Contents

Certification

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology (formerly National Bureau of Standards), to the extent allowed by that organization's calibration facility, and to the calibration facilities of other International Standards Organization members.

Warranty

This Hewlett-Packard product is warranted against defects in materials and workmanship for a period of three years from date of shipment. Duration and conditions of warranty for this product may be superseded when the product is integrated into (becomes a part of) other HP products. During the warranty period, Hewlett-Packard Company will, at its option, either repair or replace products which prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by Hewlett-Packard (HP). Buyer shall prepay shipping charges to HP and HP shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to HP from another country

HP warrants that its software and firmware designated by HP for use with a product will execute its programming instructions when properly installed on that product. HP does not warrant that the operation of the product, or software, or firmware will be uninterrupted or error free.

Limitation Of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied products or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

The design and implementation of any circuit on this product is the sole responsibility of the Buyer. HP does not warrant the Buyer's circuitry or malfunctions of HP products that result from the Buyer's circuitry. In addition, HP does not warrant any damage that occurs as a result of the Buyer's circuit or any defects that result from Buyer-supplied products.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HP SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Exclusive Remedies

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. HP SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

Notice

The information contained in this document is subject to change without notice. HEWLETT-PACKARD (HP) MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. HP shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material. This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Hewlett-Packard Company. HP assumes no responsibility for the use or reliability of its software on equipment that is not furnished by HP.

U.S. Government Restricted Rights

The Software and Documentation have been developed entirely at private expense. They are delivered and licensed as "commercial computer software" as defined in DFARS 252.227- 7013 (Oct 1988), DFARS 252.211-7015 (May 1991) or DFARS 252.227-7014 (Jun 1995), as a "commercial item" as defined in FAR 2.101(a), or as "Restricted computer software" as defined in FAR 52.227-19 (Jun 1987)(or any equivalent agency regulation or contract clause), whichever is applicable. You have only those rights provided for such Software and Documentation by the applicable FAR or DFARS clause or the HP standard software agreement for the product involved

HEWLETT® PACKARD

HP E623x VXI Pentium® Controller
Hardware Installation and Configuration Guide
Edition 3
Copyright © 1998 Hewlett-Packard Company. All Rights Reserved.

Trademark and Copyright Notification

Intel and Pentium are U.S. registered trademarks of Intel Corporation.

Microsoft, Windows 95, and Windows NT are U.S. registered trademarks of Microsoft Corporation.

Phoenix and PicoBIOS are registered trademarks of Phoenix Technologies, Ltd. Portions of this manual are copyright 1995 by Phoenix Technologies, Ltd.

Universe is a trademark of Newbridge Microsystems.

Documentation History

All Editions and Updates of this manual and their creation date are listed below. The first Edition of the manual is Edition 1. The Edition number increments by 1 whenever the manual is revised. Updates, which are issued between Editions, contain replacement pages to correct or add additional information to the current Edition of the manual. Whenever a new Edition is created, it will contain all of the Update information for the previous Edition. Each new Edition or Update also includes a revised copy of this documentation history page.

Edition 1 . . . . October 1996

Edition 2 . . . . April 1997

Edition 3 . . . . February 1998

Safety Symbols

Instruction manual symbol affixed to product. Indicates that the user must refer to the manual for specific WARNING or CAUTION information to avoid personal injury or damage to the product.

Indicates the field wiring terminal that must be connected to earth ground before operating the equipment—protects against electrical shock in case of fault.

or Frame or chassis ground terminal—typically connects to the equipment's metal frame.

Alternating current (AC)

Direct current (DC).

Indicates hazardous voltages.

WARNING Calls attention to a procedure, practice, or condition that could cause bodily injury or death.

CAUTION Calls attention to a procedure, practice, or condition that could possibly cause damage to equipment or permanent loss of data.

WARNINGS

The following general safety precautions must be observed during all phases of operation, service, and repair of this product. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the product. Hewlett-Packard Company assumes no liability for the customer's failure to comply with these requirements.

Ground the equipment: For Safety Class 1 equipment (equipment having a protective earth terminal), an uninterruptible safety earth ground must be provided from the mains power source to the product input wiring terminals or supplied power cable.

DO NOT operate the product in an explosive atmosphere or in the presence of flammable gases or fumes.

For continued protection against fire, replace the line fuse(s) only with fuse(s) of the same voltage and current rating and type. DO NOT use repaired fuses or short-circuited fuse holders.

Keep away from live circuits: Operating personnel must not remove equipment covers or shields. Procedures involving the removal of covers or shields are for use by service-trained personnel only. Under certain conditions, dangerous voltages may exist even with the equipment switched off. To avoid dangerous electrical shock, DO NOT perform procedures involving cover or shield removal unless you are qualified to do so.

DO NOT operate damaged equipment: Whenever it is possible that the safety protection features built into this product have been impaired, either through physical damage, excessive moisture, or any other reason, REMOVE POWER and do not use the product until safe operation can be verified by service-trained personnel. If necessary, return the product to a Hewlett-Packard Sales and Service Office for service and repair to ensure that safety features are maintained.

DO NOT service or adjust alone: Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

DO NOT substitute parts or modify equipment: Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification to the product. Return the product to a Hewlett-Packard Sales and Service Office for service and repair to ensure that safety features are maintained.

Declaration of Conformity
according to ISO/IEC Guide 22 and EN 45014

Manufacturer's Name: Hewlett-Packard Company
Loveland Manufacturing Center

Manufacturer's Address: 815 14th Street S.W.
Loveland, Colorado 80537

declares, that the product:

Product Name: HP VXI Pentium® Controller

Model Number: HP E6232A and HP6233A

Product Options: All

conforms to the following Product Specifications:

Safety: IEC 1010-1 (1990) Incl. Amend 1 (1992)/EN61010-1 (1993)
CSA C22.2 #1010.1 (1992)
UL 3111-1 (1994)

EMC: CISPR 11:1990/EN55011 (1991): Group1 Class A
IEC 801-2:1991/EN50082-1 (1992): 4kVCD, 8kVAD
IEC 801-3:1984/EN50082-1 (1992): 3 V/m
IEC 801-4:1988/EN50082-1 (1992): 1kV Power Line
.5kV Signal Lines

Supplementary Information: The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC (inclusive 93/68/EEC) and carries the "CE" mark accordingly.

Tested in a typical configuration in an HP C-Size VXI mainframe.

Jim White

April, 1997 **Jim White, QA Manager**

European contact: Your local Hewlett-Packard Sales and Service Office or Hewlett-Packard GmbH, Department HQ-TRE, Herrenberger Straße 130, D-71034 Böblingen, Germany (FAX +49-7031-14-3143)

Notes:

Chapter 1
Introduction

About this Manual

Welcome to the *HP E623x VXI Pentium® Controller Hardware Installation and Configuration Guide*. This guide explains how to install the VXI Pentium Controller hardware, including any peripherals you wish to connect to the controller. This guide then explains how to power up the the controller and configure the controller's interfaces within the software. Finally, this guide explains how to further customize the controller's configuration, if the default, factory-set configuration is not sufficient for your controller's specific hardware setup.

This first chapter provides an overview of the controller. In addition, this guide contains the following chapters:

- **Chapter 2 - Installing and Configuring the Hardware** explains how to set up the controller hardware, including connecting external peripherals, power up the controller, and configure the controller's interfaces within the software. This chapter then explains how to further customize the controller, if needed.

- **Chapter 3 - Using the BIOS Setup Utility** explains how to set hardware configuration values in the Phoenix(R) PicoBIOS(R) Setup utility for the controller, if needed.

- **Chapter 4 - Using the Start-Up Resource Manager (SURM)** explains how you can further customize the controller via the SURM, if needed.

- **Chapter 5 - Using the VXI Device Configurator** explains how you can further customize the controller via the VXI Device Configurator, if needed.

- **Chapter 6 - Troubleshooting** helps you resolve any error messages or other common problems you may encounter while installing or configuring the controller.

This guide also contains the following appendices:

- **Appendix A - LEDs on the Front Panel** explains the meaning of each LED indicator on the front panel of the controller.

- **Appendix B - I/O, Memory, and IRQ Details** provides more detailed technical descriptions of the VXI Pentium Controller's VGA video and SCSI controllers (including their BIOS extensions), Ethernet controller, memory map, and IRQs (interrupt request lines).

- **Appendix C - Specifications** lists the environmental, electrical, EMC, and safety specifications for the controller. This appendix also contains the Declaration of Conformity for the controller.

- **Appendix D - Freeing Up IRQ3 for a Second HP-22 Card** explains how to free up this IRQ for a second HP-22 GPIB card installed in the HP E623x VXI Pentium Controller running Windows 95.

- **Appendix E - The DEVICES File** explains the DEVICES file record entries and how you can customize the controller by manually editing the DEVICES file directly to create records, rather than by using the VXI Device Configurator.

- **Appendix F - Re-installing the Software** explains the process you should follow if you ever need to re-install the operating system and/or software on the controller.

- **Appendix G - Installing BIOS Software** explains how to reflash the system BIOS in the HP VXI Pentium Controller from the controller's floppy disk drive.

Overview

The HP E623x VXI Pentium Controller is an Intel(R) Pentium-based, dual-slot C-size computer which is PC-compatible and is designed to interface to the VXIbus and the EXMbus. The controller is shipped pre-loaded with either the Microsoft(R) Windows 95(R) or Windows NT(R) version 4.0 operating system, as well as the for Windows software.

Figure 1-1. The HP E623x VXI Pentium Controller

Major Components

The VXI Pentium Controller includes three major components:

- An I/O base board, which contains integrated PCI peripherals for graphics, SCSI, Enhanced IDE, and Ethernet. The I/O base board also supports an RS-232 serial port, a parallel port, a keyboard, and a mouse.

- The CPU submodule, which contains the Pentium computer chip that supports Pentium speeds of 133 to 166 MHz, a secondary cache of 256 KB, and DRAM capacity of 16 to 128 MB. The CPU submodule connects to the I/O base board.

- An Enhanced IDE disk drive module, which supports 2.5-inch hard disk drive densities of 1 GB and larger.

In addition to the integrated peripherals, the VXI Pentium Controller supports two EXMbus slots and a 3.5-inch disk drive. The controller also supports a USB master host port.

Front Panel

As shown below, the front panel of the VXI Pentium Controller spans two VXI slots.

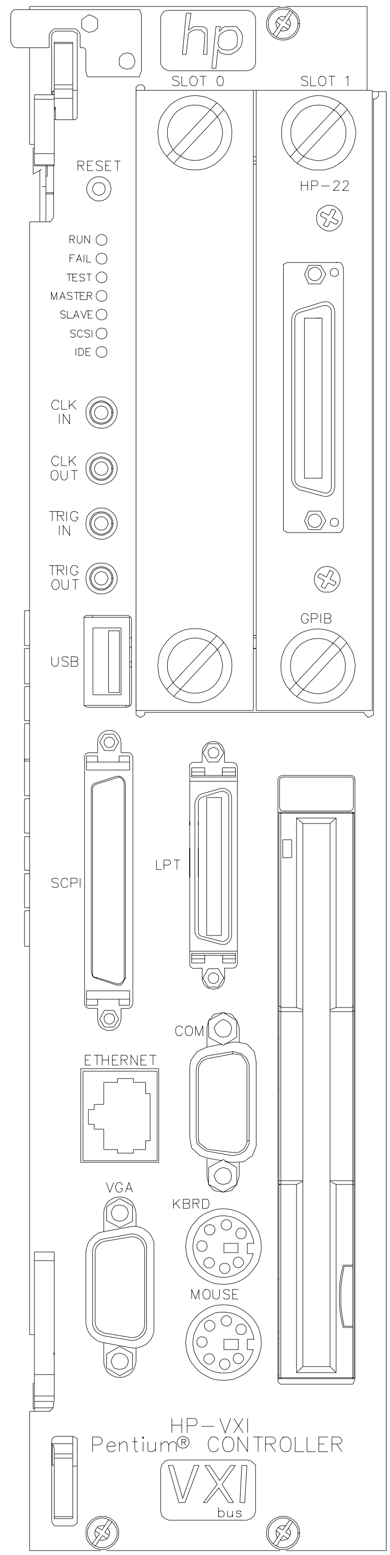

Figure 1-2. The Front Panel of the VXI Pentium Controller

The front panel of the controller contains the following:

- Recessed pushbutton hardware reset switch.
- 7 LED indicators. (For more information about the meaning of the LEDs, see Appendix A, “LEDs on the Front Panel.”)
- 4 SMB connectors, including: CLK10 IN, CLK10 OUT, TRIGGER IN, and TRIGGER OUT.
- 2 EXMbus slots. (Note that an HP-22 GPIB card is pre-installed in Slot 1.)
- 3.5-inch floppy disk drive.
- 9 ports, as listed in the following table.

Table 1-1. Ports on the Front Panel

Label	Port Type	Connector
COM	RS-232 Serial	DB9 pin-type
LPT	IEEE 1284 Parallel	Mini-D36
VGA	Video	VGA 15-pin, socket-type D-shell
SCSI	SCSI	Mini-D, 50-pin SCSI II
KBRD	Keyboard	PS/2-style, 6-pin circular mini-DIN
MOUSE	Mouse	PS/2-style, 6-pin circular mini-DIN
ETHERNET	LAN	10-base-T RJ45
USB	USB Master Host	USB
GPIB	HP-22 GPIB	GPIB

Hardware Feature Set Summary

In summary, the feature set of the VXI Pentium Controller hardware is as follows:

- Intel Pentium CPU.
- Secondary cache of 256 KB.
- Four SODIMM sockets supporting 16 to 128 MB of dual-ported, DRAM memory.

Note The VXI Pentium Controller comes with two open SODIMM sockets, whether using 16 MB or 32 MB DRAM. It is strongly recommended that a trained HP Service Technician makes any memory upgrade, as SODIMMs are more flexible than common SIMMs and are also more costly if damaged.

If you wish to purchase DRAM to have an HP Service Technician upgrade the controller's memory, note that the memory controller supports 60 ns or 70 ns Fast Page mode or EDO DRAM. The memory is interleaved, so DRAM modules must be populated with identical pairs of DRAMs. Parity checking is *not* enabled. Also note that 128 MB requires 64 Mb DRAM technology.

- Intel T2 PCI-based chipset support including interrupt controller, time of day clock, and so forth.
- PCI-based Enhanced IDE controller.
- Integrated 2.5-inch IDE disk drive module.
- Two EXMbus slots accessible from the front panel. (Slot 1 has an HP-22 GPIB card pre-installed.)
- 1.44 MB, 3.5-inch disk drive accessible from the front panel.
- RS-232 serial port with DB9 pin-type connector.
- LPT-compatible parallel port (IEEE 1284 compatible) with an IEEE-C receptacle (mini-D36 connector).
- PCI-based VGA video graphics support with 2 MB display memory and standard 15-pin, socket-type D-shell connector.
- PCI-based SCSI controller with a mini-D, 50-pin SCSI II connector.
- PCI-based Ethernet interface with a 10-base-T RJ45 connector.
- Keyboard port with PS/2-style, 6-pin circular mini-DIN connector.
- Mouse port with PS/2-style, 6-pin circular mini-DIN connector.
- USB Master Host port.
- Watchdog timer.
- Speaker.
- Phoenix-based, system BIOS with Flash update capability from 3.5-inch disk. This update capability may be either forced or disabled (the default) via a jumper.

- Full 32-bit VXIbus interface with P1 and P2 connectors. Note that the VXIbus presents the load of a two-slot module. The VXIbus interface includes the following features:
  - Master control in A16/A24/A32 address spaces for D08, D16, and D32 data widths.
  - D32/D16/D08 slave control in A32/A24 address space for dual ported DRAM.
  - D16/D08 slave control in A16 address space for dual-ported VXI configuration and message-passing registers.
  - VMEbus arbiter, SYSCLOCK driver, and bus timeout.
  - Generate and respond to all 7 VXIbus interrupts.
  - Enhanced TTL and ECL triggers.
  - MODID generation.
  - Slot 0/Slot 1 autodetection (no jumpers required).

For more information on the VGA video, SCSI, and Ethernet controllers, as well as the memory map and IRQs for the VXI Pentium Controller, see Appendix B, "I/O, Memory, and IRQ Details."

Where to Go Next

Now that you have a better understanding of the VXI Pentium Controller, you are ready to begin installing the controller hardware. Please go on to the next chapter, "Installing and Configuring the Hardware," for instructions on how to set up the hardware.

Chapter 2
Installing and Configuring the Hardware

About this Chapter

This chapter contains step-by-step instructions to help you set up the HP E623x VXI Pentium Controller hardware. To install and configure the controller hardware, you need to complete the following steps.

1. First, before you begin the actual installation:

- Verify the contents of the controller's product package to ensure you received all the items for your order.
- Check the location where the controller will be installed to ensure the environmental conditions meet the controller's specifications and to ensure future accessibility to the controller.
- Gather the appropriate equipment so it is ready to install.

2. Install the controller in the VXI card cage.
3. Install any other interface card(s) in the controller (other than the HP-22 GPIB card pre-installed in Slot 1).
4. Connect the peripherals to the controller.
5. Power up the controller.
6. Configure the controller's interfaces within the for Windows software via the **`I/O Config`** utility.
7. Further customize the controller, if needed.

The following sections explain how to complete these installation and configuration steps.

Before You Begin

This section explains what steps you should complete before you begin the actual installation of the VXI Pentium Controller hardware.

Verify the Product Package

Your VXI Pentium Controller product is shipped with the following items. Once you have unpacked the package, ensure that you received all the following items with your product order.

- The HP E623x VXI Pentium Controller.

- Note that the controllers also has one HP-22 GPIB interface card pre-installed in Slot 1.

- The following installation software. Note that the Windows operating system software, the for Windows software, and the controller drivers and utilities software are all pre-loaded on the controller's hard disk. Therefore, the following software CD-ROMs are included in your product package merely as a contingency, in case you ever need to re-install the software for any reason.

  -- Installation CD-ROM for Microsoft Windows 95 or Windows NT, as appropriate.

  -- Installation CD-ROM for the for Windows. Note that the for Windows software includes the HP VISA (Virtual Instrument Software Architecture) library and the HP Standard Instrument Control Library (HP SICL). This CD-ROM also contains the controller drivers and utilities software.

- The following manuals.

  -- This manual (the *HP E623x VXI Pentium(R) Controller Hardware Installation and Configuration Guide*).

  -- The installation/user's manual for Microsoft Windows 95 or Windows NT, as appropriate.

  -- The manual set for the for Windows, including:

    - HP I/O Libraries Installation and Configuration Guide for Windows
    - HP VISA User's Guide
    - HP VISA Quick Reference Guide for C Programmers
    - HP SICL User's Guide for Windows
    - HP SICL Reference Manual
    - HP SICL Quick Reference Guide for C Programmers
    - HP SICL Quick Reference Guide for Visual BASIC Programmers
    - RS-232 Cables Addendum

Note If you also ordered a keyboard, mouse, monitor, other interface card(s), and/or a CD-ROM drive to use with the VXI Pentium Controller, these other products (including their documentation and any software, as appropriate) will be shipped separately to you.

Check the Installation Location

Next, ensure that the environmental conditions where you will install the VXI Pentium Controller will consistently meet the conditions listed in Appendix C, “Specifications.”

Also, you may want to choose an installation location where you can easily access the controller for future updates or changes to the hardware. For example, upgrading the DRAM memory or setting any jumpers in the controller will require removing the controller from the VXI card cage, and then removing the side cover of the controller. (However, setting jumpers is only required if reflashing the BIOS for a system BIOS upgrade, which is rarely, if ever, done.)

Gather the Equipment Needed

Before you begin the installation, you should also have the following equipment ready to use:

- VXI card cage.
- Optionally, any other interface card(s) you wish to install in the controller (such as the RADI-EX22/HP-22 GPIB interface, RADI-EX10 LAN interface, or RADI-EX07 RS-232 interface).
- PS/2-style or compatible keyboard.
- PS/2-style or compatible mouse, or a serial mouse.
- VGA or better monitor.
- Optionally, a SCSI II CD-ROM drive (such as the HP C2944 CD-ROM drive). (See the following Note.)

Note Since the VXI Pentium Controller comes pre-loaded with the Windows operating system, for Windows, and controller drivers and utilities software, you do *not* need to connect a CD-ROM drive to the controller unless you need to re-install the software for some reason.

