COMPREHENSIVE SERVICES

We offer competitive repair and calibration services, as well as easily accessible documentation and free downloadable resources.

SELL YOUR SURPLUS

We buy new, used, decommissioned, and surplus parts from every NI series. We work out the best solution to suit your individual needs.

〜〜 Sell For Cash   〜〜 Get Credit   〜〜 Receive a Trade-In Deal

OBSOLETE NI HARDWARE IN STOCK & READY TO SHIP

We stock **New, New Surplus, Refurbished,** and **Reconditioned** NI Hardware.

△W APEX WAVES

Bridging the gap between the manufacturer and your legacy test system.

📞 **1-800-915-6216**

🌐 **www.apexwaves.com**

✉ **sales@apexwaves.com**

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

Request a Quote ✉ CLICK HERE VME-MXI

Getting Started with Your VME-SB2020 and the NI-VXI™ Software for Solaris

MXI
bus

February 1994 Edition

Part Number 320338-01

Limited Warranty

The National Instruments MXIbus boards and accessories 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

LabVIEW® and NI-VXI™ are trademarks of National Instruments Corporation.

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

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.

FCC/DOC Radio Frequency Interference Compliance

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

Federal Communications Commission

This device complies with Part 15 of the Federal Communications Commission (FCC) Rules for a Class A digital device. Operation is subject to the following two conditions:

1. This device may not cause harmful interference in commercial environments.

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

Canadian Department of Communications

This device complies with the limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications (DOC).

Le présent appareil numérique n'émiet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de classe A prescrites dans le réglement sur le brouillage radioélectrique édicté par le ministére des communications du Canada.

Instructions to Users

These regulations are designed to provide reasonable protection against harmful interference from the equipment to radio reception in commercial areas. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

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

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

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

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

• Reorient or relocate the receiver's antenna.

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

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

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

Contents

Contents

Appendix A
Specifications

Appendix B
Troubleshooting

Appendix C
Customer Communication

Glossary

Figures

Tables

About This Manual

This manual contains instructions for installing and configuring the National Instruments VME-SB2020 interface kit and the NI-VXI software for Solaris. The VME-SB2020 kit contains separate software distribution disks for the Solaris 1.x and 2.x platforms. Any differences between the two platforms are described specifically. This manual is meant to be used with the *NI-VXI C Software Reference Manual for VME* (part number 320389-01) and the *NI-VXI Text Utilities Reference Manual* (part number 320321-01).

How to Use the Documentation Set

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

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

Organization of This Manual

Getting Started with Your VME-SB2020 and the NI-VXI Software for Solaris is organized as follows:

*   Chapter 1, *Introduction*, describes the VME-SB2020 interface kit, lists the contents of your kit, and lists optional equipment and software.

*   Chapter 2, *Hardware Configuration and Installation*, contains the instructions to configure and install the VME-SB2020 interface kit.

*   Chapter 3, *NI-VXI Software Installation and Configuration*, lists all the programs and files located on the NI-VXI distribution diskettes and contains instructions for installing and configuring the NI-VXI software for VME. Select the proper distribution disk to install NI-VXI software for either Solaris 1.x or Solaris 2.x.

*   Appendix A, *Specifications*, lists various module specifications of the SB-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 for Solaris.

- 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, and symbols.

Conventions Used in This Manual

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

italic	Italic text denotes emphasis, a cross reference, or an introduction to a key concept.
bold italic	Bold italic text denotes a warning.
`monospace`	Text in this font denotes text or characters that are to be literally input from the keyboard, the proper names of directories, device drivers, programs, 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.
`<>`	Angle brackets enclose the name of a key on the keyboard–for example, `<Enter>`.

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

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

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

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

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

Customer Communication

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

Chapter 1
Introduction

This chapter describes the VME-SB2020 interface kit, lists the contents of your kit, and lists optional equipment and software.

The VME-SB2020 is an interface kit that links any Sun workstation or compatible (hereafter referred to as the Sun) equipped with SBus expansion slots directly to the VMEbus. A Sun equipped with a VME-SB2020 can control the VMEbus directly. The VME-SB2020 makes the Sun function as though it were an embedded CPU plugged directly into the VME backplane, or as though it has internal VME slots for plug-in VME boards. With the VME-SB2020, a Sun can use plug-in VME boards such as those designed for use with Sun-3 and Sun-4 VME-based workstations. The driver software for these boards must, of course, be ported to run on the Sun with the VME-SB2020 kit.

What Your Kit Should Contain

Your kit should contain the following components:

Component	Part Number
SB-MXI Interface Board	181155-01
Standard VME-MXI Interface Module or Enhanced VME-MXI Interface Module with INTX option	180960-01 180960-02
NI-VXI Distribution Disk for the SB-MXI and Solaris 1.x and NI-VXI Distribution Disk for the SB-MXI and Solaris 2.x	422190-58 422925-58
2 m Type M1 MXIbus Cable *	180758-02
Getting Started with Your VME-SB2020 and the NI-VXI Software for Solaris	320338-01
NI-VXI C Software Reference Manual for VME	320389-01
NI-VXI Text Utilities Reference Manual	320321-01

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

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

Optional Equipment

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

Optional Software

If you want to use LabVIEW with your VME-SB2020, you must order the LabVIEW for Sun VXI Development System that corresponds to your Sun system. You can use LabVIEW with either Solaris 1.x or Solaris 2.x.

Software	Part Number

LabVIEW for Sun VXI Development System 1 User:
    LabVIEW for Sun Full Development System,
    LabVIEW for Sun VXI Library, and
    LabVIEW for Windows/Sun VXI Instrument Library
	776684-211

LabVIEW for Sun VXI Development System 5 User:
    LabVIEW for Sun Full Development System,
    LabVIEW for Sun VXI Library, and
    LabVIEW for Windows/Sun VXI Instrument Library
	776684-213

LabVIEW for Sun VXI Development System 10 User:
    LabVIEW for Sun Full Development System,
    LabVIEW for Sun VXI Library, and
    LabVIEW for Windows/Sun VXI Instrument Library
	776684-214

LabVIEW for Sun VXI Development System 25 User:
    LabVIEW for Sun Full Development System,
    LabVIEW for Sun VXI Library, and
    LabVIEW for Windows/Sun VXI Instrument Library
	776684-215

If you want to use LabWindows/CVI with your VME-SB2020, you must order the
LabWindows/CVI for Sun VXI Development System that corresponds to your Sun system.
You can use LabWindows/CVI with either Solaris 1.x or Solaris 2.x.

