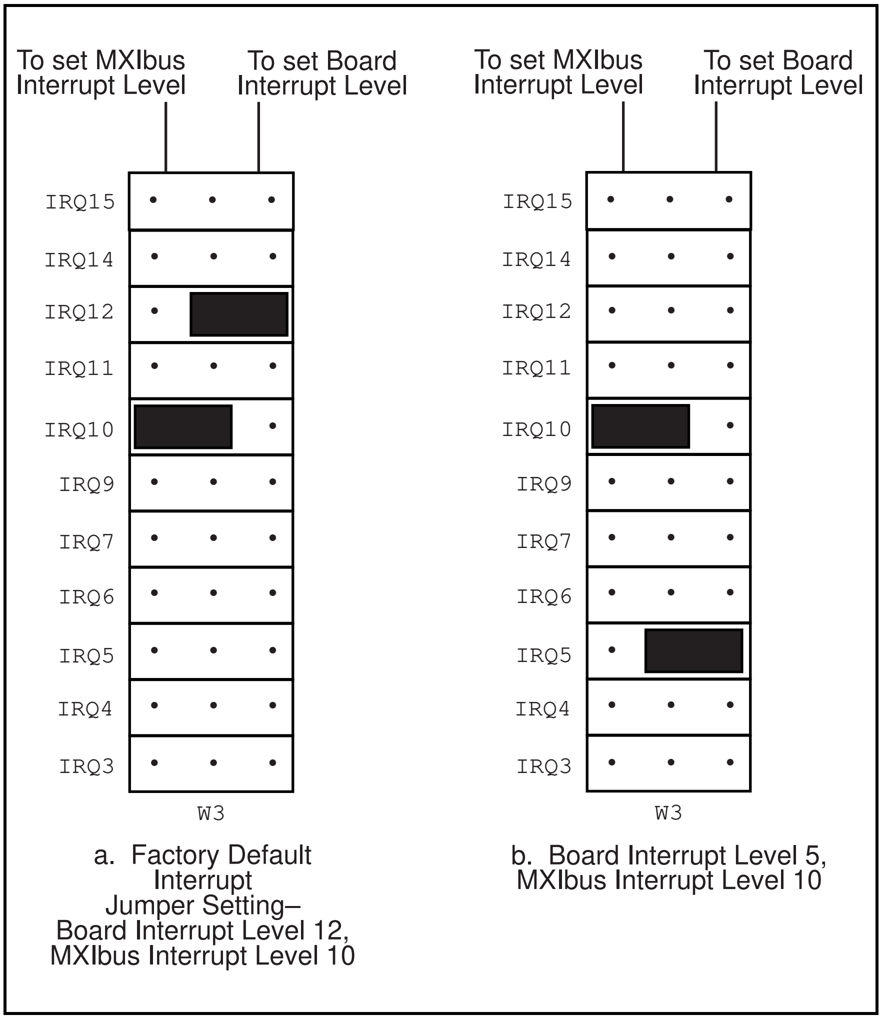
Interrupt levels are selected by the position of two jumpers on the 3 by 11 array of pins labeled *W3*, located above the I/O card-edge connector on the AT-MXI (refer to Figure 2-1). Use the jumper *farther* from the card-edge connector to select the PC AT interrupt level that will convey board status and error information. Remember to enable this level, or board interrupt, in the system software for the AT-MXI to function properly. The factory default setting is level 12 and is enabled on this level by the default configuration software.

Use the jumper on the W3 pin array *closer* to the I/O card-edge connector to select which PC AT interrupt level corresponds to the MXIbus interrupt signal *IRQ**. Because the MXIbus interrupt is also one of the conditions covered by the other jumper, you normally do not need to use a separate interrupt level for the MXIbus *IRQ** signal; it is useful only if you want a different interrupt vector or priority for MXIbus interrupts. This jumper is set at the factory to a default level of 10.

Note: ***The AT-MXI does not have the ability to share interrupt levels with other devices. If you select an interrupt level by placing a jumper on a particular level and enable that level in software, no other device in the system can use that level. Make sure that no other devices in your system use the interrupt level(s) selected and enabled for use by the AT-MXI. If they do, change the interrupt level(s) of either the AT-MXI or the other devices. If you change an AT-MXI interrupt level, make a corresponding change to the AT-MXI configuration software as described in the*** *Configuring the NI-VXI Software* ***section of Chapter 3.***

The AT-MXI can use interrupt levels IRQ3, 4, 5, 6, 7, 9, 10, 11, 12, 14, and 15. Be careful when re-assigning interrupt levels on the AT-MXI. Notice that most PC ATs use interrupt level 6 for the diskette drive controller and interrupt level 14 for the hard disk drive controller. Other interrupt levels might be used by standard logic devices on the motherboard so check your computer documentation before changing interrupt levels.

Once you have chosen an interrupt level, place the jumper on the appropriate pins to select that interrupt level. Use the two rows of pins farther from the card-edge connector to select the board interrupt level, and the two rows of pins closer to the card-edge connector to select the MXIbus interrupt level. Figure 2-3a shows the factory default interrupt jumper setting of the AT-MXI, with board interrupt level 12 and MXIbus interrupt level 10.


To set MXIbus Interrupt Level
To set Board Interrupt Level
IRQ15
IRQ14
IRQ12
IRQ11
IRQ10
IRQ9
IRQ7
IRQ6
IRQ5
IRQ4
IRQ3
W3
To set MXIbus Interrupt Level
To set Board Interrupt Level
IRQ15
IRQ14
IRQ12
IRQ11
IRQ10
IRQ9
IRQ7
IRQ6
IRQ5
IRQ4
IRQ3
W3

a. Factory Default Interrupt Jumper Setting– Board Interrupt Level 12, MXIbus Interrupt Level 10

b. Board Interrupt Level 5, MXIbus Interrupt Level 10

Figure 2-3. Board and MXIbus Interrupt Jumper Settings

To change to another interrupt level, remove the appropriate jumper from its current position and place it on the new posts. Figure 2-3b shows the board interrupt level changed to IRQ5.

DMA Channel Selection

The AT-MXI interface board can use one, two, or none of the seven DMA channels of the PC AT I/O channel. Setting up a DMA channel for operation involves two steps. First you select the DMA channel by arranging the jumpers on an array of pins. Next you enable the DMA channel in the system software.

Note: ***You must enable the DMA channels in the system software before they can function. If you do not enable the DMA channel, that channel will not be driven and can be used by other devices, regardless of the positions of the jumpers.***

Select the DMA channels on the two 3 by 7 arrays of pins labeled *W1* and *W2*, located above the I/O card-edge connector on the AT-MXI (refer to Figure 2-1). Use the W1 array to select the DMA request line(s), and use the W2 array to select the DMA acknowledge line(s). You must position two jumpers to select a single DMA channel. The DMA ACKnowledge (DACK*n*) and DMA ReQuest (DRQ*n*) lines selected must have the same numeric suffix for proper operation. Therefore, make sure that the jumper positions on the W1 array are identical to the jumper positions on the W2 array.

Master Mode Versus Slave Mode

The AT-MXI can function as both a MXIbus master and a MXIbus slave. As a MXIbus master, the AT-MXI circuitry determines whether a PC AT cycle is to be mapped into a MXIbus cycle intended for some external MXIbus device such as a VMEbus backplane. As a MXIbus slave, the AT-MXI circuitry determines whether an external device is attempting to access PC AT memory or I/O resources.

Note: ***When allocating DMA channels for use by the AT-MXI, keep in mind that master-mode and slave-mode operation are two distinct asynchronous functions and require different DMA channels.***

Although the master-mode DMA jumpers are set at the factory to use DMA channel 6, the software statically disables the master-mode DMA channel with a setting of `NONE` and cannot be changed. While the AT-MXI itself is capable of using DMA to perform high-speed block-mode transfers to or from external MXIbus devices, the processor is able to transfer data at significantly faster rates by using the `movs` (move string) instruction. DMA transfers offer superior performance only in asynchronous function calls, which begin an operation and return immediately, freeing the processor to do other useful work. However, because DMA transfers would be applicable only for the `VXImove` function, which is *not* asynchronous, the software is configured to ignore this option, making the hardware setting irrelevant. The field remains in the `VXIEDIT` software configuration program for backward compatibility only.

The slave-mode DMA jumpers are set at the factory to use DMA channel 3. Although the default software configuration enables this setting, several factors effectively disable the slave-mode DMA channel. As a result, DMA channel 3 can be used by other devices if you do not need to use slave-mode DMA with your AT-MXI.

The slave-mode DMA channel is necessary only for setting up shared access to PC AT resource from external MXIbus devices. If you intend to communicate with VME devices using a shared memory protocol that uses the PC AT memory, you must select and enable a slave-mode DMA channel. However, because the AT-MXI does not share resources in its default software configuration, you need to change some software settings before you can use this feature. Refer to the *Logical Address Configuration* section of Chapter 3, *NI-VXI Software Installation and Configuration*, to see the default settings of the fields that affect the slave-mode DMA channel. The default settings are as follows:

- The **Address Space** field is set to A16 only. To share RAM you need to change this field to A16/A24.
- The **VXI Shared RAM Window Size** field is disabled. To share RAM you need to assign a value in the allowable range.
- The **Slave I/O Window Size** field is disabled. To share I/O space you need to assign a value in the allowable range.
- The **Slave DMA Channel** field is set to use channel 3, which is acceptable. If you modified this field to the NONE setting, slave accesses to shared RAM in VME space and PC I/O space would not succeed. You could assign a NONE setting to this field if you intend to keep the three settings described above in their default settings. This change would free up the DMA channel for use by other devices in the system. On the other hand, if you ever need to switch back and forth between A16 and A24 space, it would be easier to have a slave DMA channel already reserved for the AT-MXI, and just change the appropriate settings in `VXIEDIT`. Remember that the hardware and software settings must match if you intend to use a slave DMA channel.

Note: ***Seldom, if ever, can the AT-MXI share DMA channels with other devices. If you have selected a DMA channel by placing jumpers on that channel's request and acknowledge lines and enabled the channel in software, no other devices in your system should use that channel. If DMA channels conflict, change the DMA channel(s) used by either the AT-MXI or the other device(s). If you change an AT-MXI DMA channel, make a corresponding change to the AT-MXI configuration software as described in the*** *Configuring the NI-VXI Software* ***section of Chapter 3.***

The AT-MXI can use DMA Channels 0, 1, 2, 3, 5, 6, and 7. Be careful when reassigning DMA channels on the AT-MXI. Notice that most PC ATs use DMA Channel 2 for the disk controller interface. It is possible that various standard logic devices on the motherboard may use other DMA channels, so check your computer documentation before changing DMA channels.

Notice that the PC AT makes a distinction between 8-bit and 16-bit DMA channels. The 8-bit channels are 0, 1, 2, and 3. The 16-bit channels are 5, 6, and 7. The master-mode DMA channel setting must match the data width of the intended block transfers. It is preferable to use one of the 16-bit channels for the master-mode interface because a 16-bit DMA channel can transfer twice the amount of data in the same number of cycles. The slave-mode DMA channel is used only to request the PC AT bus for an alternate PC AT bus master cycle. It can use any available 8-bit or 16-bit channel regardless of the intended data width of the transfers.