Installing the Controller in the VXI Card Cage

Caution **You *must* power down the VXI card cage before installing the VXI Pentium Controller in the card cage. You must also leave the card cage powered-down until you have finished installing any other interface card(s) in the controller (if needed) and connecting the peripherals to the controller, as explained in the next two sections of this chapter.**

To begin the hardware installation, you must first install the VXI Pentium Controller in the VXI card cage. To do this, first *power down* the VXI card cage by turning off the power switch on the card cage. Then insert the controller into two adjacent, open slots of your VXI card cage.

Leave the VXI card cage *powered down* while you complete the hardware installation steps explained in the next two sections of this chapter. You may power up the VXI card cage (and thereby power up the VXI Pentium Controller) *only* after you have completed installing the rest of the hardware as explained in the following two sections.

Installing Other Interfaces

Note If you have no other, separately-purchased interface card(s) to install in the VXI Pentium Controller, simply skip this section and continue with the next section, "Connecting the Peripherals."

The VXI Pentium Controller includes one HP-22 GPIB interface card which is pre-installed in Slot 1 of the controller. If you purchased any other interface card(s) you wish to use with the controller, then follow the documentation provided with each interface product now to install the other interface card(s) in the controller.

Here are some simple reminders for installing any other interface card(s) in the VXI Pentium Controller:

- Note that the existing HP-22 GPIB card was pre-installed in Slot 1 (rather than in Slot 0) to provide better cable clearance for the HP-22 card's connector on the front panel of the controller. Therefore, if you wish to install only one more interface card, you may want to leave the existing HP-22 card in Slot 1 and install the new card in Slot 0.

- When removing either the face plate covering Slot 0, or the HP-22 GPIB interface card in Slot 1, loosen the two screws on the face plate or interface's connector plate with your fingers *only*. Do *not* use a wrench or any other tool to loosen these screws, as they are meant to be loosened *by hand* only.

- Be careful when handling an interface card. Hold the interface only by its exterior connector plate. Never touch any other part of the interface, particularly the connector itself. Also, protect the interface from static electricity, since static electricity can damage an interface.

- To insert an interface card into Slot 0 or 1, find the interface card guides just inside the slot opening, and then slide the interface card into the controller slot on top of these guides. To seat the card in the slot, place your thumbs on either side of the exterior connector plate and press firmly until the card “clicks” into the slot. The connector plate will be flush with the front panel of the controller when the interface is completely seated. Then tighten the screws on the interface’s exterior connector plate.

- Because the system BIOS for the controller is pre-configured for an HP-22 GPIB interface card in Slot 1 only, you will need to change the BIOS configuration if you make *any* interface card installation changes. For example, if you do any of the following, you will need to change the BIOS configuration:

  -- Add another interface card to Slot 0.

  -- Move the existing HP-22 GPIB card from Slot 1 to Slot 0.

  -- Remove the existing HP-22 GPIB card from Slot 1 and replace it with some other interface card.

You will need to make any such BIOS configuration changes after you have completed installing the VXI Pentium Controller hardware and have powered up the controller. Therefore, please continue with the next three sections of this chapter now. You will then be told to reconfigure the BIOS for any interface card additions or changes at the end of this chapter (in the “Customizing the Controller” section).

Connecting the Peripherals

Caution **Do NOT power up the VXI card cage and the VXI Pentium Controller until you make all connections to the peripherals, as explained in this section.**

Caution **Use extreme caution when connecting peripheral cables to the controller. The I/O base board of the VXI Pentium Controller provides power for peripheral devices through different pins. Making incorrect connections can damage the board and may damage the peripheral device being connected.**

The VXI Pentium Controller supports several standard, PC-compatible I/O peripherals, including a keyboard, mouse, and monitor. Before you power up the controller, you must connect the peripherals as explained in the following list. Note that you can connect the peripherals in any order.

- Connect a PS/2-style or compatible keyboard (6-pin, mini-DIN connector) to the KBRD connector.
- Connect a PS/2-style or compatible mouse (6-pin, mini-DIN connector) to the MOUSE connector.
- Optionally, connect a serial mouse to serial port COM1.
- Connect a VGA or better monitor (15-pin, D-shell connector) to the VGA connector.
- Optionally, connect a SCSI II CD-ROM drive to the SCSI port.

Powering Up the Controller

Before you power up the VXI Pentium Controller, ensure that you have completed the following steps.

1. Installed any other interface card(s) in the controller, if needed.
2. Connected all peripherals (including a keyboard, mouse, and monitor) to the controller.
3. Checked the power supply to the controller.

When you have completed the previous steps, power up the controller by simply turning on the power switch to the VXI card cage in which the controller is installed.

Note Please refer to Chapter 6, “Troubleshooting,” if the RUN or TEST LED on the front panel of the controller is not lit after you power up the controller.

The first time you power up the VXI Pentium Controller, it automatically installs the pre-loaded Windows 95 or Windows NT operating system, as well as the pre-loaded for Windows software, on the controller’s hard disk. This installation process will take approximately 15 minutes to complete. During the installation, you will be prompted to enter your name, your company’s name, and your 10-digit, Windows Certificate of Authenticity (COA) number to register the operating system software on the hard disk (in the Windows 95 or Windows NT Registry, as appropriate).

When the installation process is complete, the controller will display the **`I/O Config`** utility for the for Windows software. This means that all the pre-loaded software is now completely installed on your controller. Now continue on to the next section to configure the controller’s interfaces via this configuration utility.

Note If the controller does *not* successfully install the software, please contact your HP Service and Support Center for assistance.

Configuring the Interfaces in the Software

Before you can use the for Windows software (including HP VISA and/or HP SICL) on your controller, you must configure the controller’s I/O interfaces within the software via the **`I/O Config`** utility. In particular, you will need to configure the VXI, HP-22 GPIB, and RS-232 interfaces, as well as any other interface(s) you have installed in the controller.

For information on how to do this, see the section titled “Configuring the HP I/O Libraries” in Chapter 2 of the *HP I/O Libraries Installation and Configuration Guide for Windows.* (This software manual was shipped with your controller.) This section in Chapter 2 of that manual explains how to use the **`I/O Config`** utility to configure your controller’s I/O interfaces for use with the VISA and SICL libraries.

Also note that each dialog box in the **`I/O Config`** utility has online help. Simply click on the **`Help`** button to obtain the online help for a particular dialog box.

When you have finished configuring the interfaces within **`I/O Config`**, a dialog box will be displayed which asks if you wish to restart your system now (that is, have the Windows operating system automatically reboot now), or if you wish to restart it yourself later. Restart your system now.

Once Windows finishes rebooting, go on to the next section.

Note Your VXI Pentium Controller may also require additional driver support. This depends on the ability of third-party vendors to incorporate their most current drivers into the Windows 95 or the Windows NT version 4.0 operating system.

In particular, note that AMD Ethernet drivers may not be pre-configured on your controller. Instead, unconfigured AMD Ethernet drivers may be supplied either on an update disk or in a subdirectory on your hard disk. Refer to any README.TXT files on any such disks or in any such subdirectories if you are unsure, or if you experience difficulties enabling Ethernet on your VXI Pentium Controller.

Customizing the Controller

Note If you have not changed or added to the VXI Pentium Controller's hardware in any way, you do not need to customize your controller. Simply skip this section and go on to the next section, "Where to Go Next."

Now that you have completed installing the VXI Pentium Controller and have configured the controller's interfaces in **`I/O Config`**, there are several customization steps that you may still need to complete, depending on your controller's particular hardware configuration. Such customization is *only* necessary if you changed or added to the controller hardware in any way (for example, if you added any other interface card(s) to the controller).

Read through the following list and complete any additional customization steps, as needed, to complete your controller's installation and configuration.

1. If you added a second HP-22 GPIB interface card to an HP E623x VXI Pentium Controller running Windows 95, you need to free up an IRQ (interrupt request) line for this second HP-22 card. Please follow the procedure in Appendix D, "Freeing Up IRQ3 for a Second HP-22," todo this.

   As explained at the end of the procedure in Appendix D, after you have freed-up IRQ3 in Windows 95, you must run **`I/O Config`** again (*before* you reboot Windows 95) to configure the second HP-22 card for the software.

2. If you installed any other interface card(s) in the controller (other than the HP-22 GPIB card pre-installed in Slot 1), or if any other default configuration values in the system BIOS may not be suitable for your controller's particular hardware setup, please go to Chapter 3, "Using the BIOS Setup Utility," to set the appropriate configuration values in the system BIOS.

3. If any default configuration values in the Start-Up Resource Manager (SURM) may not be suitable for your controller's particular hardware setup, please go to Chapter 4, "Using the Start-up Resource Manager (SURM)." The SURM provides resource manager capabilities that configure the devices in the system at power-up. The SURM program is an interactive program you must use to set the appropriate configuration values in the SURM for your controller.

4. If any default configuration values in the **`DEVICES`** file may not be suitable for your controller's particular VXI hardware setup, please go to Chapter 5, "Using the VXI Device Configurator." The VXI Device Configurator is an interactive program you can use to set the appropriate configuration values for VXI devices in the **`DEVICES`** file.

If you would rather manually edit the **`DEVICES`** file directly (instead of using the VXI Device Configurator), please go to Appendix E, "The DEVICES File."

Where to Go Next

Once you have completed configuring the VXI Pentium Controller's interfaces in **`I/O Config`** and have customized the controller, if needed, you are ready to begin using the HP VISA or HP SICL library on the controller. The following subsections list the documentation you should follow next to use each library.

HP VISA Documentation

Note If you are using new instruments or are developing new I/O applications or instrument drivers, HP recommends that you use VISA rather than SICL.

To use the HP VISA library for I/O applications, please refer to the following documentation:

- *HP VISA User's Guide* explains how to use VISA to develop I/O applications on Windows.
- *HP VISA Quick Reference Guide for C Programmers* helps you find VISA function syntax information quickly.
- HP VISA online help is provided in the form of Windows Help.
- HP VISA example programs are provided online to help you develop VISA applications more easily.

HP SICL Documentation

To use HP SICL for I/O applications, please refer to the following documentation:

- *HP SICL User's Guide for Windows* explains how to use SICL to develop I/O applications on Windows.
- *HP SICL Reference Manual* provides the function syntax and description of each SICL function.
- *HP SICL Quick Reference Guide for C Programmers* helps you find SICL function syntax information quickly if you are programming in C/C.
- *HP SICL Quick Reference Guide for Visual BASIC Programmers* helps you find SICL function syntax information quickly if you are programming in Visual BASIC.
- HP SICL online help is provided in the form of Windows Help.
- HP SICL example programs are provided online to help you develop SICL applications more easily.

Chapter 3

Using the BIOS Setup Utility

About this Chapter

This chapter explains the various menus, sub-menus, and fields in the BIOS (Basic Input/Output System) Setup utility that you can use to configure the HP E623x VXI Pentium Controller hardware. Since the controller is shipped with all BIOS Setup values pre-configured, you only need to use this chapter if you want to make changes to the pre-configured, default BIOS value settings.

This chapter presents each configuration menu, sub-menu, and field in the sequence that you would encounter it for the first time. Note that on-line help is also available in the **`Item Specific Help`** area of each menu and sub-menu.

Accessing BIOS Setup

Nonvolatile CMOS RAM in the controller maintains the BIOS settings that you save. The BIOS uses these settings to initialize the hardware. You can access BIOS Setup only during the system reset process. To access BIOS Setup, press the **F2** function key as the system boots.

Navigating BIOS Setup Menus

Within BIOS Setup, use the cursor (arrow) keys to navigate from menu to menu and to move between fields in a menu. Use the up and down cursor keys to move from field to field in a menu. Use the right and left cursor keys to move from menu to menu, as listed in the menu bar at the top of the BIOS Setup screen. If you leave a menu and then return, the active field is always at the top of the menu. If you select a sub-menu and then return to the main menu, note that you return to that sub-menu heading.

Main BIOS Setup Menu

The Main BIOS Setup Menu is shown below.

```
      PhoenixBIOS Setup - Copyright 1985-95 Phoenix Technologies Ltd.

  Main    Advanced     Power     EXM   Exit

                                              Item Specific Help
    System Time:                [16:17:18]
    System Date:                [03/01/96]   <Tab>, <Shift-Tab>, or
    Diskette A:                 [1.44 Mb, 3½"] <Enter> selects field.
    Diskette B:                 [Not Installed]
  > IDE Adapter 0 Master        (C: 1083 Mb)
  > IDE Adapter 0 Slave         (None)
    Video System:               [EGA / VGA]
  > Memory Cache:
  > Memory Shadow:
  > Boot Sequence:              [A: then C:]
  > Numlock:                    [Off]

    System Memory:              640 Kb
    Extended Memory:             31 Mb

F1  Help      ↑↓ Select Item  -/+   Change Values     F9  Setup Defaults
ESC Exit      ←→ Select Menu  Enter Select > Sub-Menu F10 Previous Values
```

Figure 3-1. Main BIOS Setup Menu

The fields in this menu and its various sub-menus are as follows.

System Time: and System Date: To change the values for these fields, simply move to each field and type in the desired entry. Use the **Tab** key to move from hour to minutes to seconds, or from month to day to year.

Diskette A: and Diskette B: These fields identify the type of 3.5-inch disk drive installed as the A: and B: drives. The BIOS defaults to **1.44 Mb**, **3 ½"** for drive A:, and **Not Installed** for drive B:. Other possible values are: **360 Kb**, **720 Kb**, **1.2 Mb**, and **2.88 Mb**.

IDE Adapter 0 Master: and IDE Adapter 0 Slave: Sub-Menus These fields are headings for sub-menus that allow you to enter complete disk drive information. Once the information is entered for the drive, the entry in the Main Menu shows the drive selected. For more information, see the "IDE Adapter 0 Sub-Menus" subsection later in this chapter.

Video System: This field identifies the type of CRT monitor attached to the system. The possible values are:

EGA / VGA (default)
CGA 80x25
Monochrome

Memory Cache Sub-Menu The term "Memory Cache" refers to the technique of caching BIOS images. For more information, see the "Memory Cache Sub-Menu" subsection later in this chapter.

Memory Shadow Sub-Menu The term "Memory Shadow" refers to the technique of copying information from an extension ROM into DRAM and accessing it in this alternate memory location. The controller restricts what memory is available for shadowing because of the special requirements for SCSI and the Universe™

PCI/VME bridge. For more information, see the “Memory Shadow Sub-Menu” subsection later in this chapter.

Boot Sequence: Sub-Menu

The Boot Sequence Sub-Menu allows you to change the boot delay and boot sequence, and to disable several displays during the boot process, such as the SETUP prompt, POST (power-on self-test) errors, floppy drive check, and summary screen. Once the boot sequence has been set, it is displayed in this field of the Main Menu. For more information, see the “Boot Sequence Sub-Menu” subsection later in this chapter.

Numlock: Sub-Menu (Keyboard Features)

This sub-menu enables or disables various keyboard features, including enabling the **Num Lock** key, enabling the key click, and setting the keyboard’s auto-repeat rate and delay. The Numlock setting is displayed in this field in the Main BIOS Setup Menu. For more information, see the “Keyboard Features Sub-Menu” subsection later in this chapter.

System Memory:

This field is read-only (not editable) and displays the amount of conventional memory (below 1 MB). No user interaction is required. Note that the amount is actually less than 640 KB since some memory is used for the Extended BIOS Data Area, which is required for PS/2 mouse usage.

Extended Memory:

This field is also read-only and displays the amount of extended memory (above 1 MB). No user interaction is required.

IDE Adapter 0 Sub-Menus

There are two IDE Adapter 0 Sub-Menus for the primary and secondary hard disk controllers, each having a master and slave drive screen. The detailed characteristics of the drive connected to the adapter is available in the IDE Adapter 0 Sub-Menu, which is shown below.

```
        PhoenixBIOS Setup - Copyright 1985-95 Phoenix Technologies Ltd.

 Main

      IDE Adapter 0 Master (C: 1038 Mb)           Item Specific Help

   Autotype Fixed Disk:         [Press Enter]    <Tab>, <Shift-Tab>, or
                                                 <Enter> selects field.
   Type:                        [User] 1083 Mb
   Cylinders:                   [ 2100]
   Heads:                       [   16]
   Sectors/Track:               [   63]
   Write Precomp:               [None]

   Multi-Sector Transfers:      [16 sectors]
   LBA Mode Control:            [Enabled]
   32 Bit I/O:                  [Disabled]
   Transfer Mode:               [Fast PIO 4]

F1  Help      ↑↓ Select Item   -/+   Change Values     F9  Setup Defaults
ESC Exit      ←→ Select Menu   Enter Execute Command   F10 Previous Values
```

Figure 3-2. IDE Adapter 0 Sub-Menus

Autotype Fixed Disk:

This is a function which is used when setting up new disks. It allows BIOS Setup to determine the proper settings of the disk based on information on the disk. This information is detected by Setup for drives that comply with ANSI specifications. Press the **Enter** key to invoke this function.

Existing (formatted) disks must be set up using the same parameters that were used originally when the disk was formatted. If necessary, the specific

Cylinder, Head, and Sectors/Track information, as listed on the label attached to the drive at the factory, must be entered manually on this sub-menu using the `User` Type, which is described below.

Type: Select **`None`** if there is no IDE hard disk drive for this adapter. In the case where there is an IDE disk but the **`Autotype`** function (above) cannot be employed, then select the **`User`** type and enter the correct drive values for Cylinders, Heads, Sectors/Track, and Write Precomp (precompensation) for the drive.

Multi-Sector Transfers: This field allows you to configure the System BIOS to read ahead by the specified number of sectors whenever a disk access is performed. This has the effect of reading more data at once to reduce the absolute number of discrete disk reads performed by the operating system, which may increase system performance. The possible values are: **`Disabled`**, or **`2`**, **`4`**, **`8`**, or **`16 sectors`** (default). Note that autotyping may change this value if the hard disk reports that it supports block accesses.

LBA Mode Control: When enabled, this field allows the System BIOS to reference hard disk data as logical blocks instead of using the traditional Cylinders/Heads/Sectors (CHS) method. The **`Enabled`** value can only be used if both the hard disk being configured and the operating system support Logical Block Addressing (LBA). If disabled, then CHS mode is used. Note that autotyping may change this value if the hard disk reports that it supports LBA. The default is **`Enabled`**.

32-bit I/O: When enabled, this field allows the System BIOS to access the hard disk controller with 32-bit I/O accesses, increasing system performance. This field is not affected by autotyping. The default is **`Disabled`**.

Transfer Mode: This field sets the mode that the System BIOS uses to access the hard disk. The possible values are:

`Standard`
`Fast PIO 1`
`Fast PIO 2`
`Fast PIO 3`
`Fast PIO 4` (default)
`Fast DMA`

Older hard disks only support the **`Standard`** setting. Newer hard disks adhering to Fast ATA or Enhanced IDE specifications may support the **`Fast PIO`** (programmed I/O) modes. The **`Fast DMA`** mode makes full use of the onboard bus mastering hard disk controller and should yield the highest performance when used in conjunction with multi-tasking operating systems that support it. Note that autotyping may change this value depending on the transfer modes that the hard disk reports it supports.

Memory Cache Sub-Menu

This sub-menu controls the cachability of certain memory regions, as well as the settings of the Level 2 (L2) cache. The Memory Cache Sub-Menu is shown below.

```
        PhoenixBIOS Setup - Copyright 1985-95 Phoenix Technologies Ltd.

 Main

                      Memory Cache                     Item Specific Help

  External Cache:               [Enabled]          <Tab>, <Shift-Tab>, or
                                                   <Enter> selects field.
  Cache System BIOS Area:       [Enabled]
  Cache Video BIOS Area:        [Enabled]

F1  Help     ↑↓ Select Item   -/+   Change Values     F9  Setup Defaults
ESC Exit     ←→ Select Menu   Enter Select > Sub-Menu F10 Previous Values
```

Figure 3-3. Memory Cache Sub-Menu

External Cache: This field enables or disables the Level 2 (L2) cache. If this cache is disabled, system performance will suffer. The default is **Enabled**.

Cache System BIOS Area: This field enables or disables caching of the System BIOS area in the 0E4000H through 0FFFFFH DRAM area. If this cache is disabled, system performance will suffer. The default is **Enabled**.

Cache Video BIOS Area: This field enables or disables caching of the VGA BIOS area in the 0C0000H through 0C7FFFH region. If this cache is disabled, system performance will suffer. The default is **Enabled**.