Software	**Part Number**
LabWindows/CVI for Sun VXI Development System 1 User: LabWindows/CVI for Sun Full Develoment System, LabWindows/CVI for Sun VXI Libraries, LabWindows/CVI for Sun VXI Instrument Library	776824-231
LabWindows/CVI for Sun VXI Development System 5 User: LabWindows/CVI for Sun Full Develoment System, LabWindows/CVI for Sun VXI Libraries, LabWindows/CVI for Sun VXI Instrument Library	776824-233
LabWindows/CVI for Sun VXI Development System 10 User: LabWindows/CVI for Sun Full Develoment System, LabWindows/CVI for Sun VXI Libraries, LabWindows/CVI for Sun VXI Instrument Library	776824-234
LabWindows/CVI for Sun VXI Development System 25 User: LabWindows/CVI for Sun Full Develoment System, LabWindows/CVI for Sun VXI Libraries, LabWindows/CVI for Sun VXI Instrument Library	776824-235

Chapter 2
Hardware Configuration and Installation

This chapter contains the instructions to configure and install the VME-SB2020 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 SB-MXI and VME-MXI.

2. Install the SB-MXI hardware.

3. Configure the VME-MXI hardware.

4. Install the VME-MXI hardware.

5. Connect the MXIbus cable.

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

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

1. Before attempting to configure or install the SB-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 SB-MXI board and VME-MXI module are shipped packaged in antistatic plastic bags to prevent electrostatic damage. Some of the circuitry on the SB-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 SB-MXI and VME-MXI from their bags, be sure to handle them only by their edges. Avoid touching any of the IC components or connectors. Inspect them for loose components or any other sign of damage. Notify National Instruments if either board appears damaged in any way. *Do not* install equipment that appears to be damaged.

Step 2. Install the SB-MXI Hardware

The SB-MXI does not have any jumpers or switches for configuring the hardware. The base address of the SB-MXI is determined by the geographic mechanism provided by the SBus. The SB-MXI uses SBus interrupt level 3, which it can share with other SBus boards.

Before you install the SB-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 SBus slots. Normally, this will only be a problem if the cable connector hoods for the adjacent slots are also oversized. When choosing an SBus slot in which to install the SB-MXI, verify that the MXIbus cable connector will not interfere with cables and connectors in other SBus slots. If necessary, reposition the boards in the system to prevent cabling conflicts. It may also help to install the SB-MXI in one of the end slots so that you will have to contend with the cable connectors of only one other board.

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

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

1. Plug in your Sun workstation before installing the SB-MXI. The plug grounds the system unit and protects it from electrical damage while you are installing boards.

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

2. Remove the cover of the system unit.

3. Select any available SBus slot and remove the sheet-metal protector plate that covers the slot.

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

5. Depending on the type of Sun workstation, you need to select one of the following methods for installing the SB-MXI. The difference is in regard to the mounting plate on the SB-MXI.

   a. For most older Sun workstations, slide the SB-MXI at an angle into the back panel of the system unit while making sure the mounting plate on the SB-MXI hooks into the holes on the back panel of the system unit.

   b. Most newer Sun workstations cannot accept the top part of the mounting plate on the SB-MXI. This piece has tabs on either end and is fastened to the mounting plate by two screws. Remove this piece before attempting to install the SB-MXI.

6. Align the SBus connector plug of the SB-MXI with the SBus socket and gently press the plug into the socket.

7. Check the installation.

8. Replace the cover to the system unit.

Figure 2-1 shows an SB-MXI installed in a SPARCstation 1+.

Art not available in PDF version of document.
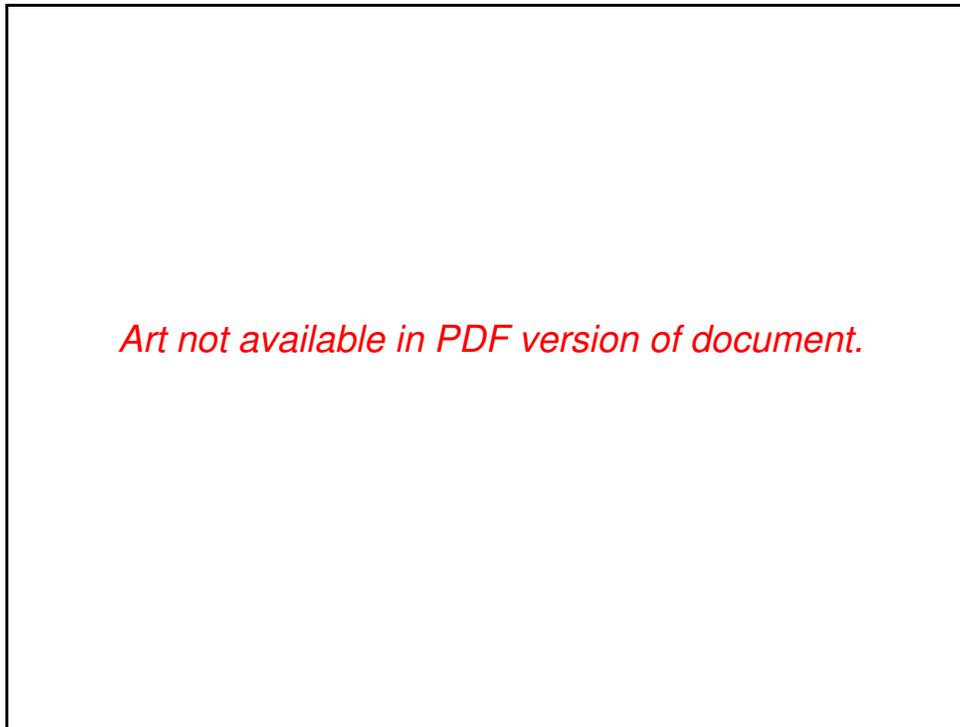
Figure 2-1. SB-MXI Installed in a SPARCstation 1+