Use the two rows of pins farther from the card-edge connector to select the master-mode DMA channel and the two rows closer to the card-edge connector to select the slave-mode DMA channel. Remember that the jumper positions should be identical on both the W1 and W2 arrays. Figure 2-4a shows the factory default DMA channel setting of the AT-MXI, with master-mode DMA Channel 6 and slave-mode DMA Channel 3.

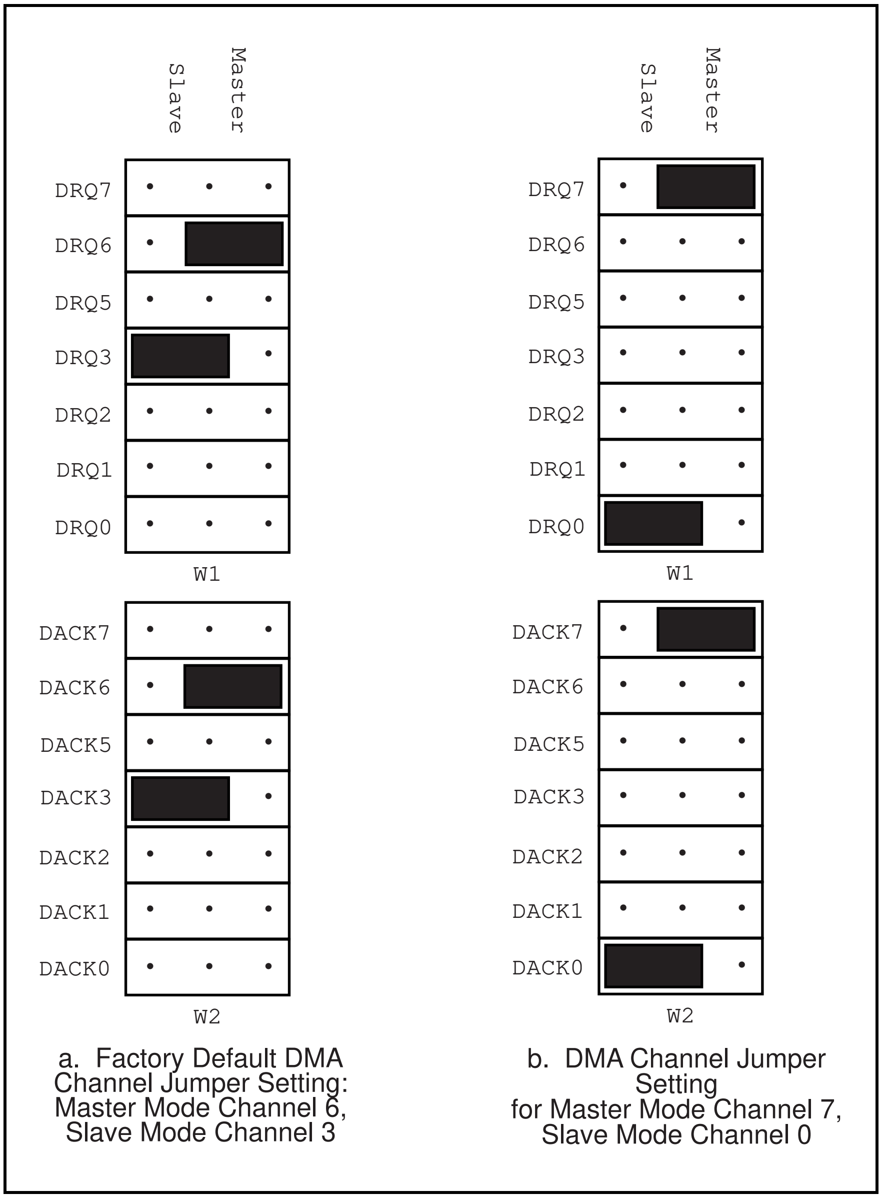

Slave
Master
DRQ7
DRQ6
DRQ5
DRQ3
DRQ2
DRQ1
DRQ0
W1
DACK7
DACK6
DACK5
DACK3
DACK2
DACK1
DACK0
W2
a. Factory Default DMA Channel Jumper Setting: Master Mode Channel 6, Slave Mode Channel 3
b. DMA Channel Jumper Setting for Master Mode Channel 7, Slave Mode Channel 0

Figure 2-4. DMA Channel Settings

To select a new DMA channel, move both the DRQ and DACK jumpers onto the appropriate pins. Remember to make a corresponding change in the system software to enable the new DMA channels. Figure 2-4b displays the jumper position for selecting master-mode DMA Channel 7 and slave-mode DMA Channel 0.

Step 3. Install the AT-MXI Hardware

In the space provided here, record the settings of the base I/O address, the DMA channel(s), and the interrupt level(s) for future reference. You will need this information when you install the software.

AT-MXI	Setting
Base I/O Address	
Board Interrupt Level	
MXIbus Interrupt Level	
Master DMA Channel	
Slave DMA Channel	

Before attempting to install the AT-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 PC AT slots. Normally, this will be a problem only if the cable connector hoods for the adjacent slots are also oversized. When choosing a PC AT slot in which to install the AT-MXI, verify that the MXIbus cable connector will not interfere with cables and connectors in other PC AT slots. If necessary, reposition the boards in the system to prevent cabling conflicts. It may also help to install the AT-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 AT-MXI to co-exist in an existing PC AT system by repositioning the boards, you can use one of the MXIbus cable options with a standard connector hood on the cable end that attaches to the AT-MXI. The standard 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 AT-MXI be the first device in the MXIbus daisy-chain because a cable with a single 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 PC AT computer before installing the AT-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 PC AT I/O bus.

3. Select any available 16-bit full-length PC AT expansion slot. The 16-bit expansion slots have two card-edge receptacle connectors.

4. Locate the metal bracket that covers the cut-out in the back panel of the PC AT chassis for the slot you have selected. Remove and save the bracket-retaining screw and the bracket cover.

5. 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 AT-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 front of the AT-MXI until its card edge connector is resting on the expansion slot receptacle. Using slow, evenly distributed pressure, press the AT-MXI straight down until it seats in the expansion slot.

7. Reinstall the bracket retaining screw to secure the AT-MXI to the back panel rail.

8. Check the installation.

9. Replace the computer cover.

Figure 2-5 shows an AT-MXI installed in a PC AT with a MXIbus cable attached.

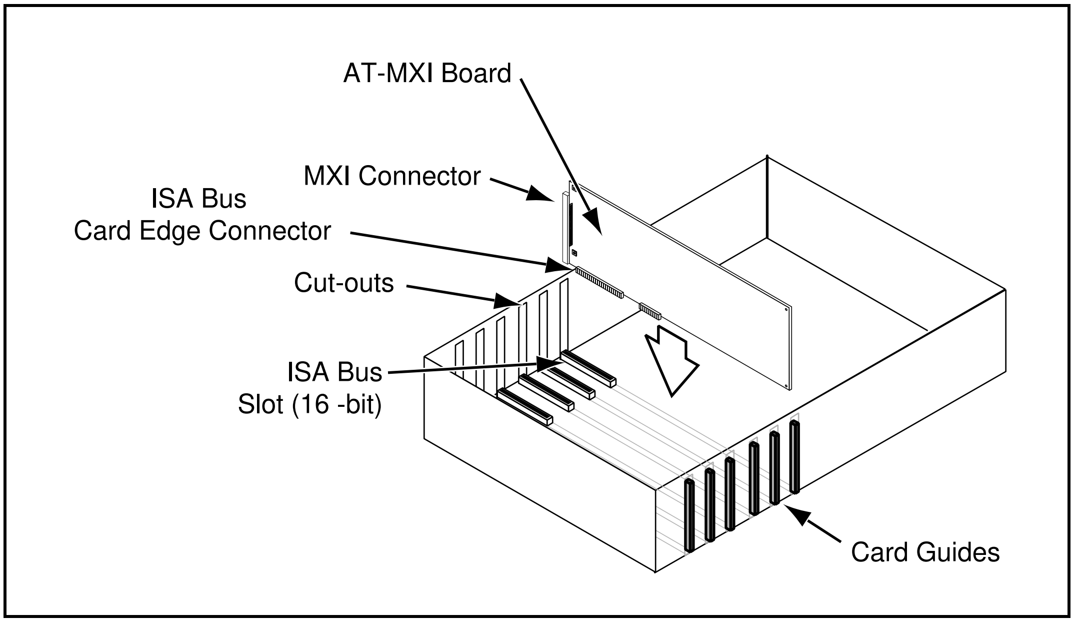


Figure 2-5. AT-MXI Installed in a Computer

Step 4. Configure the VME-MXI Hardware

Figure 2-6 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 plan to install 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.

Art not available in PDF version of document.

Figure 2-6. VME-MXI Parts Locator Diagram

Figure 2-7 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*.

Art not available in PDF version of document.

Figure 2-7. 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 2-8(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 2-8(b).

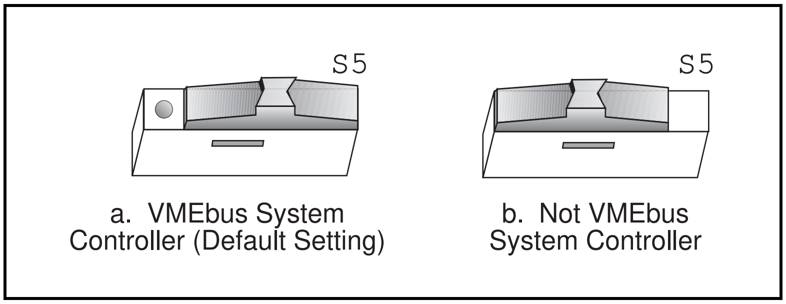

S5
S5
a. VMEbus System Controller (Default Setting)
b. Not VMEbus System Controller

Figure 2-8. VMEbus System Controller Selection

VME Address

The AT-MXI and the VME-MXI use VME short (A16) address space for their configuration, communication, and status registers. Because both boards not only are 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 KB of A16 space to each board. The starting address of a device's address space can be calculated using the following formula:

starting address (A16 space) = (*logical address* * 40h) + C000h

The AT-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 AT-MXI will use A16 address space from C000h through C03Fh. The logical address of the AT-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 AT-MXI or the VME-MXI. If they do, change the other VME devices so that they occupy unique portions of address space. It is recommended that any A16 devices have their operational registers in the lower three-quarters of A16 space (below C000h), leaving the upper quarter for logical addresses. This is required if the VME-MXI is part of a VXIbus system. 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 a PC AT computer with an AT-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 2-9.