Memory Shadow Sub-Menu

The term “shadowing” refers to the technique of copying code, such as BIOS extensions, from ROM into DRAM and accessing them from DRAM. This allows the CPU to access the BIOS extensions faster and generally increases system performance if many calls to the BIOS extensions are made. The Memory Shadow Sub-Menu is shown below.

```
        PhoenixBIOS Setup - Copyright 1985-95 Phoenix Technologies Ltd.

  Main

                  Memory Shadow                          Item Specific Help

    System Shadow:                Enabled              <Tab>, <Shift-Tab>, or
    Video Shadow:                [Enabled]             <Enter> selects field.

    Regions with Legacy Expansion ROMs:

    D000-D3FF:                   [Rom]
    D400-D7FF:                   [Rom]
    D800-DBFF:                   [Rom]
    DC00-DFFF:                   [Rom]

F1  Help       ↑↓ Select Item   -/+   Change Values      F9  Setup Defaults
ESC Exit       ←→ Select Menu   Enter Select > Sub-Menu  F10 Previous Values
```

Figure 3-4. Memory Shadow Sub-Menu

System Shadow: This field is read-only (not editable) since the System BIOS is always shadowed.

Video Shadow: While it is advisable to leave video shadowing enabled, it is possible to disable video shadowing for system performance reasons. The default is **`Enabled`**.

Regions with Legacy Expansion ROMs: These fields enable a ROM or disable shadowing for the associated memory region. There are four 16 KB areas where expansion ROMs can be shadowed:

D000 - D3FF
D400 - D7FF
D800 - DBFF
DC00 - DFFF

The default for each of these is **`ROM`**.

The shadow regions should be used only if an EXMbus card is installed in the system that contains a BIOS extension (ROM), although there is no effect on the system if a region is shadowed that does not contain a BIOS extension. Note that each shadow region in the Setup menu is 16 KB in size. Multiple shadow regions may have to be enabled if the BIOS extension to be shadowed is larger than 16 KB.

Boot Sequence Sub-Menu

The Boot Sequence Sub-Menu allows you to change options for the boot sequence. This sub-menu is shown below.

```
        PhoenixBIOS Setup - Copyright 1985-95 Phoenix Technologies Ltd.

 Main

                  Boot Options                       Item Specific Help

   Boot Delay:             [  0]                   <Tab>, <Shift-Tab>, or
   Floppy Check:           [Disabled]              <Enter> selects field.
   Boot Sequence:          [A: then C:]
   SETUP Prompt:           [Enabled]
   POST Errors:            [Enabled]
   Summary Screen:         [Enabled]

F1  Help      ↑↓ Select Item  -/+   Change Values    F9  Setup Defaults
ESC Exit      ←→ Select Menu  Enter Select > Sub-Menu F10 Previous Values
```

Figure 3-5. Boot Sequence Sub-Menu

Boot Delay: This field sets the system to delay booting for a time period in seconds. This allows for long start-up times on boot devices that spin up slowly. The default is **`0`** (zero) seconds.

Floppy Check: This field enables or disables the floppy (3.5-inch disk) drive search during the boot. To speed up booting, the floppy check should be disabled. It is still possible to boot from the A: drive even with the floppy check disabled. The default is **`Disabled`**.

Boot Sequence: This field defines how the system treats the floppy A: drive when booting. Booting can occur either from a floppy disk in the A: drive, or directly from the C: fixed disk drive. To reduce the amount of time required to boot, set the boot sequence to **`C: only`**. Note that the C: drive may be either an IDE or SCSI drive. The possible values are:

`A: then C:` (default) Used to boot from the floppy drive or, if no floppy disk is present in the A: drive, to boot from the C: drive. This is useful for troubleshooting the operating system if a boot disk is in drive A:.

`C: then A:` Used to boot from the C: drive or, if not present, to boot from the A: drive.

`C: only` Used to boot from the C: drive without searching for an A: drive.

SETUP Prompt: This field enables or disables the message **`Press F2 to enter Setup`**. Even if the message is disabled, you can still press the **`F2`** function key at the appropriate time to enter the Main BIOS Setup Menu. To speed up booting, disable the SETUP Prompt. The default is **`Enabled`**.

POST Errors: When enabled, this field is used to stop progress during the boot process if the POST (power-on self-test) encounters errors. Otherwise, the system continues to attempt to boot despite any start-up error messages that are displayed. Note that this field only affects those errors that are configured at build-time to halt the system. The default is **`Enabled`**.

Summary Screen: This field is used to enable or disable a summary of the system configuration, which is displayed before the operating system starts to load. To speed up booting, disable the Summary Screen. The default is **`Enabled`**.

Keyboard Features Sub-Menu

This sub-menu, which is accessed from the Numlock: field in the Main BIOS Setup Menu, enables or disables various keyboard features. The Keyboard Features Sub-Menu is shown below.

```
          PhoenixBIOS Setup - Copyright 1985-95 Phoenix Technologies Ltd.

  Main

                     Keyboard Features                          Item Specific Help

    Numlock;                                    [Off]          <Tab>, <Shift-Tab>, or
    Key Click:                                  [Disabled]     <Enter> selects field.
    Keyboard auto-repeat rate:                  [30/sec]
    Keyboard auto-repeat delay:                 [1/4 sec]

F1  Help      ↑↓ Select Item   -/+   Change Values      F9  Setup Defaults
ESC Exit      ←→ Select Menu   Enter Select > Sub-Menu  F10 Previous Values
```

Figure 3-6. Keyboard Features Sub-Menu

Numlock: This field enables or disables the Numlock feature of the keyboard at boot time. When enabled, Numlock permits the use of the keypad numbers on the keyboard. The default is **`Off`**, which disables the Numlock key at boot time.

Key Click: This field enables or disables the key click feature on the keyboard. When enabled, the system produces an audible click each time a key is pressed. The default is **`Disabled`**.

Keyboard auto-repeat rate: This field sets the auto-repeat rate when a key is held down on the keyboard. The rates range from 2-30 per second. The default rate is **`30/sec`**.

Keyboard auto-repeat delay: This field sets the delay between when a key is pressed and when the auto-repeat feature begins. Options are **`1/4`**, **`1/2`**, **`3/4`**, and **`1`** (one) second. The default delay is **`1/4 sec`**.

Advanced Menu

The Advanced Menu contains settings for integrated peripherals, memory shadow, cache, and large disk access mode. The Advanced Menu is shown below.

```
        PhoenixBIOS Setup - Copyright 1985-95 Phoenix Technologies Ltd.

    Main   [Advanced]    Power     EXM   Exit

 +-------------------------------------------------+-----------------------+
 |                  Warning!                       |  Item Specific Help   |
 |  Setting items on this menu to incorrect values |<Tab>, <Shift-Tab>, or |
 |  may cause your system to malfunction.          |<Enter> selects field. |
 |                                                 |                       |
 | >Integrated Peripherals:                        |                       |
 | >Advanced Chipset Control:                      |                       |
 |  Plug & Play O/S:                   [No]        |                       |
 |  Reset Configuration Data:          [No]        |                       |
 |                                                 |                       |
 |  Level 1 Cache:                     [Enabled]   |                       |
 |                                                 |                       |
 |  Large Disk Access Mode:            [DOS]       |                       |
 +-------------------------------------------------+-----------------------+

 F1  Help      ↑↓ Select Item   -/+   Change Values      F9  Setup Defaults
 ESC Exit      ←→ Select Menu   Enter Select > Sub-Menu  F10 Previous Values
```

Figure 3-7. Advanced Menu

Integrated Peripherals Sub-Menu

This sub-menu is used to configure the onboard serial (COM) and parallel (LPT) ports, as well as the onboard disk controllers. For more information, see the “Integrated Peripherals Sub-Menu” subsection later in this chapter.

Advanced Chipset Control Sub-Menu

This sub-menu is used to configure the chipset. For more information, see the “Advanced Chipset Control Sub-Menu” subsection later in this chapter.

Plug & Play O/S:

If enabled, this field informs the System BIOS that the operating system which is booted supports Plug and Play. This forces the Plug and Play portion of the System BIOS to only configure motherboard devices and those peripherals that are necessary for booting (display, hard disk, and so forth), the rest being left to the operating system to configure. The default is **No** (disabled).

Reset Configuration Data:

If enabled, this field updates the Extended System Configuration Data (ESCD) block residing in FBD main block #2. This is necessary the first time a system is turned on or if the ESCD becomes corrupted. The default is **No** (disabled). This field is automatically reset to **No** after the ESCD is updated.

Level 1 Cache:

When enabled, this field speeds up the processor by enabling the Level 1 (L1) cache. If the L1 cache is disabled, system performance will suffer. The default is **Enabled**.

Large Disk Access Mode:

If a hard disk larger than 528 MB is being used, this field should be set either to **DOS** (the default) if running MS-DOS, or to **Other** if using a different operating system. Setting this field to **DOS** causes the system BIOS to perform cylinder/head translation if the drive is configured in BIOS Setup to have more than 1024 cylinders. This allows MS-DOS systems to use hard disks up to 8 GB (1024C X 255H X 63S) in size without special drivers or LBA.

Integrated Peripherals Sub-Menu

This sub-menu is used to configure the onboard serial (COM) and parallel (LPT) ports, as well as the onboard disk controllers. The Integrated Peripherals Sub-Menu is shown below.

```
        PhoenixBIOS Setup - Copyright 1985-95 Phoenix Technologies Ltd.

            Advanced

 Integrated Peripherals                                  | Item Specific Help
                                                         |
   COM port:                      [3F8, IRQ 4]           |<Tab>, <Shift-Tab>, or
   LPT port:                      [378, IRQ 5]           |<Enter> selects field.
   LPT Mode:                      [Bi-Directional]       |
   Diskette controller:           [Enabled]              |
   Local Bus IDE adapter:         [Enabled]              |

F1  Help     ↑↓ Select Item   -/+   Change Values      F9  Setup Defaults
ESC Exit     ←→ Select Menu   Enter Select > Sub-Menu  F10 Previous Values
```

Figure 3-8. Integrated Peripherals Sub-Menu

COM port: This field configures the serial port labeled “COM1” on the front panel. The defaults for this COM port are I/O base **`3F8H`** and **`IRQ4`**.

LPT port: This field configures the parallel port labeled “LPT” on the front panel. The defaults for this LPT port are I/O base **`378H`** and **`IRQ5`**.

LPT Mode: This field sets the mode under which the LPT port operates. The selections are:

`Output only`
`Bi-Directional` (default)
`ECP`

Diskette Controller: This field enables or disables the onboard 3.5-inch disk (floppy) drive controller. The default is **`Enabled`**.

Local Bus IDE adapter: This field enables or disables the onboard PCIbus IDE hard disk controller. The default is **`Enabled`**.

Advanced Chipset Control Sub-Menu

This sub-menu allows you to control selected settings for the chipset that affect performance or function. The Advanced Chipset Control Sub-Menu is shown below.

```
        PhoenixBIOS Setup - Copyright 1985-95 Phoenix Technologies Ltd.

           Advanced

              Advanced Chipset Control                    Item Specific Help

    DRAM Speed:                    [60ns]               <Tab>, <Shift-Tab>, or
    DMA Aliasing:                  [Enabled]            <Enter> selects field.
    8-bit I/O Recovery:            [4.5]
    16 bit I/O Recovery:           [4.5]
    IRQ12 used by:                 [PS/2 Mouse]
    ECC/Parity Config:             [Disabled]

F1  Help    ↑↓ Select Item   -/+   Change Values     F9  Setup Defaults
ESC Exit    ←→ Select Menu   Enter Select > Sub-Menu F10 Previous Values
```

Figure 3-9. Advanced Chipset Control Sub-Menu

DRAM Speed: This field sets the speed of the installed DRAM SODIMMs. Selecting 70 ns for 60 ns SODIMMs decreases performance. Selecting 60 ns for 70 ns SODIMMs is invalid. The default is **`60ns`**.

DMA Aliasing: This field allows I/O accesses to the range 90-9FH (except 92H) to alias to 80-8FH. If an ISAbus device uses the address range 90-9FH, then this field must be disabled to access the device. The default is **`Enabled`**.

8-bit I/O Recovery: This field selects the number of ISAbus SYSCLKs to be inserted by the chipset between 8-bit back-to-back I/O accesses. Increasing the number of clocks decreases I/O performance but may allow slow devices to be accessed properly. Values can range from **`3.5`** through **`11.5`** SYSCLKs, in 1 SYSCLK increments. The default is **`4.5`** SYSCLKs.

16-bit I/O Recovery: This field selects the number of ISAbus SYSCLKs to be inserted by the chipset between 16-bit back-to-back I/O accesses. Increasing the number of clocks decreases I/O performance but may allow slow devices to be accessed properly. Values can range from **`3.5`** through **`7.5`** SYSCLKs, in 1 SYSCLK increments. The default is **`4.5`** SYSCLKs.

IRQ 12 used by: This field selects the routing of IRQ12. For systems without a PS/2 mouse, this field may be set to **`ISA bus`** to allow an ISAbus peripheral to use this interrupt line. Systems using a PS/2 mouse must have this field set to **`PS/2 Mouse`** (the default) for the mouse to operate correctly.

ECC/Parity Config: This field configures the DRAM controller to use no parity (**`Disabled`**), parity (**`Parity`**), or Error Checking and Correction (**`ECC`**) when accessing DRAM. Use of parity or ECC may improve system reliability since DRAM errors are likely to be detected by the chipset. Use of ECC allows for the detection of single and dual bit errors and the correction of single bit errors during DRAM reads. The parity and ECC selections require that all SODIMMs be x36 instead of x32. The no parity (disabled) selection can use either x32 or x36 SODIMMs. The default is **`Disabled`**. (Note that the VXI Pentium Controller is shipped with non-parity x32 DRAM.)

Power Management Menu

This menu provides control over the power management facilities. As shown below, only about one-half of the Power Management Menu fields are visible at any one time; however, for completeness, all of the menu fields are listed and explained in this section. Use the **Page Up** and **Page Down** keys to display the other page of fields.

Note that the supported states for system BIOS Power Management are: Fully On, Standby Mode (partial power reduction), and Suspend Mode (maximum power reduction). Also note that the Power Management features are not implemented in the default setups.

```
        PhoenixBIOS Setup - Copyright 1985-95 Phoenix Technologies Ltd.

     Main     Advanced    [ Power ]    EXM    Exit

   APM:                          [Disabled]       Item Specific Help
   Power Savings:                [Disabled]
                                                <Tab>, <Shift-Tab>, or
   Standby Timeout:              Disabled       <Enter> selects field.
   Suspend Timeout:              DIsabled
   Standby CPU Speed:            MAX
   Fixed DIsk Timeout:           Disabled
   CRT:                          ON

   Standby Timer Reset Events
   Keyboard:                     [Enabled]
   Mouse:                        [PS/2 (IRQ12)]

   Standby Break Events
   IRQ0:                         [Disabled]
   IRQ1:                         [Auto]
   IRQ3:                         [Auto]

F1  Help     ↑↓ Select Item   -/+   Change Values     F9  Setup Defaults
ESC Exit     ←→ Select Menu   Enter Select > Sub-Menu F10 Previous Values
```

Figure 3-10. Power Management Menu

APM: This field enables or disables Advanced Power Management (APM). The default is **`Disabled`**.

Power Savings: This field enables and selects the kind of power management, or it disables power management. The possible values are:

`Disabled` (default)
`Customize`
`Maximum`
`Medium`
`Minimum`

Standby Timeout: This field enables and sets the inactivity duration required to elapse before the system is placed into Standby Mode, or it disables the Standby Timeout. The possible values are:

`Disabled` (default)
`2 minutes`
`15 minutes`
`30 minutes`
`1 hour`
`2 hours`
`3 hours`
`4 hours`

Suspend Timeout: This field enables and sets the inactivity duration required to elapse before the system is placed into Suspend Mode from Standby Mode, or it disables the Suspend Timeout. The values are the same as for the Standby Timeout, listed above. The default is **`Disabled`**.

Standby CPU Speed: This field enables or disables the changing of the CPU speed based upon the current power management state. The possible values are:

`Disabled`
`LOW`
`MEDIUM`
`HIGH`
`MAX` (default)

Fixed Disk Timeout: This field enables and sets the inactivity duration of fixed disk accesses required to elapse before the system shuts off the disk drive, or it disables the Fixed Disk Timeout. The possible values are: **`Disabled`** (default), or **`1`**, **`2`**, **`3`**, **`4`**, **`5`**, **`10`**, or **`16`** minutes.

CRT: This field enables or disables power management of the CRT during system entry or exit into or from Standby Mode. The possible values are: **`OFF in Standby`** and **`ON in Standby`**. The default is **`ON`**.

Standby Timer Reset Events This group of fields enables or disables whether or not activity from the specified device causes the Standby Timer to be reset or not. The default is **`Enabled`**.

Keyboard: This field enables or disables keyboard activity to reset the Standby Timer or not. The possible values are: **`Enabled`** (default) and **`Disabled`** .

Mouse: This field enables or disables mouse activity to reset the Standby Timer or not. The possible values are:

`Disabled`
`PS/2 (IRQ12)` (default)
`COM1 (IRQ4)`
`COM2 (IRQ3)`

Standby Break Events This group of fields enables or disables a Standby Break Event for the specified IRQ. A Standby Break Event allows the system to run at full speed for the duration of the specified IRQ. Note that no such event is associated with IRQ2.

IRQ0-15: Each of these fields enables or disables the Standby Break Event for each IRQ. The possible values are: **`Disabled`** and **`Auto`**. The default setting for IRQ 1, 3, 4, and 12 is **`Auto`**. The default setting for all other IRQs is **`Disabled`**.

Standby Wakeup Events This group of fields enables or disables the keyboard or mouse to cause a Standby Wakeup Event -- that is, these fields allow keyboard or mouse activity to return the system to full speed.

Keyboard: This field enables or disables the Standby Wakeup Event for the keyboard. The possible values are: **`Enabled`** (default) and **`Disabled`**.

Mouse: This field enables and selects the IRQ that the mouse is bound to for the purposes of mouse activity detection for power management, or disables the Standby Wakeup Event for the mouse. The possible values are:

`PS/2 (IRQ12)` (default)
`COM1 (IRQ4)`
`COM2 (IRQ3)`
`Disabled`

EXM Menu

The fields in this menu allow you to change the settings for the EXM module in the system. The HP E623x VXI Pentium Controller comes with an HP-22 GPIB interface module installed in EXM Slot 1 by default. The EXM Menu is shown below.

```
        PhoenixBIOS Setup - Copyright 1985-95 Phoenix Technologies Ltd.

     Main     Advanced      Power     [EXM]  Exit

  EXM Slot 0                                         Item Specific Help
      ID:                    [FF]
      Option Byte 1:         [00]                  <Tab>, <Shift-Tab>, or
      Opiton Byte 2:         [00]                  <Enter> selects field.

  EXM Slot 1
      ID:                    [D9]
      Option Byte 1:         [FF]
      Option Byte 2:         [00]

F1  Help      ↑↓ Select Item  -/+   Change Values     F9  Setup Defaults
ESC Exit      ←→ Select Menu  Enter Select > Sub-Menu F10 Previous Values
```

Figure 3-11. EXM Menu

ID: This field selects the EXM-ID for the HP-22 GPIB expansion module. The default is **FF** for Slot 0 and **D9** for Slot 1.

Option Byte 1: This field selects the I/O address selection for the HP-22. The default is **FF**, which corresponds to the I/O address 370-377, 770-777, B70-B77, and F70-F77. Other possible values are:

F9 250-257, 650-657, A50-A57 and E50-E57
FB 270-277, 670-677, A70-A77, and E70-E77
FD 350-357, 750-757, B50-B57, and F50-F57

To disable the HP-22, enter **00**.

Option Byte 2: This field is set to **00** (disabled/unused).

To configure a second HP-22 card in EXM Slot 0, make sure that it is not sharing the same I/O address as the first card pre-installed in EXM Slot 1. For information on configuring a second card, see the hardware reference manual shipped with the card.

Exit Menu

The fields in this menu allow you to save value settings and exit BIOS Setup, or abandon value changes and exit. The Exit Menu is shown below.

```
        PhoenixBIOS Setup - Copyright 1985-95 Phoenix Technologies Ltd.

     Main    Advanced     Power     EXM  [Exit]

  ┌──────────────────────────────────────────────────┬───────────────────────┐
  │                                                  │  Item Specific Help   │
  │  Save Changes & Exit                             ├───────────────────────┤
  │  Exit Without Saving Changes                     │<Tab>, <Shift-Tab>, or │
  │  Get Default Values                              │<Enter> selects field. │
  │  Load Previous Values                            │                       │
  │  Save Changes                                    │                       │
  │                                                  │                       │
  └──────────────────────────────────────────────────┴───────────────────────┘
  F1  Help     ↑↓ Select Item   -/+   Change Values     F9  Setup Defaults
  ESC Exit     ←→ Select Menu   Enter Execute Command   F10 Previous Values
```

Figure 3-12. Exit Menu

Save Changes & Exit This field is used to save into CMOS the values that have been entered. The system then immediately reboots with the new values.