Step 3. Configure the VME-MXI Hardware

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

Art not available in PDF version of document.
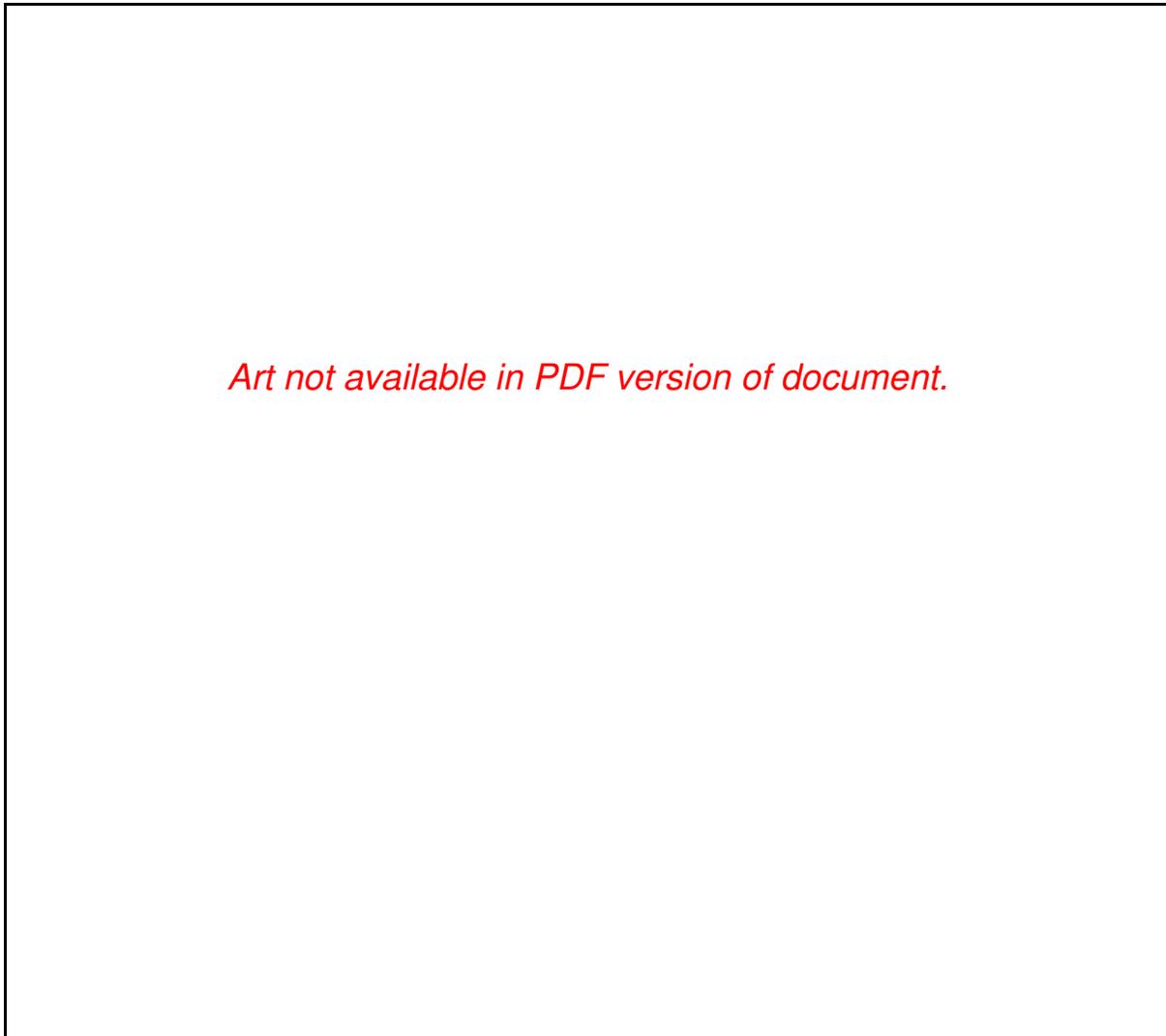
Figure 2-2. VME-MXI Parts Locator Diagram

Figure 2-3 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.
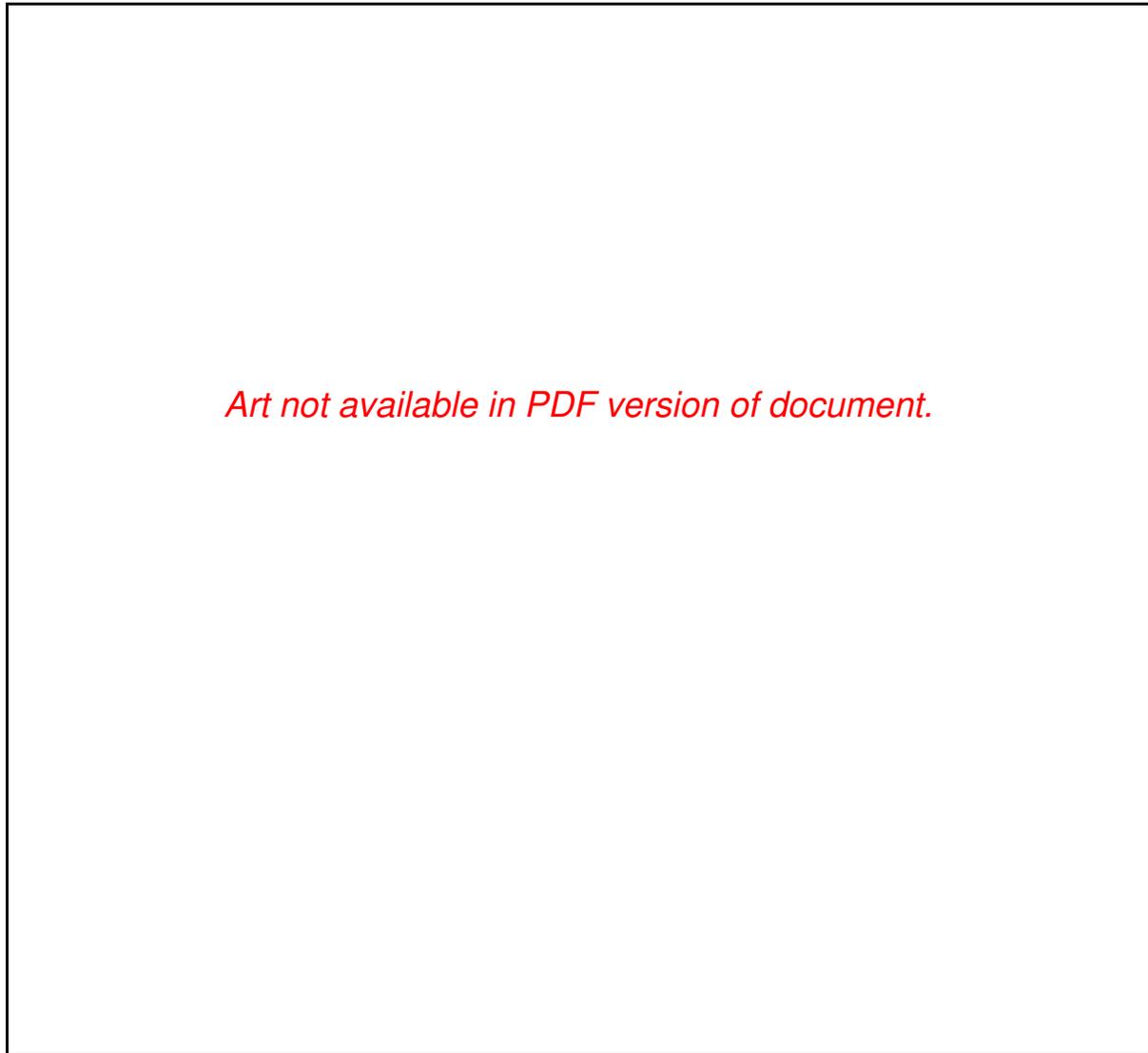
Figure 2-3. VME-MXI with INTX Parts Locator Diagram