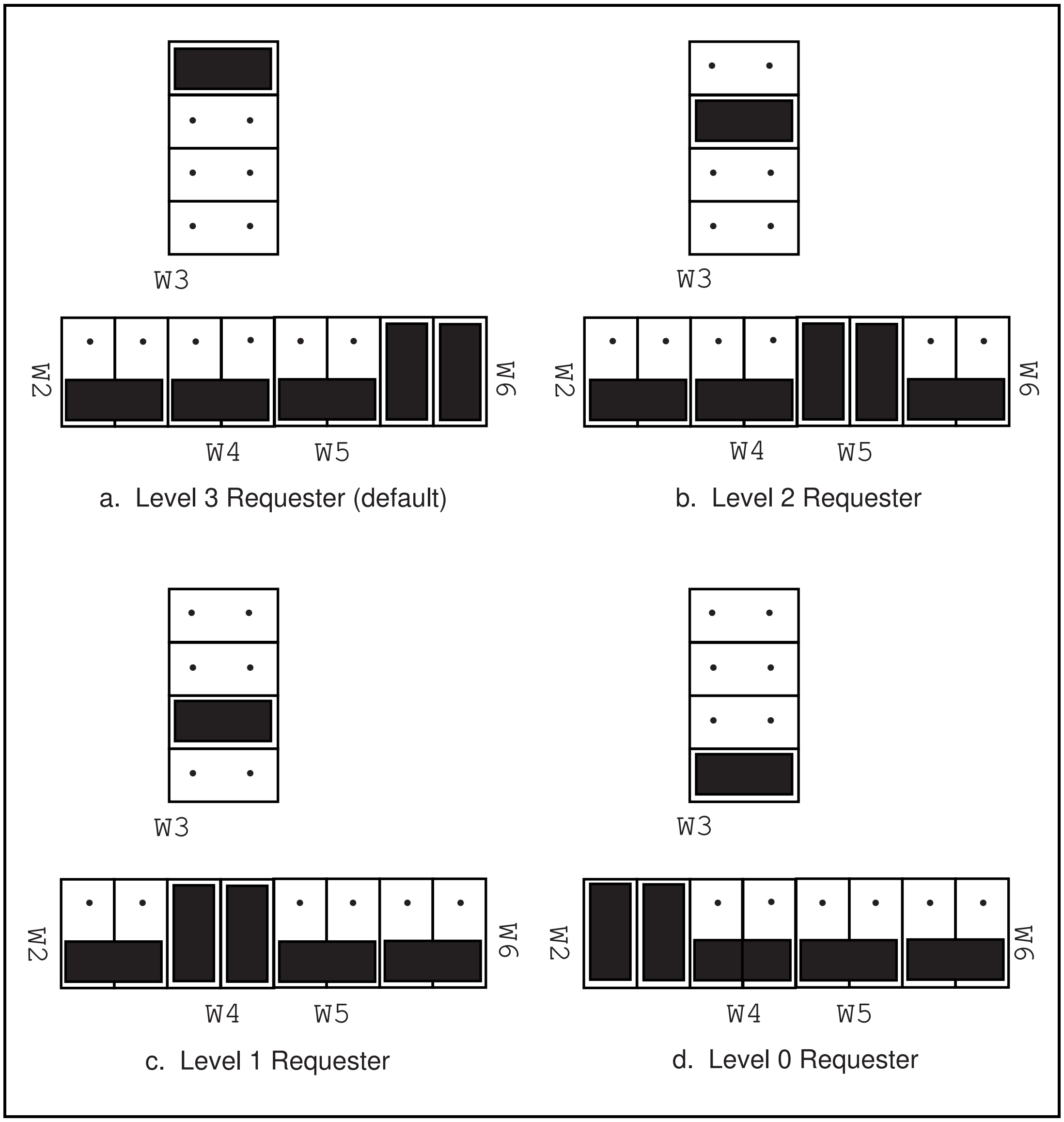


Figure 2-9. VME-MXI VMEbus Requester Jumper Settings

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 VMEbus 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 2-10.

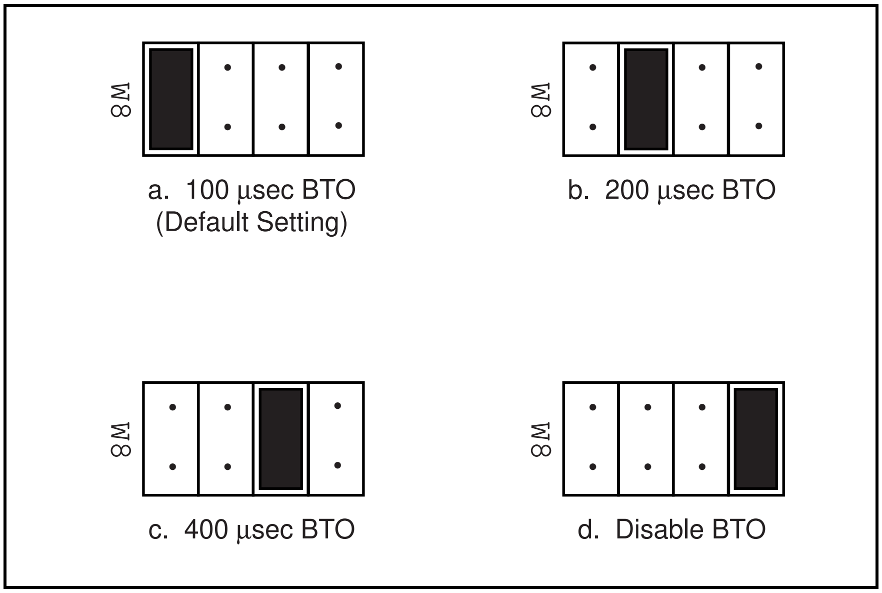


Figure 2-10. VMEbus Timeout Value Selection

Step 5. Install the VME-MXI Hardware

Given below are general installation instructions for the VME-MXI. Please read the following important warnings before installing the VME-MXI. Also consult the user manual or technical reference manual of your VMEbus chassis for specific instructions and warnings.

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

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.

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.

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 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.

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 6. 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-AT2000 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. See Figure 2-11.

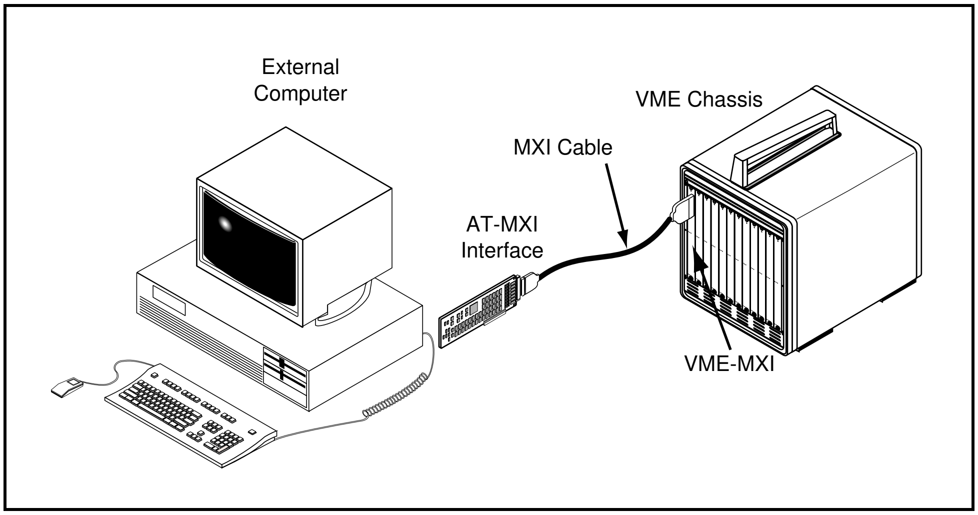


Figure 2-11. MXIbus Single-Ended Cable Configuration

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 AT-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. See Figure 2-12.

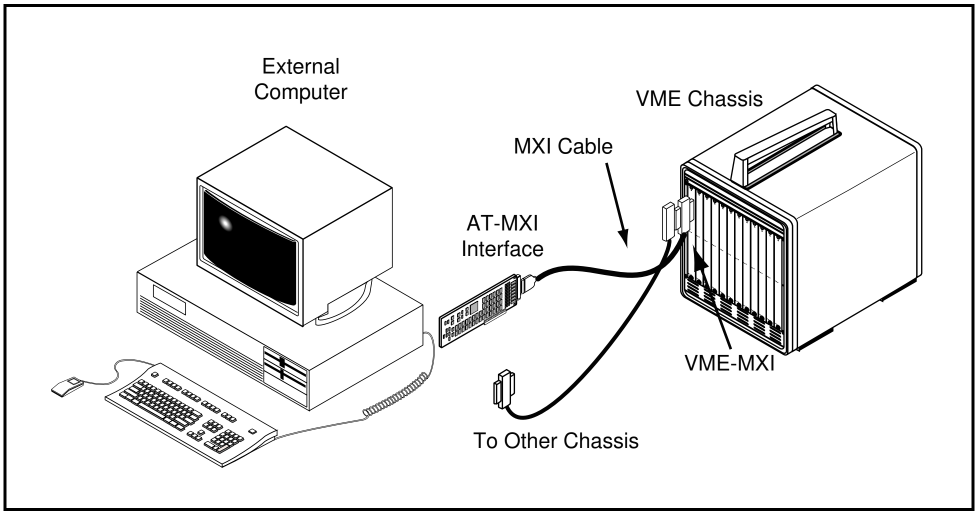


Figure 2-12. MXIbus Dual-Ended Cable Configuration

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

Chapter 3
NI-VXI Software Installation and Configuration

This chapter lists all the programs and files located on the NI-VXI distribution diskettes, contains the instructions for installing the NI-VXI software, and describes how to use the `VXIEDIT` and `VXITEDIT` configuration editors to configure the NI-VXI software. This chapter also discusses programming information for you to consider when developing applications that use the NI-VXI driver.

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-AT2000 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 only apply 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.

NI-VXI Software Overview

The following files make up the NI-VXI software.

Main Programs and Files

The main programs and files of the NI-VXI software package are found in the `C:\NIVXI` directory.

- `VXIINIT.EXE` is the AT-MXI initialization program. This program initializes the board interrupts, shared RAM, VXI register configurations, and bus configurations. `VXIINIT.EXE` can be included in the DOS batch file `AUTOEXEC.BAT` so that the AT-MXI is automatically initialized at startup. The configuration settings can be modified using the `VXIEDIT.EXE` program.

- `RESMAN.EXE` is the National Instruments multimainframe Resource Manager. `RESMAN.EXE` may be executed only after `VXIINIT.EXE` has been run.

- `VIC.EXE` is an 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 programs. This program is described in detail in the *NI-VXI DOS Utilities Reference Manual.*

- VICTEXT.EXE is a text-based interactive control program that is functionally equivalent to VIC.EXE. This program is described in detail in the *NI-VXI Text Utilities Reference Manual.*

- VXIEDIT.EXE is the VXI resource editor program. You 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 PC AT 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 model names of VXI devices and the manufacturer name and ID numbers. This program is described in detail in the *NI-VXI DOS Utilities Reference Manual.*

- VXITEDIT.EXE is the text-based VXI resource editor program that is functionally equivalent to VXIEDIT.EXE. This program is described in detail in the *NI-VXI Text Utilities Reference Manual.*

- README.DOC contains the latest updates and corrections to the manual when appropriate.