Exit Without Saving Changes This field is used to discard the changes just made and revert to the state when BIOS Setup was first entered. The system reboots with the old values.

Get Default Values This field is used to reset the BIOS Setup values to the original, default values that were set at the factory, before any suppliers or other end-users made changes. This is useful if you are having problems with the system and suspect changes made to the BIOS are the cause.

Load Previous Values This field is used to load the system with the previous values before an editing session started. This is useful if you lose track of edits and want to revert to the previous editing session, but not all the way back to the defaults.

Save Changes This field is used to save the edits made during a session. You do not exit, and you can continue to make changes.

Chapter 4
Using the Start-Up Resource Manager (SURM)

About this Chapter

This chapter explains how to use the Start-Up Resource Manager (SURM) to further customize the VXI Pentium Controller, if needed.

The SURM program performs the functions of the VXI resource manager. These functions include:

- Initializing the VXI interface hardware and processing systemconfiguration files.
- Identifying all devices in the system.
- Configuring the A24 and A32 address maps.
- Managing the system self-test and diagnostic sequences.
- Assigning symbolic names to devices.
- Configuring the system's commander/servant hierarchies.
- Allocating VXIbus IRQ lines.
- Configuring mainframe extenders.
- Initiating normal system operation.

After performing its resource manager functions, SURM outputs its decisions, actions, and any detected errors to the controller display and to ASCII text files. When the SURM screen is displayed, keyboard function keys move you between reports, let you select a menu that allows you to re-name devices, and let you obtain help.

Required Environment and Related Files

By default, SURM files are located in the directory **`C:\SICL`***XX* (that is, either **`C:\SICL95`** on Windows 95, or **`C:\SICLNT`** on Windows NT). If the SURM files are installed to another location, specify their location using the SURMDIR environment variable. The following lists the required SURM files at their default locations:

C:\SICL*XX***\VXI\DB\VXIMANUF**	The VXI manufacturer name database.
C:\SICL*XX***\VXI\DB\VXIMODEL**	The VXI devices model database.
C:\SICL*XX***\VXI\SURM.EXE**	Invokes the SURM GUI (interface).
C:\SICL*XX***\VXI\SURM32.EXE**	Helper for **SURM.EXE**. Can be run as a stand-alone console application for SURM.
C:\SICL*XX***\VXI\SURM.RC**	The runtime configuration file.
C:\SICL*XX***\VXI\SURMHELP.TXT**	Text displayed when you press the **F1** function key, when the SURM is running.

The following files are created each time the SURM runs and are not included on the release CD-ROM:

C:\SICL*XX***\VXI\DB\RESRCMGR**	A file output by SURM that contains system resource information.
C:\SICL*XX***\VXI\SURM.ERR**	Errors and warnings detected by the SURM the last time it ran. Also included are progress messages if the **/V** option was specified.
C:\SICL*XX***\VXI\SURM.LOG**	Current system configuration.

Operation Sequence

The SURM is one of the first application programs to run after the operating system boots. It should be executed from the Windows Startup group. Executing SURM more than once may invalidate existing Online Resource Manager (OLRM) data. However, the system is configured correctly. The following table lists the main actions taken by the SURM when it executes.

Table 4-1. SURM Actions

Step	Action	Description
1.	Controller initialization	SURM initializes the controller's bus interface and sets the controller's unique logical address (ULA) to 0 (zero). A ULA of 0 designates the controller as the resource manager.
2.	Read configuration files	SURM reads the **`SURM.RC`** file, the command line, and the **`DEVICES`** file for system information and how to report its actions and decisions.
3.	Static device identification	SURM searches the 256 VXIbus ULAs for statically configured VXIbus devices.
4.	Non-VXIbus device configuration	SURM examines the **`DEVICES`** file (maintained by the VXI Device Configurator) for non-VXIbus devices in the system. SURM uses this information to avoid resource assignment conflicts.
5.	Dynamic ULA configuration	SURM assigns ULAs to all dynamically configurable devices.
6.	Slot search	SURM determines the device slot and each slot's state. Valid slot states are: empty/non-VXI, operating, non-operating, and indeterminate. Also, the VXI specification level to which each device conforms is identified and reported.
7.	Address map configuration	SURM assigns address ranges to devices that have memory in A24 or A32 space.
8.	Self-test management	SURM examines each device's VXI status register to determine if the device's self-test successfully completed. If the self-test did not successfully complete before the self-test timer (see the **`SURM.RC`** file) expires, the SURM writes an error message and sets the SYSFAIL INHIBIT and RESET flags in the device's control register.

Table 4-1. SURM Actions (continued)

Step	Action	Description
9.	Symbolic naming	SURM assigns symbolic names to all devices. User-supplied symbolic names are taken from the **`DEVICES`** file (use the VXI Device Configurator to add symbolic names to the **`DEVICES`** file). If a name is not supplied for a device, SURM assigns default names of the form **`vdev`***x*, where *x* is a decimal number.
10.	Commander/servant initialization	SURM assigns a hierarchical relationship of commander and servant devices, using the hierarchy information taken from the **`DEVICES`** file. Use the VXI Device Configurator to specify the commander/servant hierarchy.
11.	Assign interrupts and handlers	SURM assigns all interrupters and handlers to the system devices.
12.	Get manufacturer name and model numbers	SURM reads manufacturer and model information from the files **`VXIMANUF`** and **`VXIMODEL`** and adds that information to the configuration file.
13.	Display SURM screen	Displays the SURM screen for user review and device name verification or change. Eliminate this step by using a command line option or runtime configuration file switch.
14.	Exit	Writes system configuration information to **`SURM.LOG`**, writes errors to **`SURM.ERR`**, and removes the SURM screen.

The SURM Screen

The following figure shows a typical SURM startup screen. This includes the VXI System Configuration Report.

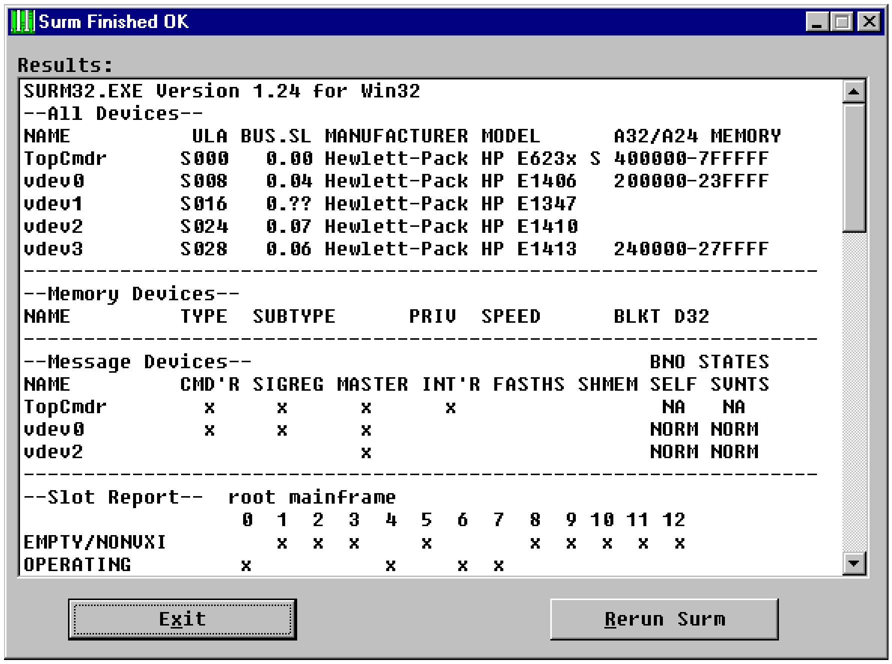

Figure 4-1. SURM Startup Screen

VXI System Configuration Report

The VXI System Configuration Report shows information about all devices and slots in the system. It also includes information about decisions made by the SURM. There are nine parts to the report:

- All Devices
- Memory Devices
- Message Devices
- Slot Report (one for each mainframe)
- Commander/Servant Hierarchy
- Interrupt Map
- ULA Usage and Bus Traversal Map
- A24 Usage and Bus Traversal Map
- A32 Usage and Bus Traversal Map

The following listing is an example VXI System Configuration Report from

SURM for a VXI Pentium Controller.

Figure 4-2. Example SURM VXI System Configuration Report

```
SURM32.EXE Version 1.24 for Win32
--All Devices--
NAME           ULA BUS.SL MANUFACTURER MODEL      A32/A24 MEMORY
TopCmdr       S000   0.00 Hewlett-Pack HP E623x S 400000-7FFFFF
vdev0         S008   0.04 Hewlett-Pack HP E1406   200000-23FFFF
vdev1         S016   0.?? Hewlett-Pack HP E1347
vdev2         S024   0.07 Hewlett-Pack HP E1410
vdev3         S028   0.06 Hewlett-Pack HP E1413   240000-27FFFF
------------------------------------------------------------------
--Memory Devices--
NAME          TYPE  SUBTYPE      PRIV  SPEED      BLKT D32
------------------------------------------------------------------
--Message Devices--                                  BNO STATES
NAME          CMD'R SIGREG MASTER INT'R FASTHS SHMEM SELF SVNTS
TopCmdr         x     x      x      x                 NA   NA
vdev0           x     x      x                       NORM NORM
vdev2                        x                       NORM NORM
-----------------------------------------------------------------
--Slot Report--  root mainframe
                  0  1  2  3  4  5  6  7  8  9 10 11 12
EMPTY/NONVXI         x  x  x     x        x  x  x  x  x
OPERATING         x           x     x  x
NON-OPERATING
INDETERMINATE
VXI 1.2                             x
-----------------------------------------------------------------
--Commander/Servant Hierarchy--
TopCmdr
  +-- vdev0
  +-- vdev1
  +-- vdev2
  \-- vdev3
--Interrupt Map--                  IRQ
  Device Name   Interrupter      7654321    Handler
H TopCmdr                        |||||+----  [H1]
H TopCmdr                        ||||+-----  [H2]
H TopCmdr                        |||+------  [H3]
H TopCmdr                        ||+-------  [H4]
H TopCmdr                        |+--------  [H5]
H TopCmdr                        +---------  [H6]
2 vdev0                          ||||||+---  [PH1]
  vdev0                          ||||||| NC  [PH2]
  vdev0                          ||||||| NC  [PH3]
  vdev0                          ||||||| NC  [PH4]
  vdev0                          ||||||| NC  [PH5]
  vdev0                          ||||||| NC  [PH6]
  vdev0                          ||||||| NC  [PH7]
-----------------------------------------------------------------
--ULA usage and bus traversal map--
bus ula
  0 000  TopCmdr
    ...
  0 008  vdev0
    ...
  0 016  vdev1
```

Figure 4-2. Example SURM VXI System Configuration Report (continued)

```
    ...
  0 024  vdev2
    ...
  0 028  vdev3
    ...
  0 255  *vacant*
-----------------------------------------------------------------
--A24 usage and bus traversal map--
bus low-high addresses
  0 000000-1fffff  *vacant*
  0 200000-23ffff  vdev0
  0 240000-27ffff  vdev3
  0 280000-3fffff  *vacant*
  0 400000-7fffff  TopCmdr
  0 800000-ffffff  *vacant*
-----------------------------------------------------------------
--A32 usage and bus traversal map--
bus low-high addresses
  0 00000000-ffffffff  *vacant*
```

All Devices

The All Devices portion of the VXI System Configuration Report lists all devices in the system. The following table identifies and describes its entries.

Table 4-2. All Devices Configuration Report

Entry	Description	Definition
`NAME`	Device symbolic name	Obtained from the VXIbus Configurator or assigned by the SURM. Device names assigned by the SURM have the form `vdev`*x*, where *x* is a decimal number.
`ULA`	Unique logical address	Valid first characters are: `S` = Static logical address `D` = Dynamic logical address `N` = Non-VXIbus device logical address The next three digits are the device's decimal ULA.
`BUS`	Bus number	The bus where the device is installed. The root mainframe bus number is always zero. All other bus numbers are the same as the ULA of the mainframe extender that connects the local bus to the parent bus.

Table 4-2. All Devices Configuration Report (continued)

Entry	Description	Definition
`SL`	Slot number	The slot where the device is installed. Contains two question marks (**`??`**) if the device is external to the root mainframe.
`MANUFACTURER`	Device manufacturer's name	The device manufacturer's name as contained in the **`VXIMANUF`** file. If the manufacturer's name is not found in the **`VXIMANUF`** file, the number read from the device is displayed in brackets.
`MODEL`	Device model name	The device model name as contained in the **`VXIMODEL`** file. If the manufacturer's model number is not found in the **`VXIMODEL`** file, the number read from the device is displayed in brackets.
`A32/24 MEMORY`	Starting and ending address of the memory space assigned to the device	Hexadecimal digits. Six-digits represent A24 memory space. Eight digits represent A32 memory space.

Memory Devices

The Memory Devices portion of the VXI System Configuration Report lists all VXI memory devices in the system. The following table identifies and describes its entries.

Table 4-3. Memory Devices Configuration Report

Entry	Description	Definition
NAME	Device symbolic name	Obtained from the Configurator or assigned by SURM. Device names assigned by the SURM have the form **vdev***x*, where *x* is a decimal number.
TYPE	Memory type	Valid values are: **RAM** = random access memory **ROM** = read-only memory **other** = other memory type **reser** = reserved
SUBTYPE	Memory subtype	Valid values are: **non-volatile** = non-volatile memory **elec-prog** = electrically programmable nothing = no subtype
PRIV	Privilege	Valid values are: **supv** = responds to supervisor-level address modifiers **both** = responds to user-level and supervisor-level address modifiers
SPEED	Memory device access speed range	Valid values are: decimal value in nanoseconds or **device dep** (device dependent.)
BLKT	Block transfer support	Checkmark specifies block transfer support.
D32	32-bit transfer support	Checkmark specifies that the device supports 32-bit transfers. Otherwise, it supports only 16-bit transfers.

Message Devices

The Message Devices portion of the VXI System Configuration Report lists all the message-based devices in the system. The following table identifies and describes its entries.

Table 4-4. Message Devices Report

Entry	Description
NAME	Device symbolic name.Obtained from the Configurator or assigned by SURM. Device names assigned by the SURM have the form **vdev***x*, where *x* is a decimal number.
CMD'R	Device is a commander.
SIGREG	Device has a signal register.
MASTER	Device is a bus master.
INT'R	Device is an interrupter.
FASTHS	Device supports fast-handshake mode.
SHMEM	Device supports shared-memory message protocol.
BNO STATES **SELF SVNTS**	Identifies the begin-normal-operation (BNO) states of the device (SELF) and its servants (SVNTS). Valid entries for each field are: **NA** = not applicable (device is not message-based) **CONF** = configure state **NORM** = normal operation

Slot Report

The Slot Report of the VXI System Configuration Report lists the state of each VXIbus slot. The following table defines the valid slot states.

Table 4-5. Valid VXIbus Slot States

Slot State	Description
EMPTY	Slot is empty or contains a port of a multi-slot device.
OPERATING	Slot occupied and operational.
NON_OPERATING	Slot occupied but not operational.
INDETERMINATE	Cannot determine the device's state.
VME	Slot contains a VME device.
VXI 1.2	Slot contains a VXI device that conforms to VXI specification revision 1.2.

Commander/Servant Hierarchy

The Commander/Servant Hierarchy portion of the VXI System Configuration Report shows the commander/servant hierarchy of all devices in the system. Its format follows a tree structure. The first device listed is the top-level commander.

Interrupt Map

The Interrupt Map portion of the VXI System Configuration Report shows the connections of the interrupters and the handlers in the system to the VXIbus IRQ lines. The following table identifies and describes its entries.

Table 4-6. Interrupt Map Configuration Report

Entry	Description
Reason for interrupt connectivity	Valid entries are: **C** = configuration data required this connection **!** = configuration data required this connection but it conflicts with another configured connection. **1** = first allocation pass (each PH commander gets one line) **2** = second allocation pass (PH devices get remaining lines) **S** = servant tracking. The line was allocated to a commander because one of its servants was configured to interrupt on it (1st pass) **T** = tracking commander. A servant was configured to interrupt on this line because its commander is handling this line. **H** = hard-wired
`Device Name`	Device symbolic name. Obtained from the Configurator or assigned by SURM. Device names assigned by the SURM have the form **`vdev`***x*, where *x* is a decimal number.
`Interrupter`	The first character specifies the interrupter type: **I** = interrupter **PI** = programmable interrupter The next digit is the interrupter number. The interrupter number is for devices with multiple interrupt controllers.
`IRQ`	A map of interrupter and handler to IRQ lines. The horizontal line on the device's name line ends at the connected IRQ line.
`Handler`	The first character specifies the handler type: **H** = handler **PH** = programmable handler The next digit is the handler number. The handler number is for devices with multiple handlers.

ULA Usage and Bus Traversal Map

The ULA Usage and Bus Traversal Map portion of the VXI System Configuration Report shows ULA to bus mapping. When the system contains mainframe extenders, it also show the busses traversed to reach each device from the root mainframe.

Each line can contain up to four pieces of information: bus number, ULA, traversal map, and device name. Bus number, ULA, and device name are obvious. The traversal map appears only for mainframe extenders and is displayed between the ULA and device name. It contains bus IDs, one for each bus traversed and in order of traversal from the root mainframe to the device. Vacant ULAs show mainframe extender logical address window boundaries.

A24 and A32 Usage and Bus Traversal Maps

The A24 and A32 Usage and Bus Traversal Maps display the range of addresses in use by devices mapped to A24 and A32 space, respectively. They also include bus traversal information for devices in mainframe extenders.

Error Messages Report

The Error Messages Report lists errors detected by the SURM. It can also contain progress messages if they are enabled when the SURM starts. Refer to the "Start-Up Resource Manager (SURM) Error Messages" section of Chapter 6 for a complete listing of SURM error messages.

Progress Message Area

The Progress Message Area displays messages as the SURM executes. These messages normally flash by and cannot be read. However, if the SURM hangs, the displayed message is useful in finding the cause of the failure.

Running the SURM

SURM Configuration (SURM.RC)

The **SURM.RC** file is the SURM runtime configuration file. It contains switches that direct SURM operation and treatment specifications that handle unique requirements of certain devices. In most cases, you will not be making changes to this file. It has the following syntax:

```
<line>            :[<comment>
                   |<switch>
                   |<variable>
                   |<treatment-spec>]<newline>

<comment>         :”#”<non-newline-character>*

<switch>          :<identifier>

<variable>        :<identifier>=<value>

<identifier>      :<letter>[(<letter>|<digit>|”_”)*]

<treament-spec>   :<device-id>”:”<item>[“,”<item>]*

<device-id>       :<manuf-code>”.”<model-code>

<manuf-code>      :<number>

<model-code>      :<number>

<number>          :<digit>*

<item>            :<device-switch>
                   |<device-variable>

<device-switch>   :<identifier>

<device-variable>:(<identifier>=<value>)
                  |(<identifier>=<manf-code>”.”<model-code>)

<value>           :<identifier>
                  |([<sign>]<number>)

<sign>            :”+”|”-”
```

SURM Switches

The **<switch>** lines direct SURM operation, but do not override options entered in the Windows 95 or Windows NT Startup group's SURM command line. The following table lists the valid switch entries and describes switch action.

Table 4-7. Valid Switch Entries for SURM.RC

Switch	Command Line Option	Action
TRANSLATE_LOG_ FILE_TO_ASCII (default)	—	Translates IBM extended ASCII characters in **SURM.LOG** into equivalent ASCII characters.
USE-TWELVE_BIT _MODEL_CODES	—	Truncate all model codes to 12 bits, including model codes read from the **VXIMODEL** file. Useful for certain VXI devices whose upper four bits of the model code are wrong or unpredictable.
NO_OLRM	**/NO**	Do not initialize OLRM, and do not write system configuration to **SURM.LOG**. OLRM will not operate.
IGNORE_OLRM_ ERRORS	—	Do not report errors that occur during initialization.
DC_BEFORE_ CHILD_BUSSES	—	Perform dynamic configuration before configuring child busses.
DO_MX_PRESCAN (default)	—	Scan for mainframe extenders before configuring any bus and close all found windows. Use when running SURM without removing power from VXI mainframes.
NEVER_SEND_ANO	—	Prevents the SURM from sending ABORT NORMAL OPERATION (ANO) to return devices to the CONFIGURE state. Send ANO is the default.