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 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 2-4(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 2-4(b).
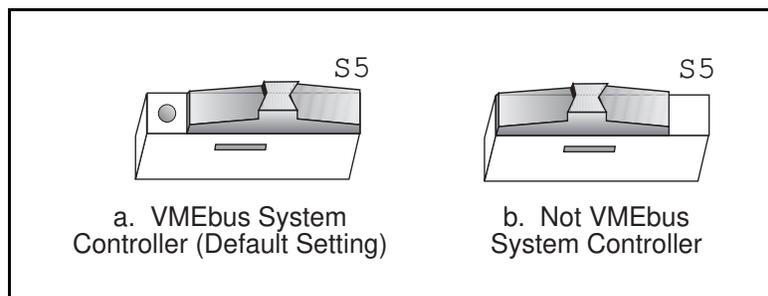
Figure 2-4. VMEbus System Controller Selection

VME Address

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

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

The SB-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 SB-MXI will use A16 address space from C000h through C03Fh. The logical address of the SB-MXI is software configurable, but you should not change this setting when using the SB-MXI 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 you should not change this setting 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 SB-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 Sun workstation with an SB-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 2-5.
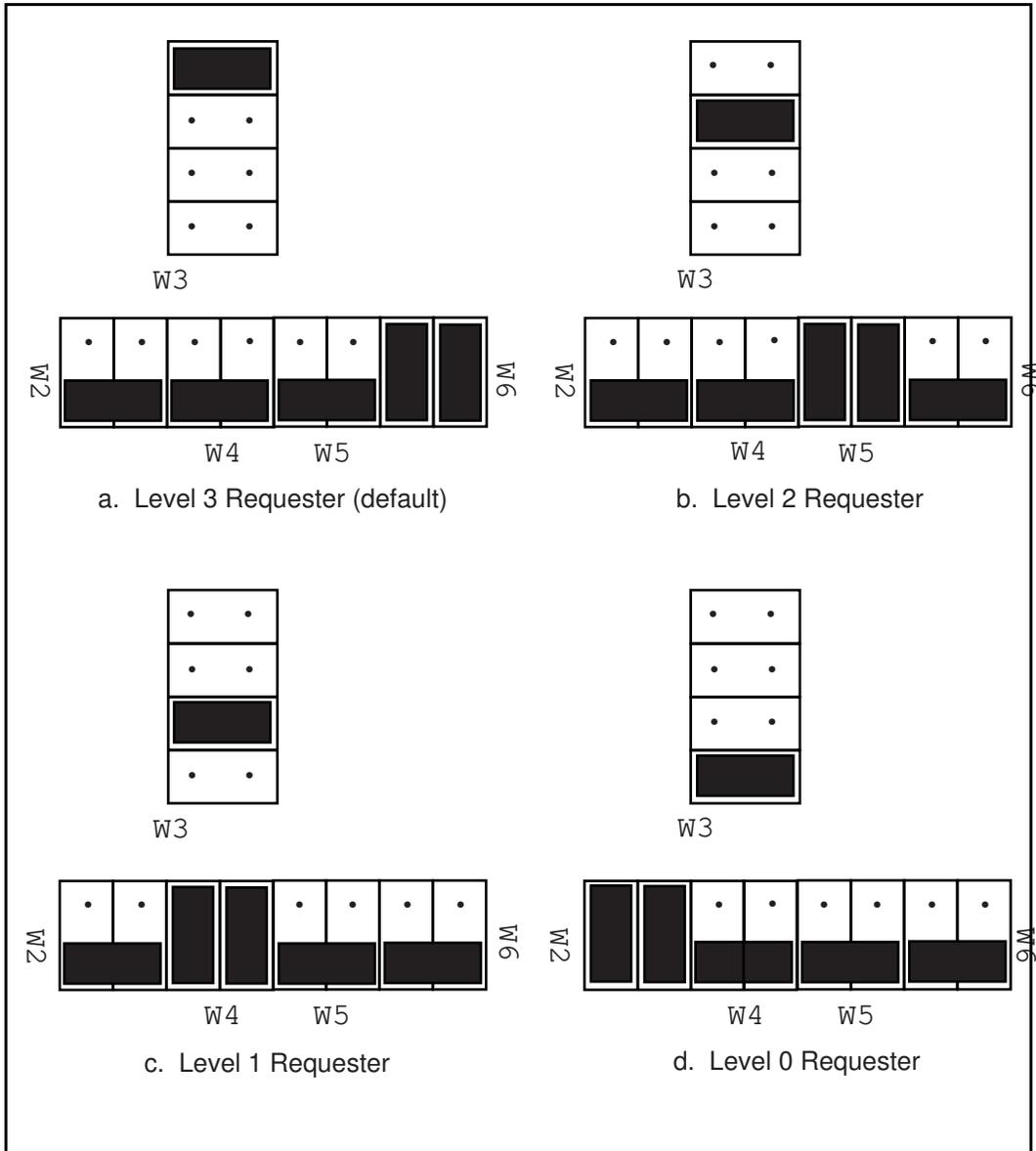
Figure 2-5. VME-MXI VMEbus Requester Jumper Settings