Additional Programs and Files

The C:\NIVXI\HLP directory contains various help files used by the VIC.EXE and VXIEDIT.EXE programs.

The C:\NIVXI\TBL directory contains the following files.

- MFNAMEID.TBL contains the database of manufacturer names and their ID numbers. This table file is used only in a hybrid VME/VXI system, because VXI devices have manufacturer IDs, while VME devices do not.

- MODEL.TBL contains the database of model names, manufacturer names, and the model codes numbers. This table file is used only in a hybrid VME/VXI system.

- DEVICE.TBL contains the database of device names, manufacturer names, model names, and frame and slot associations for devices in the system. This table file is used only in a hybrid VME/VXI system.

- NONVXI.TBL contains the 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 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 information. This table file is used only in a hybrid VME/VXI system.
- UTILBUS.TBL contains the utility bus configuration information. This file is used in a VME system but only in a multimainframe situation.
- CREG.TBL contains device-dependent information to be written to the Control register by the Resource Manager. This table file is used only in a hybrid VME/VXI system**.**
- REGS.TBL contains the register names of VXI devices. This table file is used only in a hybrid VME/VXI system**.**
- WSCMDS.TBL contains the VXI Word Serial command values. This table file is used only in a hybrid VME/VXI system**.**
- VXIBUS.CFG contains the AT-MXI MXIbus configuration information.
- VXILA.CFG contains the AT-MXI logical address configuration information.
- VXIDEV.CFG contains the AT-MXI device-specific configuration information.
- VXIMF.CFG, if present, contains information specific to the board manufacturer.
- DVXI.CFG contains information for internal use by the driver.

The C:\NIVXI\INCLUDE directory contains the following include files for the Microsoft C, Borland C, and QuickBASIC language interfaces.

- 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.
- NIVXI.INC is the include file for the Microsoft QuickBASIC Language interface.

The C:\NIVXI\DOS directory contains the following files.

- RESMAND.EXE is used by RESMAN in the DOS environment. RESMAN calls this file directly.
- VICD.EXE is used by VIC in the DOS environment. VIC calls this file directly.
- VICTEXTD.EXE is used by VICTEXT in the DOS environment. VICTEXT calls this file directly.
- VXIEDITD.EXE is used by VXIEDIT in the DOS environment. VXIEDIT calls this file directly.
- VXITEDID.EXE is used by VXITEDIT in the DOS environment. VXITEDIT calls this file directly.

The C:\NIVXI\DOS directory contains up to three subdirectories.

The C:\NIVXI\DOS\BORLANDC subdirectory is installed if you choose to install both the DOS environment and the Borland C libraries. It contains the following file and subdirectory.

- NIVXIDOS.LIB is the NI-VXI Large Memory Model function library for the Borland C Language interface.
- The C:\NIVXI\DOS\BORLANDC\EXAMPLES subdirectory contains various example programs that show how to use the NI-VXI software with Borland C.

The C:\NIVXI\DOS\MSC subdirectory is installed if you choose to install both the DOS environment and the Microsoft C libraries. It contains the following file and subdirectory.

- NIVXIDOS.LIB is the NI-VXI Large Memory Model function library for the Microsoft C Language interface.
- The C:\NIVXI\DOS\MSC\EXAMPLES subdirectory contains various example programs that show how to use the NI-VXI software with Microsoft C.

The C:\NIVXI\DOS\QB subdirectory contains the following object, library, and batch files to generate the Microsoft library for QuickBASIC.

- QBNIVXI.LIB contains the NI-VXI function library for the QuickBASIC interface.
- QBNIVXI.OBJ contains the NI-VXI symbols used to generate the QuickBASIC library for all versions.
- MKQLB*.BAT and MKQLB*.LNK are batch and response files used to generate a QuickBASIC library for the desired version.
- MKAPPBC.BAT and MKAPPBC.LNK are batch and response files that contain an example of how to build a BASIC application using NI-VXI functions.

Also included are the following files containing system functions used in the generation of QuickBASIC libraries for version 4.5 (default) and version 7.0 (7F and 7N).

- QB.OBJ
- QBMEM.OBJ
- QB1.LIB
- QB1_7F.LIB
- QB1_7N.LIB
- QB2.LIB
- B1_71F.LIB
- B1_71N.LIB
- B2_71.LIB

Installing the Software

The NI-VXI distribution disks that came with your AT-MXI contain the NI-VXI files as well as an INSTALL program. The INSTALL program is used to install the entire NI-VXI software package, a software update, or to reinstall software in the event that your files were accidentally erased. Follow these steps to install all or part of the NI-VXI software.

1. The NI-VXI software requires approximately 2 MB of free space on your hard disk. Create the necessary free space on the hard disk before starting the installation.

2. Run INSTALL.EXE on distribution disk 1. INSTALL is a text-based, interactive, self-guiding program, which will install all the necessary files (for a complete or partial installation, as user-specified) on the hard disk. In addition, it will optionally modify the AUTOEXEC.BAT initialization file accordingly. The program prompts you to enter the following information:

   - Whether to install NI-VXI for DOS: If you need to add only a certain component (for example, QuickBASIC) but already have the NI-VXI package installed, you can respond *No*. The default is to install the DOS applications.

   - Languages to install: You can choose whether to install the Microsoft C, Borland C, and/or QuickBASIC libraries. The default is to install all of them, but if you are using only one compiler, you can select only that compiler. However, if you do install both C libraries, you must let INSTALL know which one to make the default library.

   - Source drive: This is the floppy drive in which you inserted the distribution disks.

   - Destination directory: This is the complete pathname of the directory where you want to install the software.

   - Modifying DOS initialization files: You can choose whether INSTALL modifies the AUTOEXEC.BAT file to reflect the location of the NI-VXI library. If you choose to let INSTALL do this, you must also enter the drive where AUTOEXEC.BAT can be found.

   If INSTALL detects a previous installation of the NI-VXI software, it will prompt you to determine whether to delete or overwrite the previous version, or to provide an alternate path in which to install the new version.

Note: INSTALL ***will not create a backup for you. If there are files you want to preserve, be sure to exit the installation program at this time and make a backup before continuing.***

You can quit the INSTALL program at any time by choosing **Exit** from the initial screen, or by pressing <Control-C> once installation has begun. INSTALL will prompt you to be certain you really want to quit at that time. If you choose to run the INSTALL program again, it starts over and recopies all the necessary files. Of course, if you made any changes from the default configuration, you will need to specify those changes once again.

As mentioned previously, INSTALL *optionally* modifies the AUTOEXEC.BAT file. If you *do* modify it, INSTALL changes the old file's extension to .BAK and keeps it for safety purposes. If you do *not* modify it, INSTALL creates a file in the NI-VXI directory with the changes it proposes you make to the file, and names it AUTOEXEC.VXI.

Modifying the AUTOEXEC.BAT File

If you choose to let INSTALL modify your AUTOEXEC.BAT file, it updates the setting of environment variables PATH, LIB, and INCLUDE to specify the relevant subdirectories of the NI-VXI directory. The previously specified directories in PATH, LIB, and INCLUDE remain unchanged. INSTALL also adds a new environment variable NIVXIPATH, and appends a command line to execute VXIINIT.EXE automatically.

If you choose not to let INSTALL modify your AUTOEXEC.BAT file, refer to the AUTOEXEC.VXI file in your NI-VXI directory for suggestions on how to change the following lines manually.

- The PATH variable should include the full path to the subdirectory where the NI-VXI utilities are located, in addition to whatever other directories you have already specified in PATH. Normally, these files reside in the root of the NI-VXI directory.

- The LIB variable should include the full path to the subdirectories that contain the C libraries for the compiler you choose to install.

- The INCLUDE variable should include the full path to the subdirectory that contains the NI-VXI include files. By default, the include files reside in the INCLUDE subdirectory of the NI-VXI directory.

- The NIVXIPATH variable should contain the full path to the NI-VXI directory.

Notes: ***After you execute*** INSTALL***, you should reboot your machine (or at least run*** AUTOEXEC.BAT***) to make your system aware of the NI-VXI directory.***

Once the NI-VXI package is installed, run VXIINIT.EXE ***and then*** RESMAN.EXE.

Configuring the NI-VXI Software

Run VXIINIT to display the AT-MXI configuration settings. The AT-MXI is configured in hardware as Resource Manager (Logical Address 0) and Message-Based device, and cannot be changed. The default configuration is MXIbus System Controller, not MXIbus fair requester, base I/O address of 340h, master window base of D000h, and no VXI Shared RAM. To change the current settings, or to view the configuration options, run VXIEDIT and select the Configuration Editor from the main menu.

You can use the Non-VXI Device Editor from the main menu 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. It is recommended 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 can also enter manufacturer name, model name and other identification information for your VME devices. 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 only use the configuration information for 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 and device names.

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 VXIEDIT to configure the address space requirements, the Resource Manager can automatically enable the hardware windows through which your PC AT can access the corresponding VME addresses where your VME devices reside. You can also run VXIEDIT at any time to examine the software configurations.

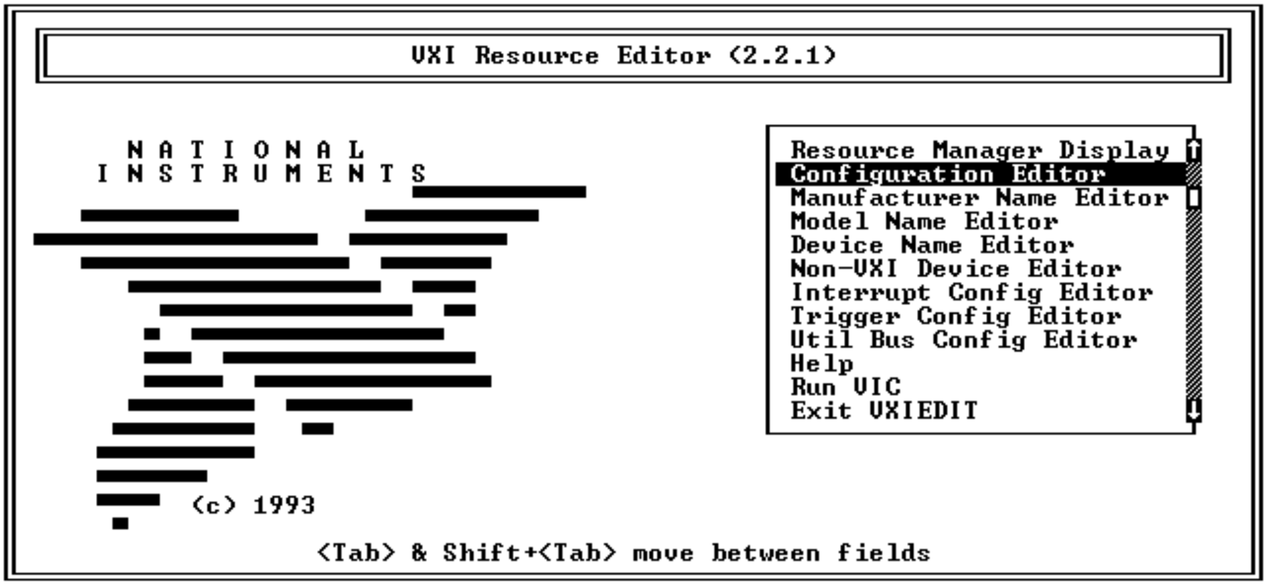
You can also use this program to configure interrupts and the utility bus for single-mainframe or multiple-mainframe system configurations. For more details on how to use VXIEDIT, refer to the remainder of this chapter and to the *NI-VXI DOS Utilities Reference Manual.*

Running VXIEDIT

VXIEDIT.EXE is the VXI resource editor program. You can use this program to configure the system, edit the manufacturer name and ID numbers, and edit the model names of VXI and non-VXI devices in the system, as well as the system interrupt configuration information. This program also displays the system configuration information generated by the Resource Manager.