SURM Command Line Options

The following table lists command line options and describes actions.

Table 4-8. Valid Command Options for SURM.RC

Command Line Option	Action
/PE	Display SURM screen if any errors are detected
/PN	Never display SURM screen.
/V	Log progess messages to **SURM.ERR**.
/R	Force SURM to run SURM32.
/D	Force SURM not to run SURM32.

Variables

The **<variable>** line sets timeout limits for self-test and word serial communications. The following table lists the valid **<variable>** constants.

Table 4-9. Valid Variables for SURM.EXE

Variable	Valid Values	Definition
NO_MODID_OK	EPC-8, EPC-6, EPC-5	Lists the controllers that don't have MODID capability.
SELFTEST_ TIMEOUT= (default is 30)	Any decimal number	Specifies the number of seconds to wait for a device self-test to pass. SURM forces this value to be at least 5.
WORD_SERIAL_ TIMEOUT= (default is 30)	Any decimal number	Specifies the number of seconds to wait before generating a word serial communications timeout. Set it large enough to allow all servants to boot (see VXI specification 1.4).

Treatment Specifications

The **<treatment-spec>** line configures system devices according to **<device-switch>** and **<device-variable>**. The device to configure is identified by **<device-id>**. Treatment specifications apply to all devices in the system with the specified **<device-id>**. These treatment specifications allow the SURM to handle unique features of certain VXI devices during configuration.

The following tables list the valid **<device-switch>** constants and the valid **<device-variable>** constants, respectively.

Table 4-10. Valid Device Switches for SURM.EXE

Device Switch	Action
NO_WS_COMMANDS	SURM does not send most word serial commands to the device. Only the BEGIN NORMAL OPERATION command is sent. No response is collected.
CONTRL_REG_RMW	Causes SURM to access the device control register with read-modify-write cycles. Device dependent bits remain unchanged after the VXI defined bits are modified.
IGNORE_WS_ERRORS	SURM ignores word serial command errors during interrupt configuration.
CONTROL_REG_DEF_ZERO	SURM writes zeros to the device dependent bits when information is written to its control registers.
OFFSET_REG_EPC	Applies special memory handling procedures to the device. Set by the default **SURM.RC** file where necessary.
ASSUME_MESSAGE_DEVICE	SURM treats the device as a message-based device regardless of its device class bits.
EXTENDED_SELFTEST_OK	Do not warn about the device being in extended self-test.

Table 4-11. Valid Device Variables for SURM.RC

Device Variable	Valid Values	Definition
ASSUME_HANDLERS	0 to 7	The number of handlers assigned to this device.
ASSUME_INTERRUPTERS	0 to 7	The number of interrupters assigned to this device.
SLOT_0_DEVICE	RADISYS_SZM	The type of slot-0 controller.
REPLACEMENT_IDS	Any arbitrary manufacturer model number	Replaces the manufacturer model number with another arbitrary pair. When used, no other device variable or device switch is valid.

Obtaining Online Help

The SURM online help contains information about error messages, the VXI System Configuration Report, the runtime configuration file (**SURM.RC**), and the SURM command line options. To obtain online help, press the **F1** function key.

Quitting the SURM

By default, the SURM stops execution and displays system configuration information before it writes this information to **SURM.LOG** and **SURM.ERR**. To exit the SURM program:

1. Press the **Esc** key until the SURM displays the Extended Key usage block.
2. Press the **Esc** key.

Logging Progress Messages

Progress messages report SURM progress and decisions it makes as it executes. Normally they flash by and you cannot read them, unless the SURM stops execution. To log the progress messages to the **SURM.ERR** file, execute the SURM with this command:

SURM /V [*options*]

Mainframe Extenders

The SURM routes memory regions to devices in remote mainframes if the **DEVICES** file contains a bus number in a device record. If there is no bus number, SURM assumes that the device is in the root mainframe. Note that the VXI Device Configurator cannot set a device's bus number.

Each VXIbus or MXIbus has a unique identification that refers to the physical location of devices. The bus identification of the root bus is zero. The identification of the non-root bus is equal to the ULA of the mainframe extender which connects the non-root bus to its parent bus.

For each mainframe extender, the SURM sets ACFIN, ACFOUT, SFIN, SFOUT, SRIN, and SROUT. This forwards the utility bus in both directions. Refer to your mainframe extender manual for additional information.

All interrupt lines are forwarded across every mainframe extender that has an interrupt configuration register. Interrupt lines are forwarded upward unless the handler for an interrupt line is located downward from the mainframe extender being configured. The result is that each interrupt handler can be interrupted from any bus.

The SURM does *not* perform trigger configuration.

Chapter 5

Using the VXI Device Configurator

About this Chapter

The VXI Device Configurator is an interactive Windows application that defines VXI devices that the VXI Pentium Controller’s programs can access. It is especially useful with systems that also contain VME modules. Use the VXI Device Configurator to:

- Name system devices.
- Set system device parameters that cannot be obtained dynamically.
- Set commander/servant hierarchy.
- Map interrupters and handlers to devices.
- Modify manufacturer codes.
- Modify manufacturer model codes.

The VXI Device Configurator can be executed directly or by selecting the **`Device Config...`** button during VXI interface configuration using the configuration utility, **`I/O Config`**.

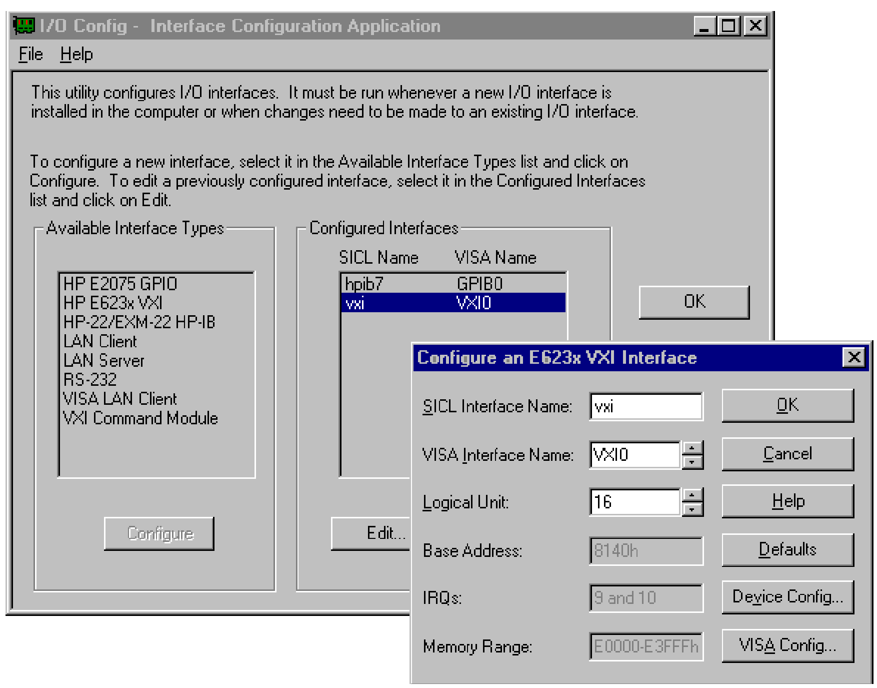

Figure 5-1. Starting VXI Device Configurator

The VXI Device Configurator modifies the **DEVICES** file, which is then used to communicate device names and parameters to programs and to the SURM. The SURM uses **DEVICES** file information to initialize system devices. In addition, the VXI Device Configurator can modify the manufacturer code file, **C:\SICL*XX*\VXI\DB\VXIMANUF**, and the manufacturer model database file, **C:\SICL*XX*\VXI\DB\VXIMODEL**. (Where **SICL*XX*** is either **SICL95** on Windows 95, or **SICLNT** on Windows NT).

For more information about the SURM, see Chapter 4, “Using the Start-Up Resource Manager (SURM).” For more information about the **DEVICES** file, see Appendix E, “The DEVICES File.”

VXI Device Configurator Start-up Window

The following figure shows the VXI Device Configurator start-up window. It consists of five pull-down menus.

- The File menu allows you to exit the VXI Device Configurator window application and to display information about the VXI Device Configurator.
- The Devices menu allows you to name and configure VXI and VME devices.
- The VXI Control menu allows you to configure the VXI commander/servant hierarchy. You can also map interrupters and interrupt handlers to the VXI interrupt lines.
- The Database menu allows you to modify the manufacturer code database, the manufacturer model name database, and manufacturer model number database.
- The Help menu provides on-line information for the VXI Device Configurator.

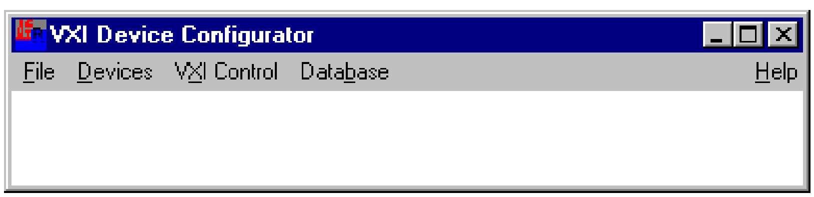

Figure 5-2. VXI Device Configurator Start-up Window

Naming VXI Devices

Selecting the VXI item from the Devices menu displays the Edit VXI device dialog box (see the following figure). This dialog box names or renames VXI devices. It is not necessary that the device physically exist in the system when you name it. This is particularly useful when the system configuration often changes.

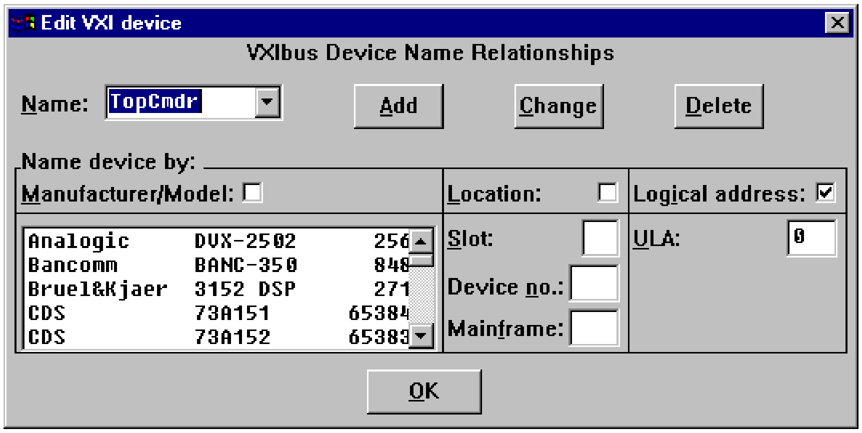

Figure 5-3. Edit VXI Device Dialog Box

The Edit VXI device dialog box contains two main boxes: the Name list box and the Name device by box. The Name list box lists the names of all defined devices. It is also the box you use to enter a new device's name. The Name device by box contains three naming options. You can use none, one, two, or three of the naming options.

Named VXI devices do not consume bus resources when they are not physically in the system, so you can name many VXI devices without affecting bus resources. The device being named is selected by specifying its manufacturer/model pair, its physical location in the system, its ULA, or any combination of these selections.

When using Manufacturer/Model to name the device, the manufacturer name and model must be in the database. If not, use the Database menu to add the manufacturer and model names.

When using Location to name the device, you must also specify at least one of the location options (Slot, Device no., or Mainframe). Slot is the physical location of the device in the mainframe. Device no. identifies the number of the device in a multi-device module (the first device is number zero). Mainframe is the ULA of the mainframe extender that contains the device.

It is not recommended to use the Logical address method of naming dynamically configured VXI devices because the device ULA is not predictable.

To add a VXI device:

1. Select the VXI item from the Devices menu.
2. Enter the new device name in the Name list box.
3. Optionally select the device naming method and enter the appropriate attributes.
4. Choose the **`Add`** button.
5. Choose the **`OK`** button to update the **`DEVICES`** file.

To change a VXI device name or its naming method:

1. Select the VXI item from the Devices menu.
2. Select the device name to change from the Name list box.
3. Select the new device naming method and enter the appropriate attributes.
4. Choose the **`Change`** button.
5. Choose the **`OK`** button to update the **`DEVICES`** file.

To delete a VXI device:

1. Select the VXI item from the Devices menu.
2. Select the device to delete from the Name list box.
3. Choose the **`Delete`** button.
4. Choose the **`OK`** button to update the **`DEVICES`** file.

Configuring VME Devices

Selecting the VME item from the Devices menu displays the Edit VME device dialog box (see the following figure). Use this dialog box to specify the VME module's attributes. Unlike VXI devices, VME modules do not include configuration registers that specify the device's system memory requirements, manufacturer code, or model ID. This dialog box also assigns names and logical addresses to VME devices. VME devices consume system resources when not installed. Therefore, it is good practice to only name VME devices that are physically in your system.

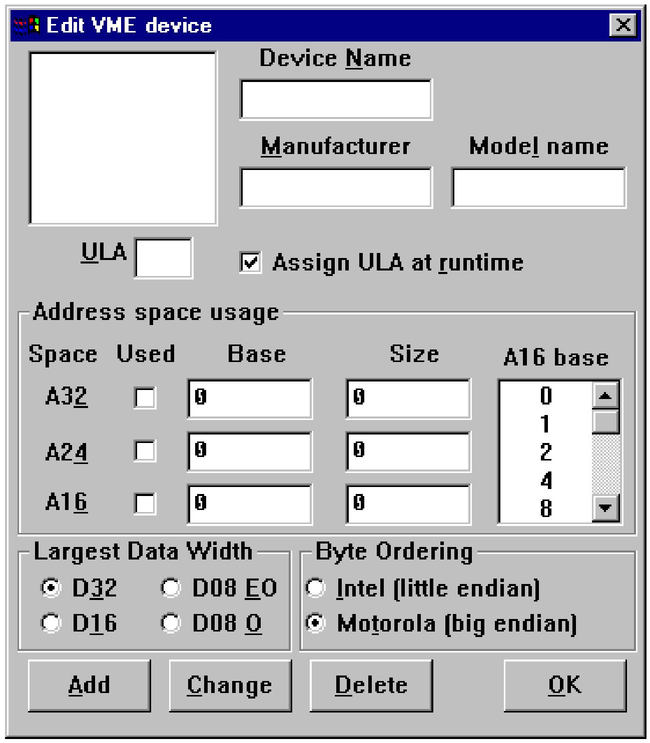

Figure 5-4. Edit VME Device Dialog Box

Entries in the Manufacturer and Model name text boxes are for reference only and are not used for name matching by the software.

The ULA box and Assign ULA at runtime are mutually exclusive.

Supplying a VME module's memory requirements is very important since it allows the SURM to avoid any conflicts when assigning memory to the VXI devices in the system. When you move to a Base or Size text box, the name and contents of the scroll box changes to reflect the selected address space and its valid values.

The Largest Data Width buttons display and select the device's data width.

The Byte Ordering buttons display and select the device's byte ordering scheme.

To add a VME device:

1. Select the VME item from the Devices menu.
2. Enter the new device name in the Device Name text box.
3. Optionally enter a manufacturer and model name.
4. Either enter a ULA or check the Assign ULA at runtime box. The OLRM uses this ULA to obtain device information.
5. Enter address space usage information.
6. Select the byte ordering and the largest data width supported by the device.
7. Choose the **`Add`** button.
8. Choose the **`OK`** button to update the **`DEVICES`** file.

To change a VME device name or its parameters:

1. Select the VME item from the Devices menu.
2. Select the device to change from the device list box.
3. Enter the new device name and/or other information.
4. Choose the **`Change`** button.
5. Choose the **`OK`** button to update the **`DEVICES`** file.

To delete a VME device:

1. Select the VME item from the Devices menu.
2. Select the device to delete from the list box.
3. Choose the **`Delete`** button.
4. Choose the **`OK`** button to update the **`DEVICES`** file.

Setting Commander/Servant Hierarchy

The Commander Hierarchy item from the VXI Control menu causes VXI Device Configurator to display the Commander/Servant Hierarchy dialog box (see the following figure). This dialog box defines and deletes a commander and its servants, adds or deletes a servant, and displays the commander of a selected device. The Commander/Servant Hierarchy dialog box has three list boxes: one lists all the system commanders, one that lists all the selected commander's servants, and one that lists all known system devices.

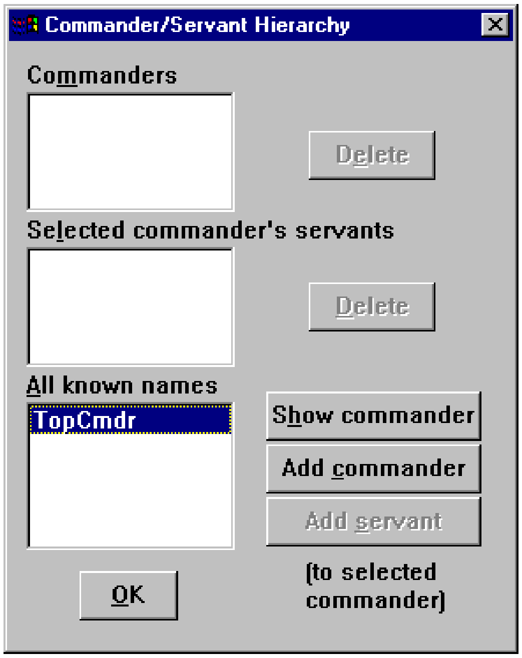

Figure 5-5. Commander/Servant Hierarchy Dialog Box

To add a commander:

1. Select the Commander Hierarchy item from the VXI Control menu.
2. Select the commander to add from the All known names list.
3. Choose the **`Add commander`** button.
4. Choose the **`OK`** button to update the **`DEVICES`** file

To add a servant to a selected commander:

1. Select the Commander Hierarchy item from the VXI Control menu.
2. Select the servant to add from the All know names list.

3. Select the commander from the Commanders list.
4. Choose the **`Add servant`** button.
5. Choose the **`OK`** button to update the **`DEVICES`** file.

To display the commander’s servants:

1. Select the Commander Hierarchy item from the VXI Control menu.
2. Select the commander from the Commanders list.
3. Choose the **`Show commander`** button.

To display a servant’s commander:

1. Select the Commander Hierarchy item from the VXI Control menu.
2. Select a servant from the All known names list.
3. Choose the **`Show commander`** button.

To delete a commander:

1. Select the Commander Hierarchy item from the VXI Control menu.
2. Select the commander from the Commanders list.
3. Choose the **`Delete`** button next to the Commanders list box.
4. Choose the **`OK`** button to update the **`DEVICES`** file.

To delete a servant:

1. Select the Commander Hierarchy item from the VXI Control menu.
2. Select the commander from the Commanders list box.
3. Select the servant from the Selected commander’s servants list.
4. Choose the **`Delete`** button next to the Selected commander’s servants list box.
5. Choose the **`OK`** button to update the **`DEVICES`** file.

Interrupter and Handler Mapping

Selecting the Interrupt Mapping item from the VXI Control menu displays the Edit Interrupt Mapping dialog box (see the following figure). This dialog box displays interrupter and interrupt handler mapping to VXIbus IRQ lines for the system devices and allows you to edit these mappings. This mapping information is used by the SURM to override the default configuration when assigning VXI interrupt lines to interrupters and interrupt handlers at system start-up.

The left half of the dialog box displays interrupter and handler mapping for the device selected in the Selected device list box, if the Assign IRQs at runtime box is not checked. If the box is checked, the mappings in the Attributes box are dimmed, indicating that you cannot change them.

The right half of the dialog box displays how the VXIbus IRQ lines are mapped to interrupters and handlers for all devices. The display is the current state and includes any changes you have entered.