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 2-6.
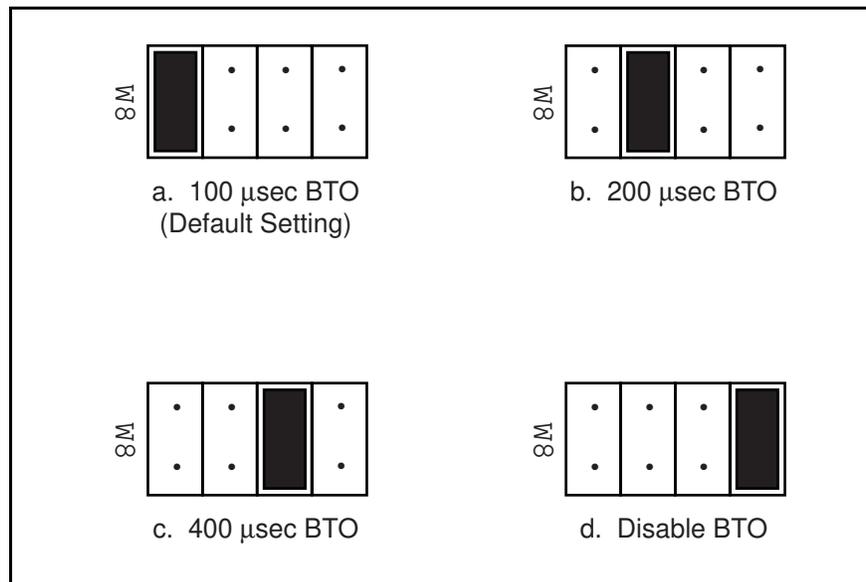
Figure 2-6. VMEbus Timeout Value Selection

Step 4. Install the VME-MXI Hardware

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

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

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

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

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

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

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

5. Check the installation.

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

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

Step 5. Connect the MXIbus Cable

There are two basic types of MXIbus cables. MXIbus cables can have either a single connector on each end or a single connector on one cable end and a double connector on the other end. Your VME-SB2020 kit comes standard with a cable with single connectors on each end.

Nonpolarized Cables

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

Polarized Cables

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

When the MXIbus cable is properly connected, turn on the VME chassis and the Sun workstation. After all devices are powered on and running properly, you can run the the Resource Manager software utility to configure the system.

Chapter 3
NI-VXI Software Installation and Configuration

This chapter lists all the programs and files located on the NI-VXI distribution diskettes and contains instructions for installing and configuring the NI-VXI software for VME. Select the proper distribution disk to install NI-VXI software for either Solaris 1.x or Solaris 2.x. Some parts of this chapter contain material specific to either Solaris 1.x or Solaris 2.x. Be sure you are following the instructions relevant to the Solaris platform on your computer.

The NI-VXI software 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 was actually developed for the VXI-SB2020 kit for VXI systems, 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 contains comprehensive tools for VME systems. Some of the utility programs and driver software routines apply only to VXI systems and are not used in VME systems. Refer to the remainder of this chapter and to the *NI-VXI C Software Reference Manual for VME* to determine which of the 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 following are the main programs and files of the NI-VXI software. Unless otherwise indicated in the descriptions, the NI-VXI software includes these programs and files for both Solaris 1.x and Solaris 2.x. Other files, as noted, are present for only one of the Solaris platforms.

* `README` contains the latest updates and corrections to the manual when appropriate.

* `nivxi.INSTALL` is the shell script for installing the NI-VXI device driver for Solaris 1.x.

* `vxiinit` is the MXIbus initialization program. This program initializes the SB-MXI board for operation. Run this program before using the NI-VXI software and after every time you start or restart your computer.

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

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

- `vxitedit` is the text-based VXI Resource Editor program you use to configure the system. Use the Non-VXI Device Editor in this program to identify details about VME devices installed in your system. You must use this editor to instruct your system about the addresses your VME devices occupy. The Resource Manager can then use this configuration information to automatically *open* the hardware windows so that your Sun 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 Text Utilities Reference Manual*.

- `libnivxi*` contains the NI-VXI interface libraries for the NI-VXI device driver.

Additional Programs and Files

The `/tbl` directory contains the following files.

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

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

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

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

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

- `trigcfg.tbl` contains TTL trigger configuration editor information. This table file is not used in VME systems.

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

- `creg.tbl` contains device-dependent information to be written to the control register by the Resource Manager.

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

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

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

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

- `datasize.h` contains data size specifications.

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

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

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

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

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

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

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

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

The `/bin` directory contains copies of the executable files, as well as the `nivxi.{add, rem, uld, info}` shell scripts for manipulating the NI-VXI device driver for Solaris 2.x.

The `/lib` directory contains the code for the NI-VXI interface libraries.

The `/drv` directory contains copies of `nivxi.*.o` and `nivxi`, the loadable device drivers for Solaris 1.x and Solaris 2.x, respectively.

Installing and Loading the NI-VXI Software for Solaris 1.x

The instructions in this section are specific to users of the Solaris 1.x platform. If your Solaris platform is Solaris 2.x, skip over this material and continue with the section *Installing and Loading the NI-VXI Software for Solaris 2.x.*

Note: ***Upgrading from a previous version of NI-VXI requires no steps other than those listed in the following installation and loading sections. You do not need to remove any files from your current NI-VXI directory, because the installation steps will overwrite any old files. However, we recommend that you back up any files that you have added or modified.***

Installing the NI-VXI Software for Solaris 1.x

1. Log on as super-user (root privileges needed).

2. Use the `cd` command to change to the directory where you want to install the NI-VXI software.

3. Type the following command:

    ```
    bar xvfZ /dev/rfd0c
    ```

Loading the NI-VXI Driver for Solaris 1.x

The NI-VXI driver for Solaris 1.x is a loadable driver. You do *not* need to go through the process of linking the driver with the kernel's object files, rebuilding the kernel, and restarting the system, as you would have to do with a nonloadable driver.

Note: ***The SB-MXI must be installed in your computer before you can load the NI-VXI driver.***

Run `nivxi.INSTALL` in your working directory and follow the instructions in the shell script. Enter the following command:

```
./nivxi.INSTALL
```

You will receive the following prompt:

```
Should the driver be loaded during each reboot? [y/n] (y):
```

The default is `y` for yes. If you do not want the driver to be automatically loaded during a restart, type `n` and then <Enter>.

If the driver is installed correctly, the following message appears:

`nivxi: module loaded; id=#`

where *#* is an identification number that is returned by the operating system.