To run VXIEDIT from any directory, make sure that the PATH environment variable has the destination directory of the NI-VXI software added to it. This program uses the different configuration files (*.CFG), table files (*.TBL), and help files (*.HLP) in its execution. Be sure that the environment variable NIVXIPATH is set to the destination directory of the NI-VXI software for proper execution. The default destination directory pathname used by the program is C:\NIVXI.

Figure 3-1 shows the functions available under `VXIEDIT`.



```
VXI Resource Editor (2.2.1)
```

```
(c) 1993
<Tab> & Shift+<Tab> move between fields
```

Figure 3-1. VXIEDIT Main Menu

The Configuration Editor, Non-VXI Device Editor, and Resource Manager Display are used in VME systems. Use the Configuration Editor to configure details of the AT-MXI hardware configuration. Use the Non-VXI Device Editor to configure the address requirements of VME devices in your VME system so that the Resource Manager program can automatically set up the MXI link. The Resource Manager Display lets you examine the status of the address space configuration after the Resource Manager program has executed. The Manufacturer Name Editor, Model Name Editor, Device Name Editor, and Trigger Configuration Editor are specific to VXI systems and are not used in VME systems (VXI is a superset of VME). The Interrupt Configuration Editor and Utility Bus Configuration Editor can be used in VME systems, but only in multimainframe system configurations.

Configuration Editor

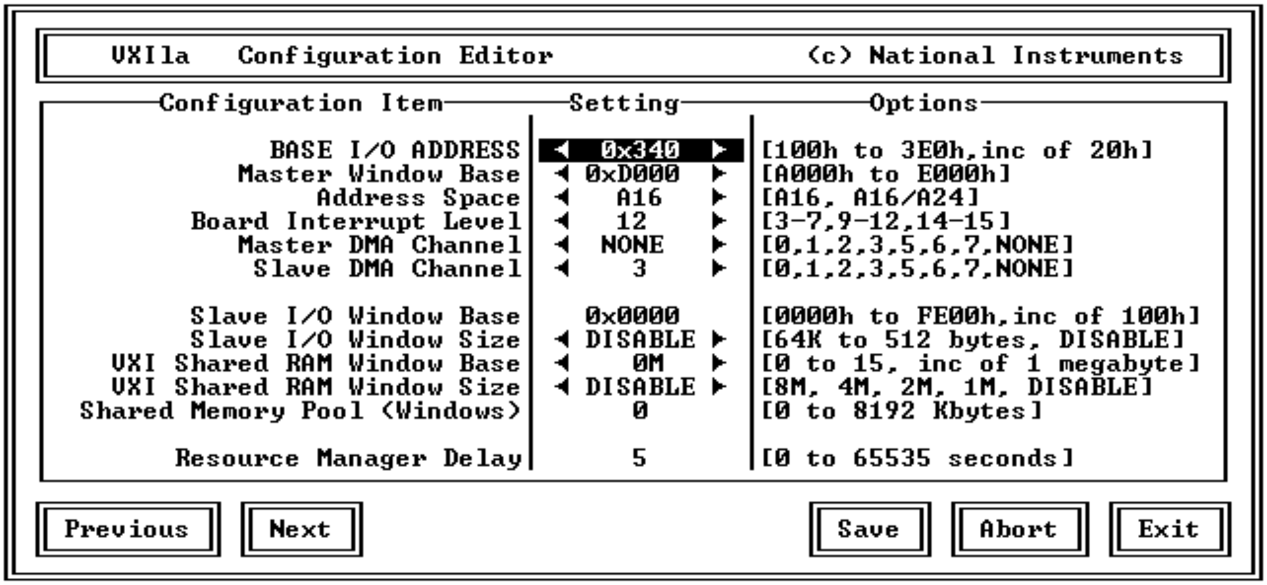
When you select the Configuration Editor from the main menu, the Logical Address Configuration Editor menu (shown in Figure 3-2) is displayed. The following sections describe the various configuration options available under the Logical Address, Bus, and Device Configuration menus.

Select **Next** and **Previous** to move between the three configuration editors. You can change the default settings to any value within the range shown to the right of each item by using the left and right arrow keys.

You can save your changes by selecting **Save**. Saving the changes updates files `VXILA.CFG`, `VXIBUS.CFG`, `VXIDEV.CFG`, and `DVXI.CFG`. These files are used by `VXIINIT.EXE` to update the soft-configured settings, including board interrupts, shared RAM size, VXI register contents, and bus interface. You can exit the program at any time without saving any changes you have made by selecting **Abort**.

Logical Address Configuration

Figure 3-2 shows the Logical Address Configuration Editor (VXIla) menu.



```
           Configuration Item        Setting              Options

             BASE I/O ADDRESS   ◄  0x340  ►  [100h to 3E0h,inc of 20h]
           Master Window Base   ◄ 0xD000  ►  [A000h to E000h]
                Address Space   ◄   A16   ►  [A16, A16/A24]
         Board Interrupt Level  ◄   12    ►  [3-7,9-12,14-15]
           Master DMA Channel   ◄  NONE   ►  [0,1,2,3,5,6,7,NONE]
            Slave DMA Channel   ◄   3     ►  [0,1,2,3,5,6,7,NONE]

         Slave I/O Window Base     0x0000    [0000h to FE00h,inc of 100h]
         Slave I/O Window Size  ◄ DISABLE ►  [64K to 512 bytes, DISABLE]
     VXI Shared RAM Window Base ◄    0M   ►  [0 to 15, inc of 1 megabyte]
     VXI Shared RAM Window Size ◄ DISABLE ►  [8M, 4M, 2M, 1M, DISABLE]
   Shared Memory Pool (Windows)      0       [0 to 8192 Kbytes]

        Resource Manager Delay       5       [0 to 65535 seconds]
```

Figure 3-2. AT-MXI Logical Address Configuration Editor

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
AT-MXI Base I/O Address	Indicates the I/O Address of the AT-MXI configuration registers. This value must correspond to the selected jumper settings on the board. Range = 100h to 3E0h in increments of 20h (hex)	0x340
Master Window Base	A 64 KB memory-mapped I/O window, which the AT-MXI board uses to access the VMEbus. Range = A000h to E000h in increments of 1000h (hex)	0xD000
Address Space	Indicates the address spaces used by the AT-MXI. A16 only A16/A24	A16 only
AT-MXI Board Interrupt Level	Indicates the interrupt level used by the board to interrupt the processor. This value must correspond to the jumper settings on the AT-MXI board. Range = 3, 4, 5, 6, 7, 9, 10, 11, 12, 14, 15	12
Master DMA Channel	Indicates the DMA channel that the AT-MXI uses to perform master mode block transfers onto the VMEbus. This value must correspond to the jumper settings on the AT-MXI board. The NONE option can be selected with any jumper settings. Range = 0, 1, 2, 3, 5, 6, 7, NONE **Note:** ***This field is currently disabled.***	NONE

(continues)

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

Characteristic	Description	Default Value
Slave DMA Channel	This indicates the slave DMA channel to be used in all slave accesses to the VME Shared RAM in A24 space or the PC I/O in A16 space. This value must correspond to jumper settings on the AT-MXI board. The NONE option can be selected with any jumper settings. Range = 0, 1, 2, 3, 5, 6, 7, NONE **Note:** ***If the NONE option is selected, slave accesses to VME Shared RAM and to PC I/O space will not succeed. Accesses to VXI-defined registers on the AT-MXI board, however, remain unaffected.***	3
Slave I/O Window Base	The I/O space in the PC with the AT-MXI can be accessed in A16 space over the VMEbus. This field sets up the base address for these accesses. Range = 0000h to FE00h in increments of 100h (hex) **Note:** ***Never select a base address above C000h. Selecting an address above C000h can conflict with the existing VXI and VME devices in the system. Because the VXIbus specification does not address automatic configuration of the lower 48 KB of A16 space,*** `RESMAN` ***does not handle any device requesting memory in A16 space, including the AT-MXI.*** `RESMAN` ***does not give a warning in the event of an address conflict in A16 space. Also, the opening of A16 windows on frame extenders in the system for A16 accesses depends only on the logical addresses of the devices in the system.***	0x0000
Slave I/O Window Size	Indicates the size of the window in A16 space for accesses to the PC I/O space over the VMEbus. Range = 64 KB, 32 KB, 16 KB, 8 KB, 4 KB, 2 KB, 1 KB, 512 bytes, DISABLE	DISABLE

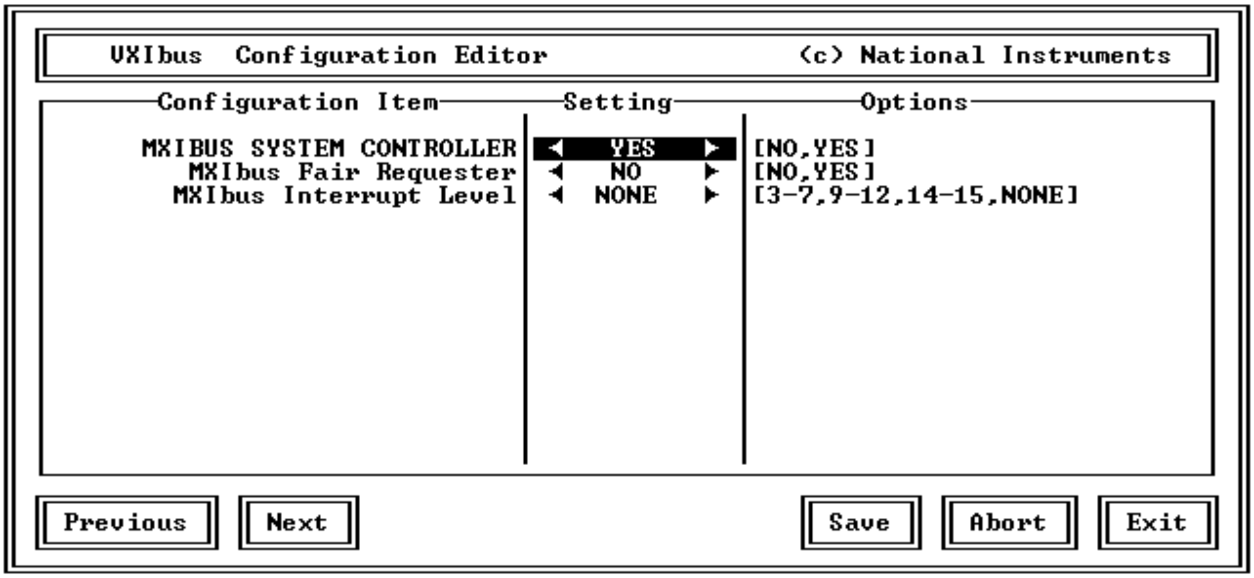
(continues)