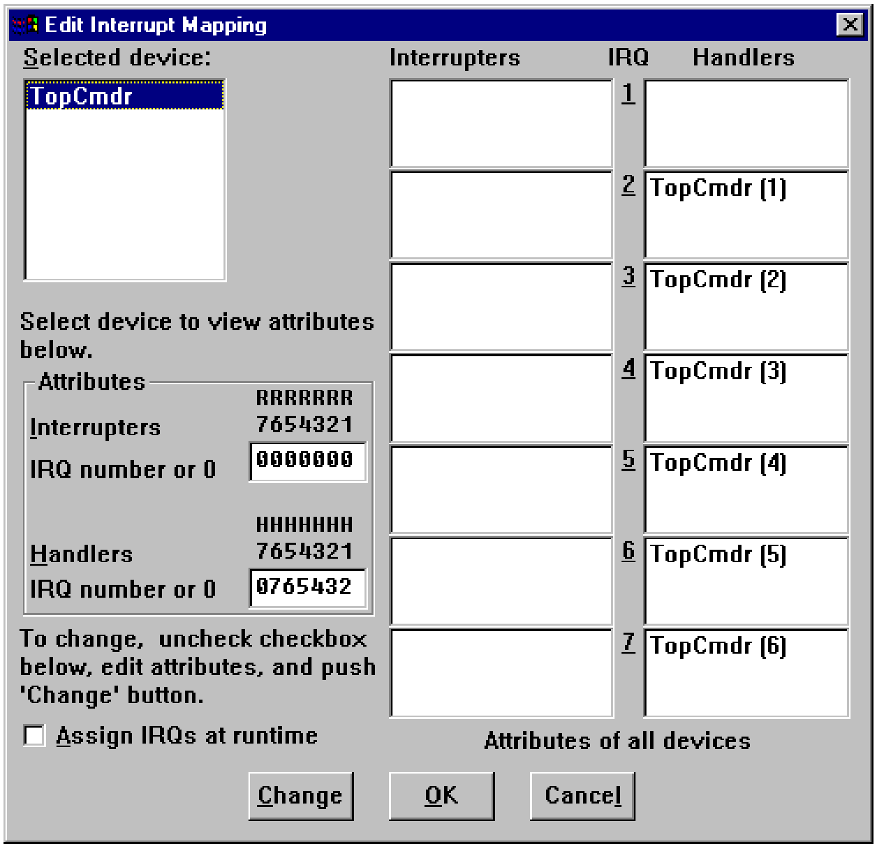

Figure 5-6. Edit Interrupt Mapping Dialog Box

To change interrupter or interrupt handler mapping:

1. Select the device to change interrupter and/or handler mapping from the Selected device list box.

2. Uncheck the Assign IRQs at runtime check box.

3. To change an interrupter to IRQ mapping, move to the Interrupters text box and enter the IRQ number to map to the interrupter below the corresponding interrupter number.

4. To change a handler to IRQ mapping, move to the Handlers text box and enter the IRQ number to map to the handler below the corresponding interrupter number.

5. Choose the **Change** button.

6. Choose the **OK** button to record the changes.

Editing the Manufacturer Code Database

Selecting the Manufacturers item from the Database menu displays the Edit Manufacturer Code Database dialog box (see the following figure). This dialog box adds, changes, or deletes the names and numbers of VXIbus device manufacturers stored in the **C:\SICL***XX***\VXI\DB\VXIMANUF** file.

The manufacturers list box displays the most recent contents of the database file plus any changes you have entered. In addition, the VXI Device Configurator creates the file **VXIMANUF.BAK** that preserves the original file. Manufacturer names are case-sensitive and are limited to 12 characters. Manufacturer codes are limited to four characters.

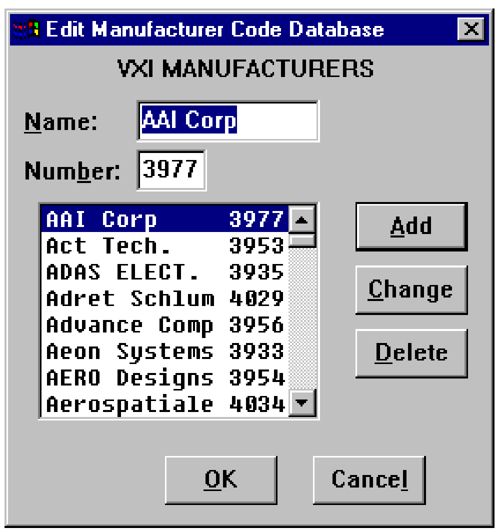

Figure 5-7. Edit Manufacturer Code Database Dialog Box

To add a manufacturer:

1. Enter the manufacturer’s name.

2. Enter the manufacturer’s ID number, in decimal. Manufacturer IDs are assigned by the VXIbus Consortium.

3. Choose the **`Add`** button.

4. Choose the **`OK`** button to update the database file.

To change a manufacturer or manufacturer number:

1. Select the manufacturer name/number to change from the list box.

2. Enter the new manufacturer name and/or the new manufacturer's number.

3. Choose the **`Change`** button.

4. Choose the **`OK`** button to update the database file.

To delete a manufacturer name/number:

1. Select the manufacturer code to delete from the list box.

2. Choose the **`Delete`** button.

3. Choose the **`OK`** button to update the database file.

Editing the Model Code Database

Selecting the Model item from the Database menu displays the Edit Model Code Database dialog box (see the following figure). This dialog box adds, changes, or deletes manufacturer model name and model numbers of VXI devices stored in the database file **`C:\SICL`*`XX`*`\VXI\DB\VXIMODEL`**.

The Manufacturer drop-down list box contains all the manufacturers described in the database file **`C:\SICL`*`XX`*`\VXI\DB\VXIMANUF`** (you cannot change the contents of this box). The main list box displays the current contents of the database file plus any changes you have entered. In addition, the VXI Device Configurator creates the file **`VXIMODEL.BAK`** that preserves the original file.

Manufacturer models are limited to 10 characters, and manufacturer model numbers are limited to five characters.

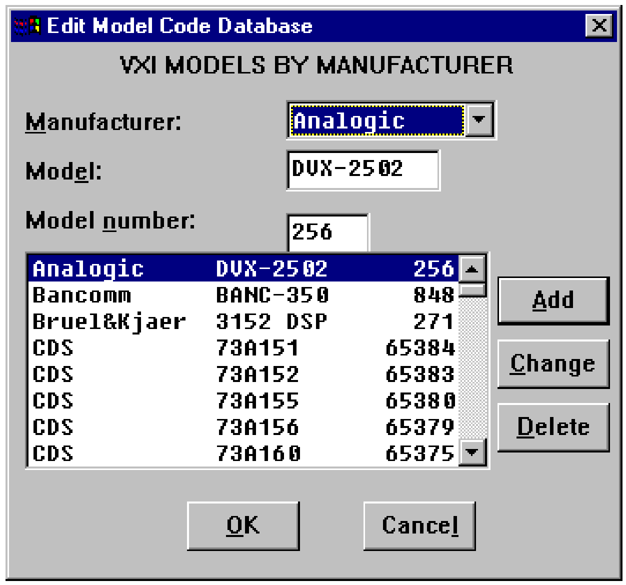

Figure 5-8. Edit Model Code Database Dialog Box

To add a model code:

1. Select the manufacturer from the Manufacturer list box or the main list box. If the manufacturer is not listed, edit the manufacturer code database to add the new manufacturer.
2. Enter the new model and model numbers. Model numbers are assigned by the device's manufacturer and are usually included in the device documentation.
3. Choose the **Add** button.
4. Choose the **OK** button to update the database.

To change a device model name and/or model number:

1. Select the model to change from the main list box.
2. Enter the new model name and/or model number.
3. Choose the **`Change`** button.
4. Choose the **`OK`** button to update the database.

To delete a model name/number pair:

5. Select the model and name/number pair to delete from the main list box.
6. Choose the **`Delete`** button.
7. Choose the **`OK`** button to update the database file.

Chapter 6
Troubleshooting

About this Chapter

This chapter helps you resolve error messages or other common problems you may encounter while installing the VXI Pentium Controller hardware. This chapter contains the following troubleshooting sections:

- If the RUN or TEST LED is Not Lit
- BIOS Error Messages
- Problems After Configuring the BIOS
- Start-Up Resource Manager (SURM) Error Messages
- If the SURM “Hangs”

Note For information on:

- The meaning of the LED indicators on the front panel of the controller, please see Appendix A.
- Using the BIOS Setup utility, please see Chapter 3.
- Using the Start-Up Resource Manager (SURM) program, please see Chapter 4.

If the RUN or TEST LED is Not Lit

After powering up the VXI Pentium Controller, if either the RUN or the TEST LED indicator on the front panel of the controller is not lit, this may indicate either that the CPU has halted (when RUN is not lit), or that the controller did not pass its self-test (when TEST is not lit).

Cycle the power to the controller to see if this problem rectifies itself. If the RUN or TEST LED indicator still is not lit, then please contact your HP Service and Support Center for assistance.

BIOS Error Messages

The following table lists the BIOS error messages you may encounter when powering up and using the VXI Pentium Controller and describes possible causes and fixes for the errors.

Note If a possible fix that is suggested in the following table requires you to run BIOS Setup, please see Chapter 3 for more information.

Table 6-1. BIOS Error Messages

BIOS Error MessagesError Message	Description
`Diskette drive A error` or **`Diskette drive B error`**	Drive A: or B: is present but fails the BIOS power-on self-test (POST) for the particular 3.5-inch disk drive. Check to see that the drive is defined with the proper disk type in BIOS Setup. Also, if the disk drive is an external drive (rather than the internal drive), ensure that the disk drive is correctly connected.
`Extended RAM Failed at offset:` *nnnn*	The extended memory is not working or is not configured properly at offset *nnnn*. Contact your HP Service and Support Center for assistance.
`Failing Bits:` *nnnn*	The hexadecimal number *nnnn* is a map of the bits at the RAM address (in System, Extended, or Shadow memory) which failed the memory test. Each 1 (one) in the map indicates a failed bit. Contact your HP Service and Support Center for assistance.
`Fixed Disk 0 Failure` or **`Fixed Disk 1 Failure`** or **`Fixed Disk Controller Failure`**	The fixed disk (internal hard disk drive) is not working or is not configured properly. Run BIOS Setup to ensure the disk type is correctly identified. If this does not fix the problem, contact your HP Service and Support Center for assistance.
`Incorrect Drive A type - run SETUP` or **`Incorrect Drive B type - run SETUP`**	The disk type for drive A: or B: is not correctly identified in BIOS Setup. Run BIOS Setup to set the correct disk type.
`Invalid NVRAM media type`	There is a problem with NVRAM (CMOS) access. Contact your HP Service and Support Center for assistance.
`Keyboard controller error`	The keyboard controller failed a system test. You may have to replace either the keyboard or the keyboard controller. Contact your HP Service and Support Center for assistance.
`Keyboard error - Keyboard not working` **`Keyboard error`** *nn*	The system BIOS discovered a stuck key and displays the scan code *nn* for the stuck key.

Table 6-1. BIOS Error Messages (continued)

BIOS Error MessagesError Message	Description
`Keyboard locked - Unlock key switch`	Unlock the system to proceed.
`Monitor type does not match CMOS - Run SETUP`	The monitor type is not correctly identified in BIOS Setup. Run BIOS Setup to set the correct monitor type.
`Operating system not found`	The Windows operating system cannot be located on either drive C: or drive A:. Run BIOS Setup to ensure drive C: (fixed disk/internal hard disk drive) and/or drive A: (the 3.5-inch disk drive) are properly identified.
`Parity Check 1`	A parity error was found in the system bus. If the system BIOS located the address where the parity error occurred, it will also display that address on the screen. Contact your HP Service and Support Center for assistance.
`Parity Check 2`	A parity error was found in the I/O bus. If the system BIOS located the address where the parity error occurred, it will also display it on the screen. Contact your HP Service and Support Center for assistance.
`Press <F1> to resume, <F2> to Setup`	This is displayed after any recoverable error message. Press the **F1** function key to start the boot process or the **F2** function key to enter BIOS Setup and change any settings.
`Press <F2> to enter SETUP`	This is not an error message. Instead, it is an optional, informational message that is displayed during the controller's power-on self-test (POST). Note that the display of this message can be turned off in BIOS Setup.
`Previous boot incomplete - Default configuration used`	The previous power-on self-test (POST) did not complete successfully. The POST then loads the default BIOS configuration values and offers to run BIOS Setup. If the failure was caused by incorrect BIOS values that were set and they are not corrected, the next boot will also likely fail. Therefore, ensure that all configuration values are re-set correctly in BIOS Setup. This error should then be cleared the next time the controller is booted.
`Real time clock error`	The real-time clock failed the system BIOS test. This may require repair of the board. Contact your HP Service and Support Center for assistance.

Table 6-1. BIOS Error Messages (continued)

BIOS Error MessagesError Message	Description
`Shadow Ram Failed at offset:` *nnnn*	The shadow RAM failed at offset *nnnn* of the 64 KB block at which the error was detected. Contact your HP Service and Support Center for assistance.
`System battery is dead -` `Replace and run SETUP`	The CMOS clock battery indicator shows the battery is dead. Have the battery replaced (contact your HP Service and Support Center for assistance). Then run BIOS Setup to reconfigure the controller.
`System cache error - Cache disabled`	The RAM cache failed the system BIOS test, and BIOS then disabled the cache. Contact your HP Service and Support Center for assistance.
`System CMOS checksum bad - run SETUP`	The system CMOS has been corrupted or modified incorrectly, perhaps by an application program that changes data stored in CMOS. Run BIOS Setup and reconfigure the controller either by restoring the default values or the previous values and/or making your own selections.
`System RAM Failed at offset:` *nnnn*	The system RAM failed at offset *nnnn* of the 64 KB block at which the error was detected. Contact your HP Service and Support Center for assistance.
`System timer error`	The timer test failed. This requires repair of the system board. Contact your HP Service and Support Center for assistance.

Problems After Configuring the BIOS

If your controller either fails, does not work properly, or displays error messages after you make configuration changes in BIOS Setup, you may be able to correct the problem by entering BIOS Setup again and restoring either the original, default BIOS configuration values, or the previous values. This can be done via either the “Get Default Values” or the “Load Previous Values” selection in the Exit Menu of BIOS Setup. For more information, see the “Exit Menu” section of Chapter 3, “Using the BIOS Setup Utility.”

If after restoring the default or previous BIOS configuration values, your controller is still not functioning properly, attempt to re-set the BIOS configuration values for your controller’s hardware setup by following the appropriate information in Chapter 3. If this still does not fix the problem, please contact your HP Service and Support Center for assistance.

Start-Up Resource Manager (SURM) Error Messages

This section contains information about interpreting SURM error messages and provides tables that list, define, and offer possible corrective actions for SURM error messages.

Error Message Interpretation

SURM error messages fall into three categories that are identified as follows (*xx* is an error number, *not* a returned error code):

E0*xx* Fatal errors that terminate the SURM program prematurely.

E1*xx* Significant errors that allow the SURM program to complete, but some devices may not be configured correctly.

E2*xx* Warning messages that inform you of possible error conditions.

Note that a SURM error message may include a return code. The return code is specific to the underlying program that generated the error.

Error Messages

This subsection contains three tables that list error messages for each category of SURM error messages: fatal, significant, and warning messages. If a posible corrective action listed requires you to run the SURM program, please see Chapter 4 for more information.

SURM Fatal Error Messages.

Table 6-2. SURM Fatal Error Messages

Error Number	Description	Corrective Action
E003	The computer running the SURM encountered a bus error while trying to access itself.	Seat computer module properly in backplane or replace computer module.
E009	The file **SURM.RC** is missing. The error message tells where the SURM expected to find it.	Place **SURM.RC** file in the directory **C:\SICL***XX***\VXI\SURM**, or in the **SURM** subdirectory under the directory specified by the SURMDIR environment variable.
E010	Another device has the same ULA as the computer running the SURM.	Reconfigure the other device to use a different ULA.

SURM Significant Error Messages.

Table 6-3. SURM Significant Error Messages

Error Number	Description	Corrective Action
E101	Not enough logical address space to assign a contiguous block of logical addresses for the devices on the module in the specified slot.	Move specified device to a lower numbered slot.
E102	The device failed its self-test. The SURM set the device state to safe.	Check for correct operation of device and correct its problem.
E103	The SURM could not find enough memory space for the device.	Remove one or more modules.
E104	The SURM could not correctly set the device's offset (address base) register.	Check for correct operation of device and correct its problem.
E105	The SURM could not correctly set the device's control or status register.	Check for correct operation of device and correct its problem.
E106	The specified device is not enabled to receive messages. If the slot number is shown as "**??**" the SURM cannot determine the device's slot.	Check device MODID line. (Note that devices without a P2 backplane connector do not have a MODID line.)
E107	The specified device identified itself as dynamically configured, but could not be found after a new ULA was assigned.	Check for correct operation of device and correct its problem.
E109	The **`DEVICES`** file describes a non-VXIbus device with the same ULA as a static VXI bus module or another non-VXIbus device. The device is ignored.	Move the static device or the non-VXI device.
E110	The data structures of the On-Line Resource Manager could not be created. The NO option is therefore assumed.	Increase the space reserved for the OLRM. Requires reconfiguration and reboot.

Table 6-3. SURM Significant Error Messages (continued)

Error Number	Description	Corrective Action
E113	A commander defined in the **DEVICES** file could not be found in the system, but its servant devices were found. The servant devices were made servants of the resource manager.	Change the configuration, or install the commander.
E114	A commander defined in the **DEVICES** file was found in the system, but its physical characteristics are not those of a valid VXIbus commander. The commander's servant devices were made servants of the resource manager.	Check device documentation for commander capability.
E115	A commander defined in the **DEVICES** file was found in the system and appears to be a valid VXIbus commander device, but the device was not ready to accept a GRANT DEVICE command.	May have resulted from a word serial timeout.Try increasing WORD_SERIAL_TIMEOUT in **SURM.RC**, and then re-run.
E117	The specified device was not the only device configured to use the specified IRQ. The device appears to be hard-wired, so the SURM could not disconnect it. This problem requires manual intervention to fix.	Reconfigure the offending device, and update the **DEVICES** file to reflect the change.
E118	The named device returned an error in its response to the READ INTERRUPTER LINE or READ HANDLER LINE command. SURM assumes that the line in question is disconnected.	Consult the device's documentation or increase the WORD_SERIAL_TIMEOUT in the **SURM.RC** file.

Table 6-3. SURM Significant Error Messages (continued)

Error Number	Description	Corrective Action
E119	An error occurred while sending (or getting the response to) the READ INTERRUPTER LINE or READ HANDLER LINE command to (or from) the named device. SURM assumes that the line in question is disconnected.	Consult the device's documentation or increase the WORD_SERIAL_TIMEOUT in the **SURM.RC** file.
E120	An error occurred while sending (or getting a response to) the READ INTERRUPTER LINES or READ HANDLER LINES command to (or from) the named device. SURM assumes that the number of lines for this device is zero.	Consult the device's documentation or increase the WORD_SERIAL_TIMEOUT in the **SURM.RC** file.
E121	The named device returned an error in its response to the ASSIGN INTERRUPTER LINE or ASSIGN HANDLER LINE command. SURM assumes that this line has its default configuration.	Consult the device's documentation or increase the WORD_SERIAL_TIMEOUT in the **SURM.RC** file.
E122	An error occurred while sending (or getting the response to) the ASSIGN INTERRUPTER LINES or ASSIGN HANDLER LINES command to (or from) the named device. SURM assumes that this line has its default configuration.	Consult the device's documentation or increase the WORD_SERIAL_TIMEOUT in the **SURM.RC** file.
E123	An error occurred while waiting for the response to the BEGIN NORMAL OPERATION command.	Consult the device's documentation or increase the WORD_SERIAL_TIMEOUT in the **SURM.RC** file. If the device is a controller, verify that the servant program is being run.

Table 6-3. SURM Significant Error Messages (continued)

Error Number	Description	Corrective Action
E124	An error occurred while sending the IDENTIFY COMMANDER command to the specified device. Only servants of the resource manager can receive this command.	Consult the device's documentation or increase the WORD_SERIAL_TIMEOUT in the **`SURM.RC`** file.
E125-E131	The specified device responded to the BEGIN NORMAL OPERATION with the indicated error.	Refer to the explanation for the specified error.
E136	The maximum number of devices (256) in a VXIbus system has been exceeded. Additional devices are ignored.	Remove extra device(s).
E138	The A24 VXIbus memory defined in the **`DEVICES`** file for the specified device is illegal.	1. Set base address to less than 0xFFFFFF. 2. Set memory size so no location exceeds 0xFFFFFF.
E139	The A32 VXIbus memory defined in the **`DEVICES`** file for the specified device is illegal.	1. Set base address to less than 0xFFFFFFFF. 2. Set memory size so no location exceeds 0xFFFFFFFF.
E141	A device was found with a model code greater than 256 (indicating a slot-0 controller), but it is not installed in slot 0.	Install the device in slot 0 or reconfigure device.
E142	More than one slot-0 controller was found.	Reconfigure system so there is only one slot-0 controller. Ensure slot-0 controller is installed in slot 0.
E144	A word serial error occurred when sending a SET UPPER MODID or SET LOWER MODID command.	Verify that slot-0 controller is operating correctly.
E145-E146	The SET UPPER MODID or SET LOWER MODID command failed.	Verify that slot-0 controller is operating correctly.