The following message appears on the console:

`NI-VXI device driver loaded.`
`Copyright (c) 1994 National Instruments Corporation`
`All Rights Reserved.`

If the driver is already loaded, the following message appears on the console:

`Can't load this module:  No such device or address.`

Note: *If you want to load the driver for this session only, use the following script instead of the* `INSTALL` *script:*

   `/dev/nivxi.LOAD`

Unloading the NI-VXI Driver for Solaris 1.x

To unload the driver, you must meet the following requirements:

- You must have super-user privilege.

- The driver must not be in use.

If the driver is in use and you try to unload it, the following message appears:

`Cannot unload the module:  Device busy.`

You can use the `modstat` utility if you want to check the status of the loaded drivers. It displays the module ID of the driver, the name of the device, and additional information about the module. Type the following command to use this utility:

`modstat`

Unload the driver by entering the following command:

`/dev/nivxi.UNLOAD`

The following message appears on the console:

`NI-VXI driver unloaded.`

To prevent the driver from being reloaded automatically at startup, type the following command to remove the file:

`rm /dev/nivxi.AUTOLOAD`

Installing and Loading the NI-VXI Software for Solaris 2.x

The instructions in this section are specific to users of the Solaris 2.x platform. If your Solaris platform is Solaris 1.x, refer instead to the previous section *Installing and Loading the NI-VXI Software for Solaris 1.x*.

Upgrading from NI-VXI Version 1.0 for Solaris 2.x

If you are upgrading from NI-VXI Version 1.0, it is important to perform the following steps before installing the new version.

1. Move any files from your current NI-VXI directory that you wish to keep. The following steps will remove the current NI-VXI directory, so you will need to move any files that you have modified and place them in a temporary directory.

2. Issue the following command to remove the old package:

   `/usr/sbin/pkgrm NIvxi`

3. Modify any environment variables or program paths to point to the new NI-VXI directory. Starting with NI-VXI Version 2.1, the default NI-VXI directory is `/opt/NICsbmxi`. Therefore, you will need to change any references to `/opt/NIvxi`, or whichever alternative directory you had chosen.

4. Continue with the installation as described in the following sections.

Installing the NI-VXI Software for Solaris 2.x

Log on as super-user (root privileges needed).

Use one of the following three commands as applies to your system:

- If you are *not* running volume management, type the following command:

  `/usr/sbin/pkgadd –d /dev/diskette.`

Caution: ***If you are running volume management, you will need to run*** `volcheck` ***to detect the disk in the floppy drive. Due to the manner in which the data is encoded on the disk,*** `volcheck` ***will warn you that the disk is unformatted. Do not select to format the disk. Simply cancel out of the warning.***

- If you *are* running volume management, and you have Version 2.3 or higher, type the following command:

  `/usr/sbin/pkgadd –d /vol/dev/rdiskette0/unlabeled`

- If you *are* running volume management, and you have Version 2.2 or lower, type the following command:

```
/usr/sbin/pkgadd –d /vol/dev/rfd0/unlabeled
```

Note: *During installation, you will be prompted for the base directory in which to install the NI-VXI software. We recommend you select* `/opt`*. This will put the software in* `/opt/NICsbmxi`*.*

Loading the NI-VXI Driver for Solaris 2.x

The NI-VXI driver for Solaris 2.x was added to the driver list automatically during installation. It is loaded the first time you open the driver (for example, running `vxiinit`). Because it is a loadable driver, you do *not* need to go through the process of linking the driver with the kernel's object files, rebuilding the kernel, and restarting the system, as you would have to do with a nonloadable driver.

Note: *The SB-MXI must be installed in your computer before you can load the NI-VXI driver.*

Unloading the NI-VXI Driver for Solaris 2.x

To unload the driver, you must meet the following requirements:

- You must have super-user privilege.

- The driver must not be in use.

If the driver is in use and you try to unload it, the following message appears.

Cannot unload the module: Device busy.

You can use the `nivxi.info` utility if you want to check the status of the loaded NI-VXI driver. It displays the module ID of the driver, the name of the device, and additional information about the module. Type the following command to use this utility:

```
nivxi.info
```

If you want to unload the driver, enter the following command:

```
nivxi.uld
```

To completely remove the NI-VXI driver from the driver list, type the following command:

nivxi.rem

Using the NI-VXI Software

The NI-VXI software expects to be loaded in the `/usr/nivxi` directory if you are using the Solaris 1.x platform. If you are using Solaris 2.x, the driver expects to be loaded in the `/opt/NICsbmxi` directory. If you have installed the software in another directory, you need to set the `NIVXIPATH` environment variable to your directory. For example, if you have installed NI-VXI in `/usr2/nivxi`, type the following command:

```
setenv NIVXIPATH        /usr2/nivxi
```

You must always set the `LD_LIBRARY_PATH` environment variable to your directory. For example, if you have installed NI-VXI in `/usr/nivxi`, type the following command:

```
setenv LD_LIBRARY_PATH /usr/nivxi
```

Place these lines in your `.cshrc` (C shell) or `.profile` (Bourne or Korn shells).

Note: *You must run* `vxiinit` *to configure the SB-MXI each time you start up or restart the computer, or if you have changed the configuration with the* `vxitedit` *program, as described in the configuration sections of this chapter.*

Using LabVIEW or LabWindows/CVI

For information on developing application programs using LabVIEW, refer to the *LabVIEW VXI VI Reference Manual*. To use LabVIEW to program your system, you must have a LabVIEW for Sun VXI Development System. Refer to the *Optional Software* section in Chapter 1, *Introduction*.

For information on developing application programs using LabWindows/CVI, refer to the *NI-VXI C Software Reference Manual for VME*. You can use the functions and the syntax as described in this software reference manual. To use LabWindows/CVI to program your system, you must have a LabWindows/CVI for Sun VXI Development System. Refer to the *Optional Software* section in Chapter 1, *Introduction*.

Configuring the NI-VXI Software

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

Run the `vxitedit` device configuration utility if you need to make any changes to the default software configuration. If you do not use `vxitedit` to make changes, the default characteristics of the software will remain in effect. You can also run `vxitedit` if you just want to examine the software configurations.

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

Type `vxitedit` to start up the device configuration utility.

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

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

Configuration Editor

If you want to view the default hardware interface configuration for your hardware link to VME or make changes to it, select the Configuration Editor from the main menu by entering the option number or by typing in the editor name. The three configuration options available under the Configuration Editor are Logical Address Configuration, Bus Configuration, and Device Configuration. The Device Configuration option is not used. For help on individual editors, type `help` while in the corresponding editor.