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

Characteristic	**Description**	**Default Value**
Shared RAM Window Base	Indicates the offset in the local RAM above which the RAM is shared over the VMEbus in A24 space. If the **Address Space** selection for the AT-MXI is A16 only, this field is ignored. Range = 0 to 15, in increments of 1MB	0 MB
Shared RAM Window Size	Indicates the size of VME Shared RAM in the A24 space. If the **Address Space** selection for the AT-MXI is A16 only, this field is ignored. Range = 8 MB, 4 MB, 2 MB, 1 MB, DISABLE **Note:** ***If the* Address Space *selection is A16/A24, you need to select a value other than* DISABLE *for this field. You must also ensure that the Slave DMA channel is set correctly.***	DISABLE
Shared Memory Pool (Windows)	Indicates the size of memory in kilobytes that is allocated on Windows startup to be used exclusively by `VXImemAlloc` function calls from both DOS and Windows NI-VXI applications. If you do not have Microsoft Windows installed on your system, or if you will not be running DOS NI-VXI applications under Windows, set this field to 0 KB. Range = 0 to 8192 KB	0 KB
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 65535 s	5 s

Bus Configuration

Figure 3-3 shows the Bus Configuration Editor (VXIbus) menu.



```
            MXIBUS SYSTEM CONTROLLER  ◄   YES   ►  [NO,YES]
              MXIbus Fair Requester   ◄   NO    ►  [NO,YES]
              MXIbus Interrupt Level  ◄  NONE   ►  [3-7,9-12,14-15,NONE]
```

Figure 3-3. AT-MXI Bus Configuration Editor

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

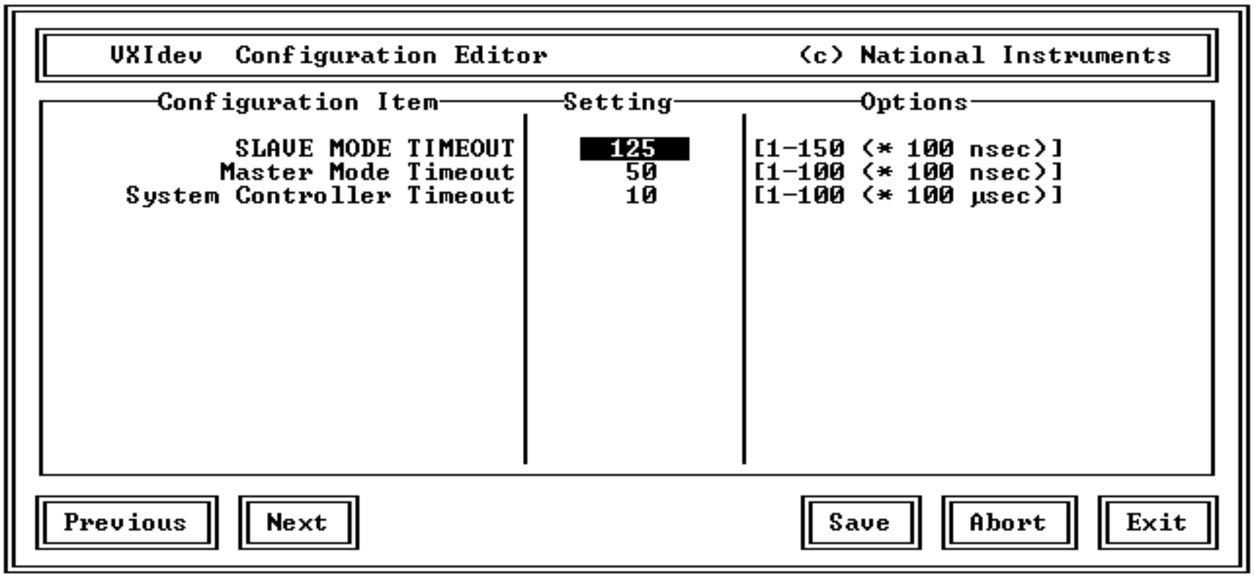
MXIbus System Controller	A MXIbus link must have a single device that is responsible for MXIbus interrupt and bus arbitration as well as bus timeouts. This device is referred to as the MXIbus System Controller and is always the first device in the MXIbus daisy-chain. The AT-MXI is always designated as the MXIbus System Controller when used in this kit and should *not* be changed.
	NO = Not MXIbus System Controller
	YES = MXIbus System Controller
		YES

MXIbus Fair Request	MXIbus fair 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 AT-MXI to be an unfair requester and should only be changed if additional MXIbus devices are added to the system. Refer to the user manual included with your additional MXIbus devices for more information.
	NO = Not a MXIbus Fair Requester
	YES = MXIbus Fair Requester
		NO

MXIbus Interrupt Level	Indicates the interrupt level used by the MXI IRQ to interrupt the processor. This value must correspond to jumper settings on the AT-MXI board; however, the NONE option can be selected with any jumper settings. If the NONE option is selected, the same interrupt level as used for AT-MXI board interrupts is used for this interrupt. Range = 3, 4, 5, 6, 7, 9, 10, 11, 12, 14, 15, or NONE	NONE

Device Configuration

Figure 3-4 shows the Device Configuration Editor (VXIdev) menu.



VXIdev Configuration Editor (c) National Instruments

Configuration Item	Setting	Options
SLAVE MODE TIMEOUT	125	[1-150 (* 100 nsec)]
Master Mode Timeout	50	[1-100 (* 100 nsec)]
System Controller Timeout	10	[1-100 (* 100 µsec)]

Figure 3-4. AT-MXI Device Configuration Editor

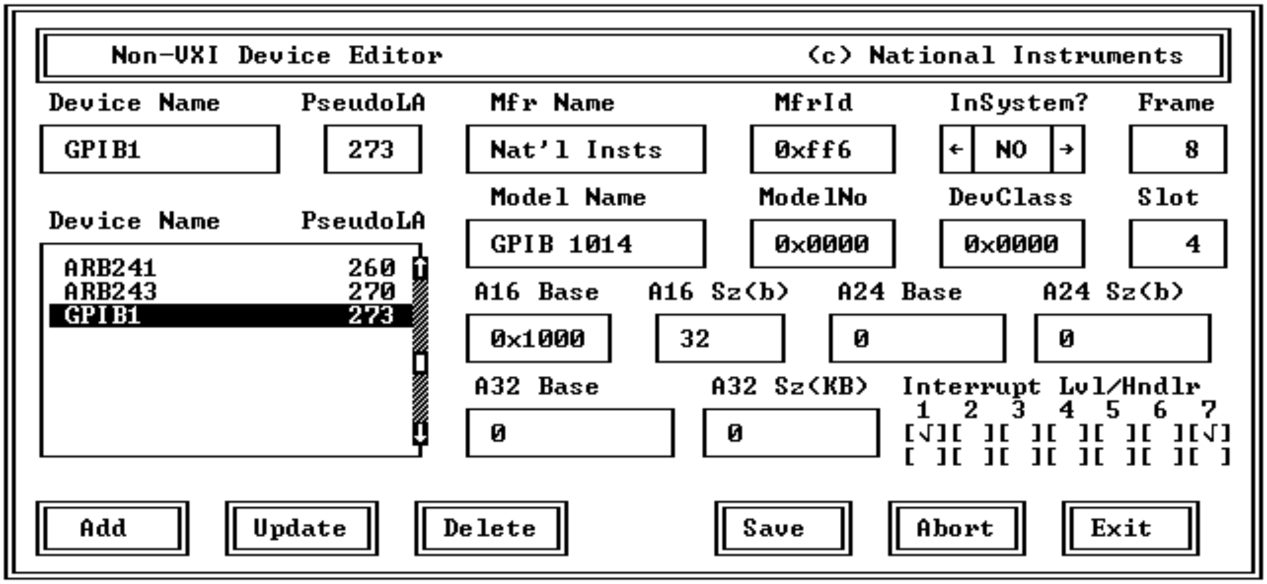
Table 3-3 describes the characteristics of the options available under the Device Configuration menu.

Table 3-3. Device Configuration Characteristics

Characteristic	Description	Default Value
Slave Mode Timeout	This field gives the maximum amount of time a slave access to the VME Shared RAM may hold the PC bus. The units are in hundreds of nanoseconds.	125 (12.5 µs)
Master Mode Timeout	This field gives the maximum amount of time a master access from the PC to the MXIbus may wait until it has to be retried. The units are in hundreds of nanoseconds.	50 (5.0 µs)
System Controller Timeout	This field gives the maximum amount of time a master access from the PC to the MXIbus may be allowed to continue without being terminated as a bus error. The units are in hundreds of microseconds. **Note:** ***If the* MXIbus System Controller *field in the Bus Configuration Editor is set to* NO, *this field is ignored.***	10 (1.0 ms)

Non-VXI Device Editor

Select the Non-VXI Device Editor from the main menu. Figure 3-5 shows the configuration available from this menu.



```
 Non-VXI Device Editor                         (c) National Instruments

 Device Name      PseudoLA   Mfr Name          MfrId       InSystem?     Frame
 GPIB1            273        Nat'l Insts       0xff6       ← NO →        8
                             Model Name        ModelNo     DevClass      Slot
 Device Name      PseudoLA   GPIB 1014         0x0000      0x0000        4
 ARB241           260        A16 Base    A16 Sz(b)   A24 Base      A24 Sz(b)
 ARB243           270        0x1000      32          0             0
 GPIB1            273        A32 Base         A32 Sz(KB)   Interrupt Lvl/Hndlr
                                                            1  2  3  4  5  6  7
                             0                0            [√][ ][ ][ ][ ][ ][√]
                                                           [ ][ ][ ][ ][ ][ ][ ]

 Add    Update    Delete         Save    Abort    Exit
```

Figure 3-5. Non-VXI Device Editor

To add or modify the address space and interrupt requirements for VME devices in your system, select the Non-VXI Device Editor from the main menu. 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, select the **Non-VXI Device Editor** field on the main menu and press <RETURN> or <Enter>. Refer also to the *NI-VXI DOS Utilities Reference Manual* for more information about this editor.

You need to assign VME devices pseudo-logical addresses (between 256 and 511 inclusive), as well as device names, manufacturer names, and pseudo-model names. The **Frame** field should contain the logical address of the VME-MXI in the chassis. The default logical address is 1. A16 VME devices should be configured to occupy memory below C000h in A16 space.