Table 6-3. SURM Significant Error Messages (continued)

Error Number	Description	Corrective Action
E147	The **SURM.RC** treatments specification SLOT_0_DEVICE contains an invalid value.	Correct SLOT_0_DEVICE variable.
E148	There is a statically configured device at address 255 that prevents the mainframe extender from mapping address 255 to the child bus for dynamic configuration.	Assign the statically configured device another address.
E149	Devices on the ancestor bus prevent enlarging its windows to include all devices on the ancestor bus.	Reconfigure system correctly.
E150	The current slot-0 device is of class extended, but not subclass mainframe extender. No other extended subclass is defined as a slot-0 controller.	Install a slot-0 controller.
E151	The slot-0 device is a memory device, not a slot-0 controller.	Install a slot-0 controller.
E155-E156	The window value read from the mainframe extender does not match the window set by the SURM.	Replace mainframe extender.
E157	SURM is unable to update the file **C:\SICL*XX*\VXI\DB\RESRCMGR**.	Check file attributes and path set by SURMDIR variable, if it is used.
E158	MXIbus topology is too deep.	Flatten MXIbus tree or free more memory for use by SURM. SURM requires about 450 KB of free memory to execute. Use MODE to reduce the screen size to 25 lines.
E159	The ABORT NORMAL OPERATION timed-out or generated an error.	Try using the **SURM.RC** treatment specification NEVER_SEND_ANO.

Table 6-3. SURM Significant Error Messages (continued)

Error Number	Description	Corrective Action
E160	A device on a mainframe extender became inaccessible because the mainframe extender window could not be expanded.	Manually set ULAs of SC devices in ancestor buses.

SURM Warning Messages

Table 6-4. SURM Warning Messages

Error Number	Description	Corrective Action
E201	A device was found with fixed logical address 255. The SURM has bypassed dynamic configuration.	If your system contains dynamically configured devices, remove the device with the fixed logical address 255, or change its logical address.
E202	The specified device was still executing its extended self-test at the time the SURMexamined it.	Wait for the device to complete the self-test, then try again.
E203	Cannot find the file **VXIMANUF**. All manufacturer names are shown as **[nnnn]**.	Install the file **VXIMANUF** in the directory **C:\SICL*XX*\VXI\DB**, or in the directory specified by the SURMDIR environment variable.
E204	Cannot find the file **VXIMODEL**. All manufacturer and model names are shown as **[nnnn]**.	Install the file **VXIMODEL** in the directory **C:\SICL*XX*\VXI\DB**, or in the directory specified by the SURMDIR environment variable.
E208	An error occurred while sending (or getting the response to) the READ PROTOCOL to (or from) the named device. SURM assumes the device in question conforms to version VXI 1.2, or earlier, of the VXI specification.	Check devices for correct installation and operation.

Table 6-4. SURM Warning Messages (continued)

Error Number	Description	Corrective Action
E209	A bad character was found in the **SURM.RC** file.	Fix the indicated line with your editor.
E210	The SURM ran out of memory to store the treatment specifications from **SURM.RC**.	Shorten **SURM.RC** file by removing unnecessary switches and treatment specifications.
E211	A **SURM.RC** file treatment specification switch is not defined.	Fix the indicated line with your editor.
E212	A **SURM.RC** file treatment specification has a syntax error.	Fix the indicated line with your editor.
E213	The **SURM.RC** file has an undefined switch.	Fix the indicated line with your editor.
E214	The **SURM.RC** file has an undefined treatment specification variable.	Fix the indicated line with your editor.
E215	The **SURM.RC** file has a syntax error in a treatment specification variable.	Fix the indicated line with your editor.
E216	The **SURM.RC** file has an out-of-range numeric value assigned to a treatment specification variable.	Fix the indicated line with your editor.
E217	The **SURM.RC** file has a syntax error in a treatment specification switch.	Fix the indicated line with your editor.
E218	The **SURM.RC** file has a syntax error in a treatment specification item. The error was detected before the SURM could distinguish the item as a switch or a variable.	Fix the indicated line with your editor.
E220	The specified device is not the only device configured to use the specified IRQ. The device was disconnected from that IRQ. If it is a commander, the device will probably be made a handler of a different line.	Correct the system configuration in the **DEVICES** file.

Table 6-4. SURM Warning Messages (continued)

Error Number	Description	Corrective Action
E222	A corrupted interrupter and/or handler map field appears in the **DEVICES** file.	Use the VXI Device Configurator to correct interrupter and/or handler map.
E224	The specified commander will not be allocated a line to handle because all lines are already handled.	Delete some handlers on other devices with the VXI Device Configurator.
E225	The specified commander did not advertise PH capability and was not configured to handle all the lines its servants were interrupting on, so interrupts from the specified servant will be ignored.	Use a different commander, or change the configuration if the commander has hard-wired handlers.
E226	The specified commander is PH capable, but did not have enough handlers to cover all the lines its servants were interrupting on.	Reconfigure servants which are hard-wired.
E230	The specified VMEbus device has been assigned A16 memory within the range reserved for VXIbus devices. This causes problems, because the logical address of a VXIbus device is configured (either dynamically or statically) such that its A16 memory overlaps with the memory specified for this device. VXIbus A16 reserved memory range is 0xC000 - 0xFFFF.	Reconfigure servants which are hard-wired.
E232	This controller has not been assigned any slave VMEbus memory. This is caused by the memory selections for VMEbus devices and/or custom devices in the **DEVICES** file. No memory chunk large enough to accommodate the controller memory remained.	Reconfigure the VME devices to leave more space for the controller.

Table 6-4. SURM Warning Messages (continued)

Error Number	Description	Corrective Action
E237	The SURM used an obsolete invocation option (switch). A new switch was substituted for the obsolete one.	Use the new switch in subsequent SURM invocations in the Windows 95 Startup group.
E238	The specified bus has no slot-0 controller.	Install a slot-0 controller on the specified bus.
E241	Missing base or size variable for an address space in **`DEVICES`**.	Edit the **`DEVICES`** file with your editor.
E242	Unable to read **`DEVICES`** file.	Check file attributes and path set by SURMDIR variable if it is used.
E243	Invalid character in **`SURM.RC`** variable.	Fix the indicated line.
E244	Invalid SURM invocation variable.	Fix the indicated line.
E245	Can't get the servant area for the named device. Servant area assumed to be 0.	Check device. There is probably a word serial error.

If the SURM "Hangs"

The SURM performs error checking, but under rare circumstances the SURM cannot complete execution and hangs. You can recognize this situation when the screen still displays the SURM banner after about 30 seconds. (In most cases, the interval to wait depends on the longest device timeout.)

When the SURM hangs, the following are the most common causes:

- The controller is not properly seated in the backplane.
- The controller has not been configured as the slot-0 controller (for example, VXI-MXI is not seated).
- The system contains one or more devices configured as slot-0 controllers.
- There is a fault on the VXIbus backplane. Check the bus-grant, bus-request, and SYSCLK lines.

Appendix A
LEDs on the Front Panel

There are seven LED indicators on the front panel of the VXI Pentium Controller which provide information about the state of the controller. The following table lists the label, color, and meaning of each LED indicator.

Table A-1. LED Indicators

Label	Color	Meaning
RUN	Green	Lit whenever a write access is made to DRAM. If not lit, either the CPU is executing entirely out of the on-chip cache, or the CPU is halted.
SYSFAIL	Red	Lit whenever the VME SYSFAIL line is asserted.
TEST	Yellow	Reflects the state of the PASS bit in the VXI registers. If not lit, then PASS is 0 (zero), and the controller has not passed its self-test.
MASTER	Green	Lit when a VME access is made by this device as a master.
SLAVE	Green	Lit when another VME device accesses the memory of this controller.
SCSI	Green	Lit when the SCSI port is busy.
IDE	Green	Lit when the IDE disk is busy.

Appendix B
I/O, Memory, and IRQ Details

This appendix provides more detailed technical descriptions of the following for the VXI Pentium Controller:

- VGA video controller, including its BIOS extension
- SCSI controller, including its BIOS extension
- Ethernet controller
- Memory map
- IRQs (interrupts)

VGA Video Controller

The VXI Pentium Controller contains the Cirrus CL-GD5436, PCI-based, super-VGA compatible video graphics controller with 2 MB of display memory. The VGA video controller resides on the PCIbus and is PCI device number 1.

The Video BIOS extension is shadowed and write-protected in the region 0x0C0000 to 0x0C7FFF (32 KB) on any system reset or power-up. The features of the Video BIOS include:

- Standard VGA modes up to 640 X 480 resolution with 16.8 million colors.
- Super-VGA modes up to 800 X 600 resolution with 16.8 million colors.
- Extended video modes up to 1280 X 1024 resolution with 256 colors (non-interlaced).

SCSI Controller

An AMD53C974A is used to implement the SCSI controller in the VXI Pentium Controller. The SCSI BIOS extension is automatically shadowed in system DRAM at memory location 0x0C8000 to 0x0CFFFF. SCSI hard disk enumeration begins with the logical drive C:. If there are two 3.5-inch disk drives connected to the ISAbus, then they will be drive A: and B:. If there is a SCSI 3.5-inch disk drive connected, then it follows the same enumeration rules as the hard disk.

The SCSI device resides on the PCIbus and is mapped as PCI device number

3. The SCSI controller uses PCIbus interrupt INTC. The default I/O base address and interrupt is 0xC180 and IRQ11, respectively.

The SCSI BIOS extension supports booting DOS, Windows 95, and Windows NT from the SCSI controller.

Ethernet Controller

The Ethernet controller is implemented using the AMD79C970A. The Ethernet controller resides on the PCIbus and is mapped as PCI device number 2. It uses PCIbus interrupt INTB. The Ethernet ID is stored in a PROM (8-pin serial data) that is connected to the AMD79C970A. The I/O base address, interrupt, and DMA channel are at 0x0300, IRQ15, and DMA5, respectively.

Network booting is *not* supported.

The Ethernet port supports only a 10-base-T interface.

Memory Map

The memory map for the VXI Pentium Controller is summarized in the following table.

Table B-1. Memory Map

Pentium Address Range[a]	Region	Cached
00000000-0009FFFF	DRAM	Yes
000A0000-000BFFFF	VGA Memory	No
000C0000-000C7FFF	Shadowed VGA BIOS	Yes
000C8000-000CFFFF	Shadowed SCSI BIOS	Yes
000D0000-000DFFFF	On-board EXMbus	No
000E0000-000E3FFF	VXI Interface Chip	No
00100000-07FFFFFF	DRAM If no DRAM, ISA memory	Yes No
08000000-0FFFFFFF	EXMbus	No
10000000-1FFFFFFF	Reserved	No
20000000-3FFFFFFF	Reserved	No
40000000-7FFFFFFF	Reserved	No
80000000-83FFFFFF	VXI Interface Chip Image 0	No
84000000-87FFFFFF	VXI Interface Chip Image 1	No
88000000-8BFFFFFF	VXI Interface Chip Image 2	No

Table B-1. Memory Map

Pentium Address Range[a]	Region	Cached
8C000000-8FFFFFFF	VXI Interface Chip Image 3	No
90000000-93FFFFFF	VXI Interface Chip Image 4	No
94000000-EFFFFFFF	Reserved	No
F0000000-FFF7FFFF	EXMbus	No
FFF80000-FFFFFFFF	Flash Boot Device	No

a.All addresses are hexadecimal.

IRQs (Interrupt Request Lines)

The following table lists the IRQs for the VXI Pentium Controller.

Table B-2. IRQs (Interrupt Request Lines)

Interrupt	Source
IRQ0	Timer
IRQ1	Keyboard Controller
IRQ2	Cascade Interrupt Input
IRQ3	COM2 (Not externally available) Use for second HP-22 on Windows 95[a]
IRQ4	COM1
IRQ5	ECP Printer Port (LPT1) Use for second HP-22 on Windows NT[b]
IRQ6	3.5-inch Disk Drive
IRQ7	Reserved for HP-22
IRQ8	Real-time Clock
IRQ9	VXI Interface Chip
IRQ10	VXI Interface Chip
IRQ11	SCSI
IRQ12	Mouse
IRQ13	Numeric Co-processor
IRQ14	IDE
IRQ15	Ethernet
NMI	EXMbus I/O Channel Check Asserted

Table B-2. IRQs (Interrupt Request Lines) (continued)

Interrupt	Source
SMI	Power Management
INTA	VGA (not mapped)
INTB	Ethernet (mapped to IRQ15)
INTC	SCSI (mapped to IRQ11)
INTD	VXI Interface Chip (mapped to IRQ9)

a. See Appendix D for information on freeing up IRQ3 for a second HP-22 GPIB card on an HP E6232 VXI Pentium Controller running Windows 95.

b. On the HP E6233 VXI Pentium Controller running Windows NT, IRQ5 is available for a second HP-22 GPIB card. Although IRQ5 is the same IRQ used by the printer port (LPT1), it does not conflict during actual use.

Appendix C
Specifications

This appendix lists the environmental and electrical specifications for the VXI Pentium Controller. See the Declaration of Conformity before Chapter 1 for information on EMC and Safety standards compliance.

Environmental

The following table defines the environmental specifications for the I/O base board in the VXI Pentium Controller.

Table C-1. Environmental Specifications

Characteristic	Operating Value	Non-Operating (Storage) Value
Temperature	5°C to 45°C, derated 2°C per 1,000 feet (300 meters) over 10,000 feet (3,000 meters); 2°C per minute maximum excursion gradient	-40°C to +65°C, 5°C per minute maximum excursion gradient
Humidity	8% to 90% non-condensing	5% to 95% non-condensing
Altitude	0 to 10,000 feet (3,000 meters)	0 to 40,000 feet (12,000 meters)
Vibration	0.015 inch (0.38 mm) P-P displacement with 2.5 g peak (maximum) acceleration over 5 to 300 Hz	0.030 inch (0.76 mm) P-P displacement with 5.0 g peak (maximum) acceleration over 5 to 300 Hz
Shock	30 g, 11 ms duration, half-sine shock pulse	50 g, 11 ms duration, half-sine shock pulse

Electrical

The following table defines the electrical specifications for the controller's I/O base board.

Table C-2. VXI Pentium Controller Power Requirements[a]

Voltage	DC Current	Dynamic Current
+5 V dc	11 A	0.31 A
+12 V dc	40.2 mA	0.08 A
-12 V dc	2.4 mA	0.03 A
-2 V dc	120 mA	0.08 A
-5.2 V dc	335 mA	0.192 mA

a. These specifications are for the E6233A with a 166 MHz processor. The power requirements for the E6232A with a 133 MHz processor will be lower.

Appendix D

Freeing Up IRQ3 for a Second HP-22 Card

Note The information in this appendix applies only to the HP E6232 VXI Pentium Controller running Windows 95. This is *not* applicable to the HP E6233 controller running Windows NT.

On the HP E6233 running Windows NT, note that IRQ5 is available for a second HP-22 GPIB card. Although IRQ5 is the same IRQ used by the printer port (LPT1), it does not conflict during actual use.

When the HP E6232 VXI Pentium Controller is running Windows 95, there are no free interrupt request lines (IRQs) available for a second HP-22 GPIB card. Therefore, if you installed a second HP-22 GPIB card in your HP E6232 controller, you need to free up an IRQ for this second card. The following procedure explains how to free up IRQ3 (used by COM2 by default) for use by this second card.

1. From the **`Start`** button in Windows 95, select **`Settings`** and then **`Control Panel`**.
2. Double-click on the **`System`** icon in the **`Control Panel`** window.
3. Select the **`Device Manager`** tab in the **`System Properties`** dialog box.
4. Select the **`View devices by type`** button if it is not already selected.
5. Locate the **`Ports (COM & LPT)`** label in the scroll box and click on the small box to the left of this label to expand the list of ports. (When the list is expanded, the "**`+`**" symbol in the box will change to a "**`-`**" symbol.)
6. Click on **`Communications Port (COM2)`** to highlight it.
7. Click on the **`Remove`** button below the scroll box to remove COM2 from the configuration.
8. Click on **`OK`** in the warning dialog box to verify that you really want to remove COM2 from the system. This will make IRQ3 available for use by the second HP-22 card.
9. Close the **`System Properties`** dialog box.

10. Run the **`I/O Config`** utility by double-clicking on the icon in the **`HP I/O Libraries`** program group. Now configure the second HP-22 GPIB card in **`I/O Config`**. Note that IRQ3 is now available for use by this second HP-22 GPIB card.

Note You must configure the second HP-22 GPIB card via **`I/O Config`** after removing COM2 but *before* rebooting Windows 95. If you reboot Windows 95 before configuring the second HP-22 in **`I/O Config`**, Windows 95 will find and re-add COM2.

Also note that even after the second HP-22 is configured, Windows 95 will find COM2, but it will not override the IRQ3 assignment for the second HP-22 once it is configured via **`I/O Config`**.

Appendix E
The DEVICES File

This appendix explains how you can customize the VXI Pentium Controller by manually editing the **DEVICES** file directly to create record entries, rather than by using the VXI Device Configurator. Therefore, this appendix explains the **DEVICES** file records and provides sample **DEVICES** file entries.

If you would like to use the interactive VXI Device Configurator program to edit the **DEVICES** file (rather than editing the file manually, as explained in this appendix), please see Chapter 5, “Using the VXI Device Configurator.”

The **DEVICES** file is key to the SURM environment. It defines the names and other parameters of VXI devices that can be accessed. Devices are defined by a series of records that consist of variables and value parameters. It is not required that an instrument be physically present in the system when you define it. This is handy when you often change the system configuration.

A **DEVICES** file record has this form:

```
<name-assignment>[,<variable-assignment>]*<newline>
```

Where:

```
<name-assignment>          :   "name" "=" <devicename>

<variable-assignment>      :   <variable> ["=" <value>]

<variable>                 :   <string>

<value>                    :   <string>
```

Creating Records

The installation process places a **DEVICES** file in the **C:\SICL95\VXI** or **C:\SICLNT\VXI** directory; however, the file contains no records for the system devices. You can create the required **DEVICES** file records either by using the VXI Device Configurator, or by manually editing the **DEVICES** file.

The recommended method is using the VXI Device Configurator, which is an interactive Windows application that automatically uses the required file syntax for creating **DEVICES** file record entries.

However, if you choose to manually edit the **DEVICES** file instead, you must observe these syntax rules:

- New-line characters delimit records. Multi-line records require a backslash character at the line's end for continuation onto another line.

- Lines beginning with **#** are comments and are ignored.

- A line cannot contain more than 80 characters. Use **** to break up a definition line longer than 80 characters.

- Quote a single character by placing a backslash (****) in front of it (for example, **\=** or **\,**). Quote an entire string by enclosing the string with double quotation characters (**"***string***"**). Quoting a string is necessary to ensure leading and trailing spaces are not removed. Character or string quoting is also necessary to embed an equal sign character or comma character.

- Redundant variable name handling is application-dependent. The SURM and the SICL **iopen** function always use first assignment of a given variable name and ignore any others.

- A string is a series of any characters except a non-quoted comma or equal sign. Non-quoted whitespace characters will be removed from the beginning and end of strings.

- Valid variables depend on the device type. Incorrectly named variables are ignored. The form of the value assigned depends on the variable.

- Variable name matching is not case-sensitive. If the value is an enumerated type (for example, VME, VXI, GPIB), value matching is also not case-sensitive.

Variables

DEVICES file variables are not case-sensitive and include the following.