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

Logical Address Configuration

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

<p align="center">Table 3-1. Logical Address Configuration Characteristics</p>

Characteristic	Description	Default Value
Logical Address	An 8-bit number that uniquely identifies the SB-MXI. If it is set to 0, it is the Resource Manager. Range = 0 to 254	0
Device Type	Indicates the classification of the SB-MXI. 0 = Memory 1 = Extended 2 = Message-Based 3 = Register-Based	2
Address Space	Indicates the addressing mode(s) of the SB-MXI's operational registers. 0 = A16/A24 1 = A16/A32 2 = Reserved 3 = A16 only	3
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 s to 65535 s	5

<p align="right">(continues)</p>

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

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

Bus Configuration

Table 3-2 describes the characteristics of the MXIbus System Controller option 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 SB-MXI is always designated as the MXIbus System Controller when used in this kit and should *not* be changed. 0 = Not MXIbus System Controller 1 = MXIbus System Controller	
		1

VME (Non-VXI) Device Configuration

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

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

You also need to enter device names, manufacturer names, model names, and other identification information for your VME devices. The `Frame` field should contain the logical address of the VME-MXI in the chassis. The default logical address is 1. A16 VME devices should be configured to occupy memory below C000h in A16 space. Be sure to configure the `InSystem` parameter to instruct the Resource Manager that this particular VME device is indeed installed in your system.

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

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

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

An easy way to create a new entry is to first duplicate a similar entry by typing `duplicate` and giving the index of the existing entry. After you duplicate an entry, modify its pseudo-logical address and the other address and interrupt requirements. Optionally, you can type `add` and enter all the values for a new entry.

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

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

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

Exiting vxitedit and Reinitializing the Hardware

To exit `vxitedit`, type `exit` when the main menu is displayed. If you have changed any information, the program prompts you to save your changes before exiting the menu. Saving the configuration information updates the configuration files `vxila.cfg` and `vxibus.cfg`. Run `vxiinit` to reinitialize the hardware according to the new settings.

If you changed any of the software configuration settings from the default settings, record the new settings on the *VME-SB2020 Hardware and Software Configuration Form* in Appendix C.

Developing Your Application Program

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

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

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

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

Refer to the `/example` directory for further development information.

Appendix A
Specifications

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

SB-MXI

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

Capability Codes

MXIbus

Capability Code	Description
MA32	Master Mode A32, A24 and A16 addressing
MBLT	Master Mode block transfers
MD32	Master Mode D32, D16, and D16 data sizes
SC	Optional MXIbus System Controller
LOCK	Can lock the MXIbus for indivisible transfers
TERM	Can terminate the MXIbus

SBus

Capability Code	Description
NBSD32	Slave D32, D16, and D08 data sizes
INT	Can interrupt the SBus

Electrical

Source	Typical	Direct Current (max)
+5 VDC	2.5 A	3.5 A

Environmental

Component Temperature	0° to 70° C (32° to 158° F) operating -55° to 150° C (-67° to 302° F) storage
Emissions	FCC Class A
Relative Humidity	0% to 95% noncondensing; operating 0% to 100% noncondensing; storage
Safety	Not applicable
Shock and Vibration	Not applicable

Physical

Board Size	Standard single-wide SBus board 5.78 in. by 3.3 in. (146.7 mm by 83.82 mm)
Connectors	Single fully implemented MXIbus connector
Slot Requirements	Single SBus slot

Reliability

MTBF	Contact Factory

Requirements

Memory space required	32 MB

Timing

Master Mode

Transfer Type	Transfer Rate
Write	730 ns
Read	730 ns
Block Write	490 ns
Block Read	330 ns

Other

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

VME-MXI

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

Capability Codes

VMEbus

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

MXIbus

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

Electrical

Source	Typical	Maximum
+5 VDC	5.25 A	7.0 A

Environmental

Component temperature   0° to 70° C operating;
                               -40° to 85° C storage

Relative humidity   10% to 90% noncondensing operating;
                               0% to 95% noncondensing storage

Emissions   FCC Class A

Safety   Not applicable

Shock and Vibration   Not applicable

Physical

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

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

Slot Requirements   Single slot

VXI Keying Class   Class 1 TTL

Fully compatible with VMEbus specification

Reliability

MTBF   Contact Factory

Requirements

A16 Space   64 bytes

Timing

Master Mode

Transfer Type	Transfer Rate
Write	675.5 ns
Read	625.5 ns
Block Write	320 ns
Block Read	279 ns

Slave Mode

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

Other

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

Appendix B
Troubleshooting

This appendix addresses certain problems you may encounter when using the NI-VXI bus interface software for Solaris. If you are unable to resolve your problem, please fill out the appropriate forms in Appendix C, *Customer Communication*, and contact National Instruments.

Problems when Using vxiinit

vxiinit Cannot Find .cfg 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`, `mfnameid.tbl`, and `model.tbl`. There are three possible causes for this error message.

- Ensure that you are using the proper `vxiinit` program for your system. You cannot use the `vxiinit` program distributed with another National Instruments controller to initialize the SB-MXI.

- The file(s) may have been deleted accidentally. If the file does not exist in your `tbl` directory—`nivxi/tbl` for Solaris 1.x or `NICsbmxi/tbl` for Solaris 2.x— after you installed your software, you can find it on your original NI-VXI distribution media and restore it manually.

- The `NIVXIPATH` environment variable may be pointing to the wrong directory. The path should be set in the `.cshrc` or `.profile` files, depending on the shell type you are using (Refer to the *Using the NI-VXI Software* section of Chapter 3). Set the correct path for the `NIVXIPATH` variable, for example, `/usr/nivxi` for Solaris 1.x, or `/opt/NICsbmxi` for Solaris 2.x.

vxiinit Cannot Find Device Driver

If you receive the message **VXI device driver not found**, `vxiinit` has been unable to communicate with the driver. This can be due to the driver either not being in memory or it has become corrupted.

First, verify that the driver is in memory. To do this, use the appropriate command associated with your Solaris platform:

- For Solaris 1.x, type `modstat.`

- For Solaris 2.x, type `nivxi.info.`

If the driver is *not* in memory and you are using Solaris 1.x, use the command `nivxi.LOAD` to load the driver. If you are using Solaris 2.x, or if `nivxi.LOAD` did not load the driver, there is a problem with the system configuration. Check your installation and check with your system administrator.