Select **Add** to add a new entry to the alphabetical list of device names. After modifying an entry, select **Update** to replace the original entry. To delete an entry, select the entry in the list of device names and then select **Delete**. Select **Save** to save the information that was changed or select **Abort** to ignore the modifications. **Exit** returns to the main menu. Make sure to set or clear the **InSystem?** field of an entry (by typing a `y` for yes or an `n` for no), depending on whether that device is actually present in your system.

In the example in Figure 3-5, we have a VME memory device in the system that requires 16 bytes of memory in A16 space. We 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 A16 memory base and size fields according to its memory requirements, and set the **`InSystem?`** field.

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 VXIEDIT and Reinitializing the Hardware

Select **Exit** or press the <Esc> key to quit the `VXIEDIT` program. If you changed any of the software configuration information, `VXIEDIT` prompts you to save your changes before exiting the configuration menu. Saving the configuration information updates the configuration files `VXILA.CFG`, `VXIBUS.CFG`, `VXIDEV.CFG`, and `DVXI.CFG`. Run `VXIINIT.EXE` to reinitialize the hardware according to the new settings.

If you changed any of the software configuration settings from the default, record the new setting on the *VME-AT2000 Hardware and Software Configuration Form* in Appendix C.

Developing Your Application Program

The easiest way to learn how to communicate with your instruments is by controlling them interactively. Use the VXI interactive control program (`VIC` or `VICTEXT`) to write to and read from your instruments. Both programs display the status of your VME transactions and inform you of any errors that occur. Refer to the *NI-VXI DOS Utilities Reference Manual* on how to use `VIC` and to learn about its features, and to the *NI-VXI Text Utilities Reference Manual* on how to use `VICTEXT` and to learn about its features.

`NIVXIDOS.LIB` is the NI-VXI Large Memory Model function library for the Microsoft C and Borland C Language interfaces. To use the NI-VXI software in your application program, you must compile your program for the large memory model, and include this library along with other libraries at link time. Make certain that the environment variables `LIB` and `INCLUDE` are set correctly as described in the *Installing the Software* section earlier in this chapter. The include files for the NI-VXI software are in the `INCLUDE` subdirectory.

Note: ***It is essential to include*** `"NIVXI.H"` ***before any other include file in your application program.***

The `EXAMPLES` subdirectories contain various example programs along with makefiles that show how to use various functions in the NI-VXI software and how to develop application programs using these functions.

Note: ***When compiling NI-VXI applications, you must define the macro*** `VXIDOS` ***in your makefile/project. Refer to the example programs for details.***

`QBNIVXI.LIB` is the file that contains the NI-VXI function library for the Microsoft QuickBASIC interface. This is used to generate `QBNIVXI.QLB`, the Microsoft QuickBASIC library. The corresponding include files `*.INC` are in the `INCLUDE` subdirectory in the distribution software.

Appendix A
Specifications

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

AT-MXI

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

Capability Codes

MXIbus

Capability Code	Description
MA32	Master Mode A32, A24 and A16 addressing
MBLT	Master Mode block transfers
SA24	Slave Mode A24 and A16 addressing
SBLT	Slave Mode block transfers
MD16	Master Mode D16 and D08 data sizes
SD16	Slave Mode 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

ISA Bus

Capability Code	Description
AM	Can function as an AT Alternate Master
LOCK	Can lock the AT bus for indivisible transfers
DMA16	Supports D08 or D16 DMA transfers
INT	Can interrupt on the PC AT bus

Requirements

Characteristic	Specification
Memory Space	64 KB
I/O Space	32 B

Environmental

Characteristic	Specification
Component Temperature	0° to 70° C operating; -55° to 150° C storage
Relative Humidity	0% to 95% noncondensing, operating; 0% to 100% noncondensing, storage
Emissions	FCC Class A
Safety	Not applicable
Shock and Vibration	Not applicable

Physical

Characteristic	Specification
Board Dimensions	Standard full-length AT-height board 13.36 in. by 4.8 in. (339.72 mm by 121.92 mm)
Connectors	Single fully implemented MXIbus connector
Slot Requirements	Single AT (ISA) slot
MTBF	Contact Factory

Electrical

Source	Typical	Direct Current (max)
+5 VDC	3.3 A	4.4 A

Timing

Master Mode	
Transfer Type	**Transfer Rate**
Write	530 ns
Read	430 ns
Block Write	290 ns
Block Read	190 ns

Master Mode	
Transfer Type	**Transfer Rate**
Write	840 ns
Read	840 ns
Block Write	590 ns
Block Read	590 ns

Other	
Transfer Type	**Transfer Rate**
Daisy-Chain Delay (Passing GIN to GOUT or GOUT generation from System Controller)	120 ns

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

Requirements

Characteristic	Specification
A16 Space	64 B

Environmental

Characteristic	Specification
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

Characteristic	Specification
Board Dimensions	9.187 in. by 13.386 in. (233.35 mm by 340 mm)
Connectors	Single fully implemented MXIbus connector Single INTX connector (on boards equipped with optional INTX daughter card)
Slot Requirements	Single slot
Compatibility	Fully compatible with VMEbus specification
VXI Keying Class	Class 1 TTL
MTBF	Contact Factory

Electrical

Source	Typical	Maximum
+5 VDC	5.25 A	7.0 A

Timing

Master Mode	
Transfer Type	**Transfer Rate**
Write	675.5 ns
Read	625.5 ns
Block Write	320 ns
Block Read	270 ns

Slave Mode	
Transfer Type	**Transfer Rate**
Write	381 ns
Read	381 ns
Block Write	238 ns
Block Read	238 ns

Other	
Transfer Type	**Transfer Rate**
Daisy-Chain Delay (Passing GIN to GOUT or GOUT generation from System Controller)	120 ns

Appendix B
Troubleshooting

This appendix addresses certain problems you may encounter when using the NI-VXI bus interface software on the AT-MXI platform. This material covers the following issues:

- Running the VXIINIT initialization program results in an error message or causes the system to hang.
- Running the RESMAN Resource Manager program causes the system to hang or results in incorrect data.
- The mouse behaves erratically after you install the NI-VXI software.

If you encounter any of these problems, your system may have a conflict with one of the AT-MXI configuration settings or you may have a problem with the MXIbus cable(s). For proper operation, the AT-MXI requires the allocation of the following system resources:

- Thirty-two bytes of I/O register space
- At least one interrupt level that cannot be shared with another device
- A DMA channel (optional)
- A contiguous 64 KB block of memory space in the upper memory region of the PC AT.

The following sections discuss each of these system requirements along with the possible problems that may be encountered with a bad MXI cable.

If you are unable to resolve your problem, please fill out the appropriate forms in Appendix C, *Customer Communication*, and contact National Instruments.

Problems with VXIINIT

This section discusses what to do if running the VXIINIT initialization program results in an error message or causes the system to hang.

VXIINIT Error Messages

VXIINIT can give you error messages that inform you that it could not find the AT-MXI, that it could not find a specific file, or that an error occurred while writing a log file.

VXIINIT Could Not Find the AT-MXI

When you run `VXIINIT`, it attempts to locate the AT-MXI board by accessing the I/O registers that exist on the board. The AT-MXI board occupies 32 bytes of I/O space at a default base address of 340h (hex). The base I/O address of the AT-MXI is set in both hardware and software and the two settings must agree. If the hardware and software settings do not agree, you may receive an error message that reads **`VXI device not found`**.

In hardware, you use a set of DIP switches on the AT-MXI board itself if you need to change the setting of the base I/O address. Refer to the *Base I/O Address Selection* section of Chapter 2, *Hardware Configuration and Installation*, for more information about selecting a base I/O address for the AT-MXI. Keep in mind that if you change the base I/O address of the AT-MXI, you must make a corresponding change to the AT-MXI configuration software.

To change the I/O address in software, you need to run the VXI resource editor program, either `VXIEDIT` or `VXITEDIT`, and use the Logical Address Configuration Editor to make the change. Refer to the *Configuring the NI-VXI Software* section of Chapter 3, *NI-VXI Software Installation and Configuration*, for instructions on how to use the VXI resource editor program.

After you verify that the software settings correspond to the hardware setting, try the following methods to deal with this problem.

1. Ensure that you are using the proper `VXIINIT` program for your system. For example, the `VXIINIT` program distributed with the MC-MXI cannot be used to initialize the AT-MXI.
2. If the hardware and software settings match, ensure that the I/O registers settings of the AT-MXI do not conflict with any other hardware in the PC AT. If they do, change the settings of the AT-MXI or the other device.
3. If the hardware and software settings correspond and do not conflict with any other hardware in the PC AT, make sure the AT-MXI is securely seated in the ATbus slot or try another ATbus slot.
4. If the message persists, there is still some reason that the NI-VXI software cannot access the I/O registers of the AT-MXI. If possible, try the AT-MXI in another computer.

VXIINIT Could Not Find Specific Files

`VXIINIT` may inform you that it cannot find one or more of the configuration or table files, for example, **`File VXILA.CFG not found.`** Other files for which error messages are generated include `VXIBUS.CFG`, `VXIDEV.CFG`, `DVXI.CFG`, `MFNAMEID.TBL`, and `MODEL.TBL`. There are two possible causes for this error message.

1. The file(s) may have been deleted accidentally. If the file does not exist in your `TBL` directory after you installed your software, you can find it on your original NI-VXI distribution media and restore it manually.
2. The `NIVXIPATH` variable may be pointing to the wrong directory. The path should be set in the `AUTOEXEC.BAT` file to point to the root directory of the NI-VXI files. By default, this directory is `C:\NIVXI`.

Error Writing Log File

Another VXIINIT error message is **Error writing configuration to file.** This is only a warning message, and does not indicate a problem with the configuration. Either the log file could not be created, or only a partial log was written.

VXIINIT creates a file VXIINIT.OUT in the TBL subdirectory. This file is a text file that contains the same configuration information you see printed to the screen. The likely cause of this message is that the disk drive is full.

VXIINIT Causes System to Hang

If the system hangs when you run VXIINIT, first make sure that the I/O registers of the AT-MXI do not conflict with any other hardware in the PC AT, as discussed in the previous section. If this is not the case and the problem still occurs, the most likely reason is a DMA channel conflict.

In its default configuration, the AT-MXI uses one DMA channel, channel 3, for slave accesses. You can change this channel by using the Logical Address Configuration Editor of the VXIEDIT or VXITEDIT program and modifying the **Slave DMA Channel** field. Try setting this parameter to NONE, saving the new configuration, and running VXIINIT. If the system does not hang, the problem was due to a DMA channel conflict.

An application uses slave DMA when it needs to dual-port PC memory so that it can be accessed from the VXI/VME chassis. If the PC is simply going to read and write from devices, it does not need slave DMA and you can keep the parameter in the NONE setting. If your application needs to have slave accesses to the PC, you need to find an available DMA channel that you can set up successfully. Follow these steps to find a slave DMA channel that will work:

1. Determine if there are other devices that are configured to use DMA channels.

2. Choose a level that is not used in your system and set the AT-MXI to use this level. Refer to the *DMA Channel Selection* section of Chapter 2 for instructions on making this hardware change.