Table E-1. DEVICES File Variable Definitions

Variable	Description
name	Required variable and must: • Be 1 to 12 characters in length. • Limit characters to letters, digits, and underscores. • Begin with a letter and not end with an underscore.
media	Required variable. Valid values are: **VXI**, **GPIB**, **VME**, **NETLINK**, and **CUSTOM**. This variable's value affects the interpretation of other variables and determines how the SURM and the treat the device.
ifname	Optional variable that associates a device driver to an interface (via the Windows registry) other than a pre-defined interface. Cannot be set with the **I/O Config** utility and only used by SICL.
make *model*	Optional variables that specify the device's make and model. Used for name binding by the SURM and for reference by user applications. For VXI devices, *make* and *model* must match exactly one of the make or model names defined in the files **C:\SICL*XX*\VXI\DB\VXIMANUF** and **C:\SICL*XX*\VXI\DB\VXIMODEL**, respectively.
slot	Optional variable that defines the VXI slot number. Valid range is **0** to **12**. Specify in decimal. For name binding purposes only.
bus	Optional variable that defines the VXIbus in which the device resides. Valid range is **0** to **255**. Specify in decimal. For VXI devices, this variable is for name binding purposes only. For VME devices, this variable is used for mapping the device's memory across the appropriate mainframe extenders.
device	Optional variable for name binding a VXI device when multiple logical devices exist in a single VXI module. This variable provides a way to distinguish them by name. Specify in decimal. The first device in a module is number **0**.
ula	Optional variable whose value is the device's statically configured logical address used for name binding a VXI device. Specify in decimal. Using this variable is not the recommended method to name dynamically configured devices, because these ULAs are not predictable. All devices are given ULAs regardless of whether they have configuration registers in A16 or not. For non-VXI devices, *ula* is assigned based on this variable if it is defined. Otherwise, the SURM selects a value.
commander	Optional variable whose value is the device name of the commander. Any device may have a commander variable defined. If the device and its commander are both VXI, the SURM makes the device a VXI servant of the named commander. For other media types, the relationship is stored for reference only.

Table E-1. DEVICES File Variable Definitions (continued)

Variable	Description
handlermap *interruptermap*	Optional variable that defines interrupt maps for any VXI, VME, or CUSTOM device. The *handlermap* variable selects the IRQ line(s) on which a device is to handle interrupts. If the device is VXI and has programmable handler (PH) capability, it will be programmed by the SURM to handle the interrupts specified here. The *interruptermap* selects the IRQ line on which the device generates interrupts. If the device is VXI and has programmable interrupter (PI) capability, it will be programmed by the SURM to interrupt as specified here. IRQ lines specified by *handlermap* are not assigned by the SURM to any other PH-capable VXI device. Maps for VXI devices not found in the system are ignored (no IRQs reserved). VME and CUSTOM devices are always assumed to be present. The value of these variables is a string of 7 digits. The rightmost digit corresponds to interrupter or handler number 1 on the device (this is for devices with multiple handlers or interrupters). The digits are **`1`** through **`7`**, and indicate the IRQ line to which the interrupter or handler is connected. **`0`** (zero) indicates the interrupter or handler is not connected to an IRQ. Unspecified digits are assumed to be zero. The SURM automatically configures PI/PH-capable devices with no maps specified. Once an interrupt number appears in the *handlermap* of a device that exists, that interrupt assignment is used and no other devices in the system can handle that interrupt. (*handlermap* for VXI devices that are configured but not present do not have this effect).
a16base *a16size* *a24base* *a24size* *a32base* *a32size*	Optional variables for VME and CUSTOM devices. Defines which section of the address space to assign to the device. Values are in hexadecimal. If either **size* or **base* is missing, it is assumed that the address space is not used by the device.
byteorder	Optional variable for VME and CUSTOM devices. Valid entries are: **`I`** for little-endian (Intel), or **`M`** for big-endian (Motorola).
datawidth	Optional variable for VME and CUSTOM devices. Valid entries are: **`D08`** (8-bit), **`D08O`** (8-bit odd address only), **`D16`** (16-bit), or **`D32`** (32-bit).
bridge *mdsparams*	Optional variables for CUSTOM devices (both must appear). *bridge* is the mnemonic for the user-supplied bridge, and *mdsparams* is the string of initialization parameters for the device. Devices on other media will be registered with the appropriate *bridge* by the SURM with an equivalent of *mdsparams* derived by the SURM.

Table E-1. DEVICES File Variable Definitions (continued)

Variable	Description
primary *secondary*	Required entry for GPIB devices that forms the device's address on the GPIB bus. *primary* is a required parameter, *secondary* is not required and, if not entered, defaults to **`0`** (zero).
sendeoioneos *sendeoiwithlast* *terminatereadoneos*	Optional variables for GPIB devices. No value is required. When specified, the action defined by that variable takes place.
eosbitscompared	Optional variable for GPIB devices that controls the number of bits that identify the EOS character in messages. Valid values are **`7`** or **`8`**.
eoschar	Required variable for GPIB devices that specifies the hexadecimal value of the EOS character.
timeout	Required variable for GPIB and NETLINK devices. For NETLINK devices, this is the decimal value of the network timeout for the device in half seconds. For GPIB devices, it is the GPIB timeout value for the device. For GPIB devices, valid values are: **`none`**, **`10us`**, **`30us`**,**`100us`**, **`300us`**, **`1ms`**, **`3ms`**, **`10ms`**, **`30ms`**, **`100ms`**, **`300ms`**, **`1s`**, **`3s`**, **`10s`**, **`30s`**, **`100s`**, **`300s`**, and **`1000s`**.

Example DEVICES File Records

The following are example records in the **DEVICES** file.

```
name=vxidev, \
 media= VXI, \
 make= Tektronix, \
 model= VX4236

name=GPIBdev, \
 media=GPIB, \
 make=HPAF6, \
 model=E1445A, \
 primary=1, \
 eoschar=0d, \
 terminatereadoneos, \
 sendeoiwitheos, \
 sendeoiwithlast, \
 eosbitcompare=8, \
 timeout = “300 ms”

#  In this example TopCmdr handles all interrupts (0-7).
#  This example also enables TopCmdr to assert interrupts
#  on IRQ2.  Unlike handler assignment, other devices can
#  specify duplicate interrupt map IRQs.

name=TopCmdr, \
 media=VXI, \
 ula=0, \
 handlermap=7654321

#  This is a minimal VME DEVICES record for a device
#  occupying 64 KB in A24 address space:

name=vmedev, \
 media=VME, \
 make=Motorola, \
 model=MVME121, \
 a24base=8000, \
 a24size=10000, \
 datawidth=D16, \
 byteorder=M

#  This is a custom DEVICES record where interactions
#  with the device are handled by a user-supplied bridge:

name=custdev, \
 media=CUSTOM, \
 mdsparms=”init string”, \
 bridge=USER
```

Appendix F
Re-installing the Software

The VXI Pentium Controller comes pre-loaded with either the Microsoft Windows 95 or Windows NT operating system, as well as the for Windows software. This appendix explains what process you should follow if you ever need to re-install the operating system and/or the software on the controller for any reason.

To Re-install the Windows Operating System

If you need to re-install the Windows 95 or Windows NT operating system software on your VXI Pentium Controller, follow all the installation instructions provided in the Microsoft Windows installation/user's guide that was shipped with your controller.

To complete the re-installation, you will need to use the Windows 95 or Windows NT installation CD-ROM, which was also shipped with your controller. To use this CD-ROM, you will need to connect a compatible CD-ROM drive (such as the HP C2944) to the controller, if one is not already connected.

Once you have finished re-installing the Windows operating system on your controller, you will need to re-install a particular driver needed for your controller. This is because the Windows 95 and Windows NT version 4.0 installation CD-ROMs shipped with the HP E623x VXI Pentium Controller do not contain all of the drivers needed for the controller hardware. In particular, the Windows 95 CD-ROM does not contain the required video driver for the HP E6232, and the Windows NT version 4.0 CD-ROM does not contain the required network driver for the HP E6233. To overcome this problem, HP has placed the necessary drivers in the **`drivers`** directory of the for Windows CD-ROM. These drivers are supplied by the chip vendors and are distributed with their permission.

Note Each of these drivers is pre-installed on the VXI Pentium Controller when you purchase it from HP. It is only necessary to re-install the driver on your controller if you are re-installing *both* the Windows operating system and the for Windows software on your controller from scratch.

To install these drivers, you must connect a CD-ROM drive to your controller, if one is not already connected, and place the for Windows CD-ROM in this drive. For the procedures in this appendix, the CD-ROM drive is assumed to be drive **`D:`**. If this is not the case, substitute the appropriate drive letter for your CD-ROM drive when it is required in the following procedures.

Please go to the following, appropriate subsection in this appendix and follow the procedures for re-installing the driver needed for your particular controller now. Once you have finished re-installing the appropriate driver on your controller, go on to the next main section, “To Re-install the for Windows,” at the end of this appendix.

Re-installing the Video Driver on the HP E6232

After you have re-installed the Windows 95 operating system software on your HP E6232 controller, insert the for Windows CD-ROM in the CD-ROM drive, and do the following to re-install the video driver on your controller.

1. Right-click on the Windows 95 desktop.
2. Select **`Properties`**.
3. Select the **`Settings`** tab.
4. Click on the **`Change Display Type`** button.
5. Click on the **`Change`** button in the **`Adapter Type`** box.
6. Click on the **`Have Disk`** button.
7. In the text box labeled **`Copy manufacturer’s files from:`**, enter the path:

   `D:\DRIVERS\WIN95\CIRRUS`

   where **`D:`** is the drive letter for your CD-ROM drive. After entering the correct path, click on **`OK`**.
8. The name of the driver will appear in the next dialog box, as follows:

   `Cirrus Logic 5436/5446 PCI DirectDraw/VPM (v.1.12)`

   Click on **`OK`** in this dialog box to copy the driver files and install the driver on your controller.
9. At this point, you can select your monitor type or close the display properties dialog boxes.

Re-installing the Network Driver on the HP E6233

After you have re-installed the Windows NT operating system software on your HP E6233 controller, insert the for Windows CD-ROM in the CD-ROM drive, and do the following to re-install the network driver on your controller.

1. From the Windows NT version 4.0 Start button, go to **`Settings`** and then open **`Control Panel`**.
2. Double-click the Network icon and select the Adapters tab.
3. Click on the **`Add`** button.

4. Click on the **`Have Disk...`** button.

5. In the text box on the Insert Disk dialog box, enter the path:

   `D:\DRIVERS\WINNT\AMDENET\WINNT`

   where **`D:`** is the drive letter for your CD-ROM drive. After entering the correct path, click on **`OK`**.

6. The name of the driver will appear in the next dialog box, as follows:

   `AMD PCNET Family Ethernet Adapter`

   Click on **`OK`** in this dialog box.

7. The setup dialog box for the network adapter appears. Select the following options in this dialog box:

`I/O Port:`	**`0x0300`**
`IRQ Number:`	**`15`**
`DMA Number:`	**`5`**
`Full Duplex:`	**`OFF`**
`Bus Timer:`	**`Default`**
`LED 0:`	**`Default`**
`LED 1:`	**`Default`**
`LED 2:`	**`Default`**
`LED 3:`	**`Default`**
`TP Box:`	**`Checked`**
`Bus to Scan:`	**`PCI`**

   Once you have set the options correctly, as listed above, click on the **`OK`** button.

8. In addition to adding the adapter, be sure the other networking information, such as Identification, Services, and so forth, is set correctly. You can then close the Network dialog box.

To Re-install the HP I/O Libraries for Windows

Note The for Windows drivers and utilities for the VXI Pentium Controller are included on the for Windows installation CD-ROM. These drivers and utilities will be installed along with the for Windows software – you do *not* need to follow any separate installation procedures to re-install them.

However, if you are re-installing *both* the Windows operating system and the for Windows software on your controller, note that the Windows 95 video driver for the HP E6232 and the Windows NT network driver for the HP E6233 must be re-installed off the for Windows CD-ROM using a separate installation procedure. This should be done *before* re-installing the for Windows software on the VXI Pentium Controller. Please see the previous section in this appendix, “To Re-install the Windows Operating System,” for the procedures to follow.

If you need to re-install the for Windows software on your VXI Pentium Controller, follow all the installation and configuration instructions in Chapter 2, “Installing and Configuring the,” of the *HP I/O Libraries Installation and Configuration Guide for Windows*. This manual was shipped with your controller.

To complete the re-installation, you will need to use the for Windows installation CD-ROM, which was also shipped with your controller. To use this CD-ROM, you will need to connect a compatible CD-ROM drive (such as the HP C2944) to the controller, if one is not already connected.

Once you have finished re-installing and then re-configuring (via the **`I/O Config`** utility) the for Windows software on your controller, you may need to further customize your controller, depending on your controller’s particular hardware setup. Please see the section, “Customizing the Controller,” toward the end of Chapter 2 in this manual for more information.

Appendix G

Installing BIOS Software

This appendix explains how to reflash the system BIOS in the HP VXI Pentium Controller from the controller's floppy disk drive. This process is rarely done *only* when the BIOS stored in the controller's flash memory has been damaged or when HP releases a new version of the BIOS to correct a specific problem or to add support for new system features.

Note The special BIOS boot diskette referred to in the instructions below is *not* shipped with your HP VXI Pentium Controller. If required, it must be obtained from an HP Support Representative.

Caution **It is strongly recommended that a trained computer service technician performs the installation and removal of the jumpers referred to in the instructions below. An HP Service Technician can perform this task for you. Contact your HP Service and Support Center for assistance.**

Overview

The HP VXI Pentium Controller's System BIOS is updated by using a special boot diskette that contains the System BIOS image as well as the code to perform the update.

To enable the recovery process and to allow writing to the flash memory which contains the BIOS, two jumpers must be installed on jumper-pin connector blocks located inside the controller. To access the jumper-pin connector blocks, the controller must be powered down and removed from the VXI mainframe in order to remove the side cover panel.

Once the jumpers have been installed and the controller returned to the VXI mainframe, the system must be booted with the special BIOS boot diskette loaded in the floppy disk drive. The recovery process will then automatically update the BIOS.

Once the recovery process is complete, the controller must be powered down and the jumpers removed before you can return to normal operation.

Record CMOS/BIOS Settings

Prior to reflashing the system BIOS in the HP VXI Pentium Controller, you should record all CMOS/BIOS system settings.

1. Apply power to the VXI mainframe. During system boot-up, press F2.

2. Record all settings.

3. Power-down the VXI mainframe.

Jumper Installation Procedure

Caution **The circuitry in the HP VXI Pentium Controller can be damaged by electrostatic discharge. All work must be done by a trained technician at a static-free workstation.**

Tools Required

You will need the following tools to access the controller's main circuit board and install the pin jumpers:

- two, 2-position, 2.5 mm, removable jumpers; Optionally, you may need a third, jumper.
- a grounding wrist strap and static-free workstation; and
- a #0 (small) Phillips screwdriver.

Disassembly Steps

1. At a static-free workstation, place the HP VXI Pentium Controller on its left side such that the front panel is facing you and the sheet-metal cover with its bar-coded serial number is on the far edge away from you.

2. Attach the grounding strap to your wrist and take appropriate anti-static precautions during the rest of this procedure.

3. With the #0 Phillips screwdriver, remove the eight Phillips-head screws holding the sheet-metal cover to the chassis. Do not lose these very small screws. They will be needed to re-attach the cover.

4. Remove the sheet-metal cover and set it aside.

   The two jumperconnector blocks, labelled P7 and P8, are located just to the left of the power cable connector (see the multi-colored wire harness) on the main I/O base board near the rear-center of the unit.

5. Place a jumper to connect pin 7 to pin 8 on jumper block P7 and a jumper to connect pin 1 to pin 3 on jumper block P8 as shown below:

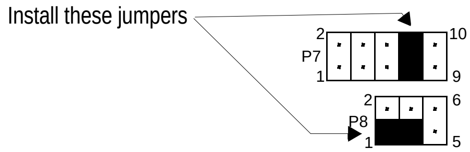

Figure G-1. Jumper-Pin Connector Blocks

6. Replace the sheet-metal cover such that the bent edges on the sides are down and the bar-coded serial number label is on the edge nearest the VXI connector blocks on the rear of the unit.

7. Using the #0 Phillips screwdriver, replace the eight Phillips-head screws holding the sheet-metal cover to the chassis. Gently tighten the screws being careful not to strip the threads.

8. Return the controller to its slot in the VXI mainframe.

Booting the Special BIOS Boot Diskette

Perform the following procedure to boot the special BIOS boot diskette and write the system BIOS into the controller's flash memory:

Note The video monitor is not operative during the recovery process. The progress of the recovery is communicated to the operator via audible beeps from the controller's internal speaker.

---

1. With the VXI mainframe powered down, insert the special BIOS boot diskette into the controller's floppy disk drive.

2. Power on the VXI mainframe. The controller will start booting from its floppy disk drive.

   The boot program will issue three beeps when the actual recovery process begins. The first series of beeps from the recovery program denotes the reading of the System BIOS file. The last series of beeps denotes the erasing of the flash memory and reloading of the BIOS code. A single long beep denotes the successful recovery of the System BIOS and that the system is about to reset itself.

Note If the system will not boot from the floppy diskette and the system has SO DIMM installed in only the right memory bank (bank 1), move the memory from bank 1 to bank 0 (left bank) and retry this procedure. See Appendix E, “Installing Additional DRAM Memory”, for instructions on how to move the memory.

Alternate Installation Note If after 45 to 50 seconds, the single long beep has not occurred; but instead, you repeatedly hear a short beep followed by the muted chatter of the disk drive (it may sound like a muffled buzz), turn off the VXI mainframe. Refer to “**Alternate Installation**,” later in this section.

3. When the single long beep has been issued and the system reset begins, you must immediately power down the VXI mainframe. Otherwise, the boot from the special BIOS boot diskette will start again and the recovery process will repeat.

4. Remove the BIOS boot diskette from the controller’s disk drive.

5. Remove the HP VXI Pentium Controller from the VXI mainframe and take it to a static-free workstation in order to remove the jumpers you installed earlier.

Note All jumpers installed on P7 and P8 must be removed to prevent later accidental writing of the System BIOS.

6. Return to the section above titled **Disassembly Procedure** and follow the instructions to remove the cover panel. But, this time, remove the jumpers from jumper blocks P7 and P8.

7. Connect the video monitor, keyboard, and mouse to the HP Pentium Controller.

8. Apply power to the VXI mainframe.

9. Press **F2** during the normal boot of the VXI system.

10. Select the EXIT menu, then select Get Default Values. Press the **ENTER** key. This resets everything in the BIOS.

11. Reset all the CMOS/BIOS system settings to those recorded earlier. See **Record CMOS/BIOS Settings**.

12. Press the **F2** function key during the system boot process. This accesses the BIOS Setup Utility.

13. Use the left/right arrow keys to select the MAIN menu. Use the keyboard up/down arrow keys to select: IDE Adapter 0 Master. Press the **ENTER** key to select the sub menu. Press the **ENTER** key to auto-detect the fixed disk on the controller. Press the **ESC** key.

14. Select the ADVANCED menu. Use the up/down arrow keys to select: Advanced Chipset Control. Press the **ENTER** key to select the sub menu. Use the up/down arrow keys to select: DRAM Speed. Use the +/- keys to select: `60ns`. Press the **ESC** key.

    Use the up/down arrow keys to select: Reset Configuration Data. Use the +/- keys to select: YES. Press the **ESC** key.

15. Select the EXM menu. Use the up/down arrow keys to select: EXM Slot 1 ID. Use the +/- keys to select: `D9`. Select Option Byte 1 and set to FF. Leave Option Byte 2 set to `00`. Press the **ESC** key.

16. Select the EXIT menu. Select Save Changes & Exit. This saves the values into the CMOS memory and then reboots the system using the new values.

For additional information about any of these menu items, refer to Chapter 3.

This completes the reflashing of the HP VXI Pentium Controller BIOS. The system is now ready for normal operation.

Alternate Installation

Follow this installation procedure *only* if the normal BIOS boot procedure resulted in an alternating beep and the muted chatter of the disk drive.

1. Remove the HP VXI Pentium Controller module from the VXI mainframe.

2. Following the **Disassembly Steps** earlier in this section, remove the sheet metal cover.

3. Install a third jumper between pins 2 and 6 on jumper block P8. Refer to Figure D-2.

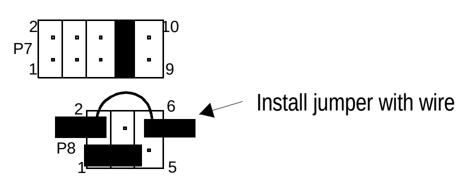

Figure G-2. Jumper-Pin Connector Blocks for Alternate Installation

4. Reassemble the sheet metal cover.

5. Return the controller to its slot in the VXI mainframe.

6. Follow the procedures in **Booting the Special BIOS Boot Diskette**.

Index