If the driver *is* loaded in memory, the version may have become corrupted. Proceed by unloading and then reloading the driver as described in Chapter 3.

vxiinit Cannot Find libnivxi.so

When using `vxiinit`, you may receive the following error message:

ld.so.l: vxiinit: fatal: libnivxi.so: can't open file: errno = 2

This message is caused by the driver being unable to find the shared object file, `libnivxi.so`. Check to make sure the environment variable `LD_LIBRARY_PATH` is pointing to the directory containing the NI-VXI software. Refer to the *Using the NI-VXI Software* section of Chapter 3 for more information.

Note: *This error message can also occur when using* `resman` *or* `victext`*.*

Problems when Using resman

If you receive errors or warnings from `resman`, the problem could be in a number of areas. The first thing to remember is that `resman` is a Resource Manager for VXI systems. Because you are using a VME system, if you receive a warning that `resman` cannot find a Slot 0 device, or that devices are not responding to MODID lines, you can ignore these warnings. These items are relevant to VXI systems only.

If you receive warnings other than those just discussed, you may have resource conflicts between the SB-MXI and the VME chassis in your system. First make sure that you have used the non-VXI Device Editor in `vxitedit` to inform `resman` of the memory requirements of all the non-VXI devices in your system.

Another cause can be due to system conflicts with the SB-MXI configuration. If you have verified that you do not have a conflict with any of the SB-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 SB-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 SB-MXI is the MXIbus System Controller (set by the Bus Configuration Editor) and the VME-MXI is *not* the MXIbus System Controller (set by an onboard slide switch). If you are having problems, check that the SB-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
(800) 433-3488 (toll-free U.S. and Canada)
Technical Support fax: (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 65 33 00	1 48 65 19 07
Germany	089 7 14 50 93	089 7 14 60 35
Italy	02 48301892	02 48301915
Japan	03 3788 1921	03 3788 1923
Netherlands	01720 45761	01720 42140
Norway	03 846866	03 846860
Spain	91 640 0085	91 640 0533
Sweden	08 730 49 70	08 730 43 70
Switzerland	056 27 00 20	056 27 00 25
U.K.	0635 523545	0635 523154
	or 0800 289877 (in U.K. only)	

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-SB2020 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 Label: *NI-VXI Distribution Disk for the SB-MXI and Solaris*, for 1.x or 2.x)

- Programming Language Interface Revision   _____

- SB-MXI Logical Address   _____

- SB-MXI Device Type   _____

- SB-MXI Address Space   _____

- SB-MXI Resource Manager Delay   _____

- SB-MXI Servant Area Size   _____

- SB-MXI Protocol Register   _____

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

- Number of SB-MXI Handlers   _____

- Number of SB-MXI Interrupters   _____

- SB-MXI is MXIbus System Controller?   _____

- VME-MXI Installed in Slot 1?   _____

- VME-MXI is VMEbus System Controller?   _____

- VME-MXI Logical Address   _____

- VME-MXI VMEbus Request Level   _____

- VMEbus BTO Timeout Value   _____

- VME-MXI Hardware Revision   _____

- SB-MXI Hardware Revision   _____

Other Products

- Computer Make and Model _____

- Microprocessor _____

- Clock Frequency
  (Bus and Microprocessor) _____

- Type of Video Board Installed _____

- Amount of RAM _____

- Hard Disk Capacity and Brand _____

- Solaris Platform and Version _____

- Programming Language Version _____

- Other Boards in System _____

- VMEbus Mainframe Make and Model _____

- Other VXI/VMEbus Devices in System _____

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-SB2020 and the NI-VXI™ Software for Solaris**

Edition Date: **February 1994**

Part Number: **320338-01**

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

_____

_____

_____

_____

_____

_____

_____

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

_____

_____

_____

_____

_____

_____

_____

_____

Thank you for your help.

Name   _____

Title   _____

Company   _____

Address   _____

_____

Phone   (_____) _____

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

Glossary

_____

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

Symbols

°   degrees

%   percent

±   plus or minus

A

A   amperes

A16 space   VXIbus address space equivalent to the VME 64 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.

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.
ANSI	American National Standards Institute

B

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

C

C	Celsius
CLK10	A 10 MHz, ± 100 ppm, individually buffered (to each module slot), differential ECL system clock that is sourced from Slot 0 and distributed to Slots 1 through 12 on P2. It is distributed to each slot as a single-source, single-destination signal with a matched delay of under 8 ns.
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

IACK | Interrupt Acknowledge

IC | Integrated Circuit

IEEE | Institute of Electrical and Electronics Engineers

in. | inches

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

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

interrupt handler	A VMEbus functional module that detects interrupt requests generated by Interrupters and responds to those requests by requesting status and identify information.
interrupt level	The relative priority at which a device can interrupt.
INTX	Interrupt Timing and Extension

K

KB	1,024; kilobytes of memory

L

LabVIEW	Laboratory Virtual Instrument Engineering Workbench
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.
modstat	A utility that checks the status of the loaded drivers. It displays the module ID of the driver, the name of the device, and additional information about the module.
MTBF	Mean Time Between Failure

MXIbus	Multisystem eXtension Interface Bus; a high-performance communication link that interconnects devices using round, flexible cables.
MXIbus System Controller	A functional module that has arbiter, daisy-chain driver, and MXIbus cycle timeout responsibility. Always the first device in the MXIbus daisy-chain.

N

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

P

propagation	The transmission of signal through a computer system.

R

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

S

s	seconds
Servant	A device controlled by a Commander; there are Message-Based and Register-Based Servants.
shell	An interactive program for commands.
shell script	The commands used to perform batch processing.

slave	A functional part of a MXI/VME/VXIbus device that detects data transfer cycles initiated by a VMEbus master and responds to the transfers when the address specifies one of the device's registers.
slave-mode operation	A device is in slave mode it if is responding to a bus cycle.
statically configured device	A device whose logical address cannot be set through software; that is, it is not dynamically configurable.
SYSFAIL	A VMEbus signal that is used by a device to indicate an internal failure. A failed device asserts this line.

T

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

V

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

`vxitedit`	Text-based VXI Resource Editor program, a part of the NI-VXI bus interface software package. Used to configure the system, edit the manufacturer name and ID numbers, edit the model names of VXI and non-VXI devices in the system, as well as the system interrupt configuration information, and display the system configuration information generated by the Resource Manager.

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.