3. After you change the AT-MXI slave DMA channel jumper, use the Logical Address Configuration Editor of VXIEDIT or VXITEDIT to enter the new slave DMA channel.

4. Save the changes and run VXIINIT.

As mentioned previously in the *Master Mode Versus Slave Mode* section of Chapter 2, the NI-VXI software does not use master-mode DMA for block transfers because the movs instruction is faster on most PCs. The latest version of the NI-VXI software sets the default value for the **Master DMA Channel** field in the Logical Address Configuration Editor to NONE, and this value cannot be modified. If you are using a version of the NI-VXI software for the AT-MXI prior to Version 3.10, ensure that this value is set to NONE.

Problems with RESMAN

If the `VXIINIT` program runs successfully but you encounter problems with `RESMAN`, the Resource Manager program, the cause is probably the Master Window. The Master Window is a 64 KB block of contiguous memory in the PC AT upper memory space that the AT-MXI uses to access the VXI/VME bus. The upper memory space of a PC AT is the region from 640 KB (A000h) to 1024 KB (FFFFh). Several system resources, such as system ROM and video memory, reside in this area. By default, the Master Window occupies a memory region from a base address of D000h to DFFFh. You can change the Master Window base address by running the VXI resource editor program, either `VXIEDIT` or `VXITEDIT`, and using the Logical Address Configuration Editor to make the change.

A conflict in the Master Window of the AT-MXI can cause `RESMAN` to behave erratically. The following are likely symptoms of a conflict in the Master Window configuration.

- The `RESMAN` display shows multiple devices at Logical Address 0. (The AT-MXI should be the *only* device at Logical Address 0.)
- `RESMAN` detects nonexistent devices.
- The computer hangs when you run `RESMAN`.

The following paragraphs discuss the two main causes of a Master Window Base Address conflict.

Memory Area Used by Another Device

Your system may contain a device that already occupies all or some of the upper memory area that is allocated for the Master Window.

To solve this problem, first determine what area of upper memory your devices use. Then move either your device or the Master Window Base Address to an unused area of upper memory.

If you have a Super VGA monitor, your system is likely to use video RAM. Generally, any video RAM used by your PC AT occupies all or some of the upper memory from A000h to BFFFh. If you have Ethernet networking cards, be aware that many of them occupy upper memory. Many Ethernet cards have an upper memory base address of D000h or E000h.

Memory Area Used by Expanded-Memory Manager or Emulator

Your system may contain an expanded-memory manager or an expanded-memory emulator that uses a block of upper memory within the range that is allocated for the Master Window.

To solve this problem, verify which expanded-memory manager or emulator you are using, and prevent it from using upper memory within the Master Window address range. The following examples show how to prevent various expanded-memory managers or emulators from using upper memory within the default Master Window address range of D000h to DFFFh.

Note: ***If you have changed your Master Window address range, use it instead of the default range (D000 to DFFF) in the following examples.***

If you are using EMM386, which is included with MS-DOS, prevent it from using the default Master Window address range by editing the `CONFIG.SYS` file and appending the flag `X=D000-DFFF` to the line `DEVICE=EMM386.EXE`.

If you are using QEMM386 from Quarterdeck, prevent it from using the default Master Window address range by editing the `CONFIG.SYS` file and appending the flag `X=D000-DFFF` to the line `DEVICE=QEMM386.SYS`.

If you are using 386MAX from Qualitas, prevent it from using the default Master Window address range by editing the `386MAX.PRO` file located in the `386MAX` directory, and inserting the line `RAM=D000-DFFF`.

If you are using any other expanded-memory manager, refer to its documentation to learn how to prevent it from using the default Master Window address range of D000h to DFFFh.

Problems with Mouse

If you notice your mouse behaving erratically after you run `VXIINIT`, the problem is most likely an interrupt conflict. As stated earlier, the AT-MXI requires one interrupt line that it cannot share with another device in the system.

To correct an interrupt conflict, first verify that the AT-MXI board interrupt level (set by a jumper) matches the software setting in the Logical Address Configuration Editor of the VXI resource editor program, `VXIEDIT` or `VXITEDIT`. Then determine if any other devices are configured to use the AT-MXI interrupt level. The default AT-MXI interrupt level is 12.

If there is a conflict, move either the conflicting device or the AT-MXI to a different interrupt level. If you decide to change the AT-MXI interrupt level, you must change the interrupt jumper hardware setting, enter the new interrupt level in the Configuration Editor, save the changes, and run `VXIINIT`.

Problems with MXIbus Cable

If you have verified that you do not have a conflict with any of the AT-MXI configuration settings, your problem might be with the MXIbus cables in your system.

The following symptoms can indicate a problem with the MXIbus cable.

- `RESMAN` finds the AT-MXI board at Logical Address 0 but does not detect the VXI/VME chassis.

- `RESMAN` previously worked, but now produces errors.

Perform the following checks to determine the problem with your MXIbus cables.

1. Check the pins on your cable connectors. Sometimes the pins get bent and fail to provide the proper electrical connection.

2. Make sure the each end of the MXIbus cable is plugged into the proper connector. Certain MXIbus cables have a single-point connector on one end and a daisy-chain connector on the other end. With these cables, you must connect the single-point connector to the MXIbus device that is closer in the MXIbus chain to the MXIbus System Controller. The MXIbus System Controller determines who gains control of the MXIbus. By default, the AT-MXI is the MXIbus System Controller (set by the Bus Configuration Editor) and the VME-MXI is not the System Controller (set by an onboard slide switch). If you are having problems, check that the AT-MXI and the VME-MXI are set accordingly.

3. If the cabling problem is not apparent, try another MXIbus cable if you have one available. If this solves the problem, contact National Instruments for repair information.

Appendix C
Customer Communication

For your convenience, this appendix contains forms 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.

National Instruments provides comprehensive technical assistance around the world. In the U.S. and Canada, applications engineers are available Monday through Friday from 8:00 a.m. to 6:00 p.m. (central time). In other countries, contact the nearest branch office. You may fax questions to us at any time.

Corporate Headquarters
(512) 795-8248
Technical support fax: (800) 328-2203
(512) 794-5678

Branch Offices	**Phone Number**	**Fax Number**
Australia	(03) 879 9422	(03) 879 9179
Austria	(0662) 435986	(0662) 437010-19
Belgium	02/757.00.20	02/757.03.11
Denmark	45 76 26 00	45 76 71 11
Finland	(90) 527 2321	(90) 502 2930
France	(1) 48 14 24 00	(1) 48 14 24 14
Germany	089/741 31 30	089/714 60 35
Italy	02/48301892	02/48301915
Japan	(03) 3788-1921	(03) 3788-1923
Netherlands	03480-33466	03480-30673
Norway	32-848400	32-848600
Spain	(91) 640 0085	(91) 640 0533
Sweden	08-730 49 70	08-730 43 70
Switzerland	056/20 51 51	056/27 00 25
U.K.	0635 523545	0635 523154

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 __________MB Display adapter __________

Mouse ________yes ________no Other adapters installed ____________

Hard disk capacity __________MB 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-AT2000 Hardware and Software Configuration Form

Record the settings and revisions of your hardware and software on the line to the right of each item. Complete a new copy of 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 Labels: *NI-VXI Distribution Disks for the AT-MXI and MS-DOS*)
- Application Programming Language ____________________
  (QuickBASIC or C)
- Programming Language Interface Revision ____________________
- AT-MXI Base I/O Address ____________________
- AT-MXI Master Window Base ____________________
- AT-MXI Address Space ____________________
- AT-MXI Board Interrupt Level ____________________
- AT-MXI Master DMA Channel ____________________
- AT-MXI Slave DMA Channel ____________________
- AT-MXI Slave I/O Window Base ____________________
- AT-MXI Slave I/O Window Size ____________________
- AT-MXI Shared RAM Window Base ____________________
- AT-MXI Shared RAM Window Size ____________________
- AT-MXI Shared Memory Pool (Windows) ____________________
- AT-MXI Resource Manager Delay ____________________
- AT-MXI is MXIbus System Controller? ____________________
- AT-MXI is Fair Requester? ____________________
- MXIbus Interrupt Level ____________________
- Slave Mode Timeout ____________________
- Master Mode Timeout ____________________
- System Controller Timeout ____________________

- 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 ______________________________
- AT-MXI Hardware Revision ______________________________

Other Products

- Computer Make and Model ______________________________
- Microprocessor ______________________________
- Clock Frequency (Bus and Microprocessor) ______________________________
- Total Memory in System ______________________________
- Type of Video Board Installed ______________________________
- MS-DOS Version ______________________________
- Programming Language Version ______________________________
- Other Boards in System ______________________________
- Base I/O Address of Other Boards ______________________________
- DMA Channels of Other Boards ______________________________
- Interrupt Level of Other Boards ______________________________
- VMEbus Chassis Make and Model ______________________________
- VMEbus Devices in System ______________________________
- Address of other VMEbus 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-AT2000 and the NI-VXI™ Software for MS-DOS**

Edition Date: **August 1994**

Part Number: **320332B-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
National Instruments Corporation
6504 Bridge Point Parkway, MS 53-02
Austin, TX 78730-5039

Fax to:
Technical Publications
National Instruments Corporation
MS 53-02
(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 KB *short* address space. In VXI, the upper 16 KB of A16 space is allocated for use by VXI devices configuration registers. This 16 KB region is referred to as VXI configuration space.

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

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

ACFAIL* — A VMEbus backplane signal that is asserted when a power failure has occurred (either AC line source or power supply malfunction), or if it is necessary to disable the power supply (such as for a high temperature condition).

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.

asynchronous

Not synchronized; not controlled by time signals.

B

B

bytes

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.

binary

A numbering system with a base of 2.

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.

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.

configuration 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

DACK

DMA Acknowledge

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.

DIP

Dual Inline Package

DMA

Direct Memory Access; a method by which data is transferred between devices and internal memory without intervention of the central processing unit.

DRQ

DMA Request

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
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
GPIB	General Purpose Interface Bus; the industry-standard IEEE 488 bus.

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

IRQ*
Interrupt signal

ISA
Industry Standard Architecture

K

KB
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

MB
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	Module Identification lines
`movs`	Move string command
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

PC AT	Personal Computer Advanced Technology
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.
RM	See *Resource Manager*.

S

s	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. In VXI, a device that fails also clears its PASSed bit in its Status register.

SYSRESET*
A VMEbus signal that is used by a device to indicate a system reset or power-up condition.

T

trigger
Either TTL or ECL lines used for intermodule communication.

TTL
Transistor-Transistor Logic

V

V
volts

VDC
volts direct current

VIC
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. Called *VICTEXT* in text-based applications.

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.

VXI
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.

VXIEDIT 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. Called *VXITEDIT* in text-based applications.

W

Word Serial Protocol The simplest required communication protocol supported by Message-Based devices in the VXIbus system. It utilizes the A16 communication registers to transfer data using a simple polling handshake method.