COMPAQ

AlphaServer ES40

Service Guide

Order Number: EK–ES240–SV. A01

This guide is intended for service providers and self-maintenance customers responsible for *Compaq AlphaServer* ES40 systems.

Compaq Computer Corporation

First Printing, July 1999

Warning! This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

Achtung! Dieses ist ein Gerät der Funkstörgrenzwertklasse A. In Wohnbereichen können bei Betrieb dieses Gerätes Rundfunkstörungen auftreten, in welchen Fällen der Benutzer für entsprechende Gegenmaßnahmen verantwortlich ist.

Attention! Ceci est un produit de Classe A. Dans un environnement domestique, ce produit risque de créer des interférences radioélectriques, il appartiendra alors à l'utilisateur de prendre les mesures spécifiques appropriées.

警告使用者:

這是甲類的資訊產品，在居住的環境中使用時，可能會造成射頻干擾，在這種情況下，使用者會被要求採取某些適當的對策。

に基づくクラスＡ情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

FCC Notice: This equipment generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules, which are designed to provide reasonable protection against such radio frequency interference.

Operation of this equipment in a residential area may cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

Any modifications to this device—unless expressly approved by the manufacturer—can void the user's authority to operate this equipment under part 15 of the FCC rules.

Contents

Preface xvii

Chapter 1 System Overview

1.1 System Architecture 1-2
1.2 System Enclosures 1-4
1.3 System Chassis—Front View/Top View 1-6
1.4 System Chassis—Rear View 1-7
1.5 I/O Ports and Slots 1-8
1.5 Control Panel 1-10
1.6 System Motherboard 1-12
1.7 CPU Card 1-14
1.8 Memory Architecture and Options 1-16
1.9 PCI Backplane 1-18
1.10 Remote System Management Logic 1-20
1.10.1 System Power Controller (SPC) 1-22
1.10.2 Remote Management Console (RMC) 1-23
1.11 Power Supplies 1-24
1.12 Fans 1-26
1.13 Removable Media Storage 1-28
1.14 Hard Disk Drive Storage 1-29
1.15 System Access 1-30
1.16 Console Terminal 1-32

Chapter 2 Troubleshooting

2.1 Questions to Consider 2-2
2.2 Diagnostic Tables 2-3
2.3 Service Tools and Utilities 2-9
2.3.1 Error Handling/Logging Tools (Compaq Analyze) 2-9
2.3.2 Loopback Tests 2-9
2.3.3 SRM Console Commands 2-9

2.3.4 AlphaBIOS Menus 2-10
2.3.5 Remote Management Console (RMC) 2-10
2.3.6 Operating System Exercisers (DEC VET) 2-11
2.3.7 Crash Dumps 2-11
2.3.8 Revision and Configuration Management Tool (RCM) 2-11
2.3.9 StorageWorks Command Console (SWCC) 2-12
2.4 Information Resources 2-13
2.4.1 Compaq Service Tools CD 2-13
2.4.2 AlphaServer ES40 Service HTML Help File 2-13
2.4.3 Alpha Systems Firmware Updates 2-13
2.4.4 Fail-Safe Loader 2-14
2.4.5 Software Patches 2-14
2.4.6 Late-Breaking Technical Information 2-15
2.4.7 Supported Options 2-15

Chapter 3 Power-Up Diagnostics and Display

3.1 Overview of Power-Up Diagnostics 3-2
3.2 System Power-Up Sequence 3-3
3.3 Power-Up Displays 3-6
3.3.1 SROM Power-Up Display 3-8
3.3.2 SRM Console Power-Up Display 3-10
3.3.3 Resizing SRM Console Heap 3-14
3.3.4 SRM Console Event Log 3-19
3.3.5 AlphaBIOS Startup Screens 3-20
3.4 Power-Up Error Messages 3-22
3.4.1 SROM Messages with Beep Codes 3-22
3.4.2 Checksum Error 3-24
3.4.3 No MEM Error 3-26
3.4.4 RMC Error Messages 3-28
3.4.5 SROM Error Messages 3-30
3.5 Forcing a Fail-Safe Floppy Load 3-32
3.6 Updating the RMC 3-34

Chapter 4 SRM Console Diagnostics

4.1 Diagnostic Command Summary 4-2
4.2 buildfru 4-4
4.3 cat el and more el 4-8
4.4 clear_error 4-10
4.5 crash 4-11
4.6 deposit and examine 4-12

4.7 exer ... 4-16
4.8 floppy_write ... 4-21
4.9 grep ... 4-22
4.10 hd ... 4-24
4.11 info ... 4-26
4.12 kill and kill_diags ... 4-31
4.13 memexer ... 4-32
4.14 memtest ... 4-34
4.15 net ... 4-39
4.16 nettest ... 4-41
4.17 set sys_serial_num ... 4-45
4.18 show error ... 4-46
4.19 show fru ... 4-49
4.20 show_status ... 4-52
4.21 sys_exer ... 4-54
4.22 test ... 4-56

Chapter 5 Error Logs

5.1 Error Log Analysis with Compaq Analyze ... 5-2
5.1.1 WEB Enterprise Service (WEBES) Director ... 5-3
5.1.2 Invoking the GUI ... 5-4
5.1.3 Problem Found Report ... 5-6
5.2 Fault Detection and Reporting ... 5-12
5.3 Machine Checks/Interrupts ... 5-14
5.1.1 Error Logging and Event Log Entry Format ... 5-16
5.4 Environmental Errors Captured by SRM ... 5-18
5.5 Windows NT Error Logs ... 5-20
5.5.1 Viewing a Formatted Text-Style Error Frame ... 5-24
5.5.2 Viewing a Binary Dump of the Error Frame ... 5-26
5.5.3 Saving the Error Frame to the Floppy ... 5-27
5.5.4 Deleting an Error Frame ... 5-30

Chapter 6 System Configuration and Setup

6.1 System Consoles ... 6-2
6.1.1 Switching Between Consoles ... 6-4
6.1.2 Selecting the Console and Display Device ... 6-5
6.1.3 Setting the Control Panel Message ... 6-7
6.2 Displaying the Hardware Configuration ... 6-8
6.3 Setting Environment Variables for Tru64 UNIX or OpenVMS ... 6-10
6.4 Setting Up a System for Windows NT ... 6-20

6.4.1 Setting the Date and Time 6-21
6.4.2 Setting Up the Hard Disk 6-22
6.4.3 Setting the Level of Memory Testing 6-23
6.5 Setting Automatic Booting 6-24
6.5.1 Windows NT and Auto Start 6-25
6.5.2 Setting Tru64 UNIX or OpenVMS Systems to Auto Start 6-26
6.6 Changing the Default Boot Device 6-27
6.7 Running AlphaBIOS-Based Utilities 6-28
6.7.1 Running Utilities from a VGA Monitor 6-29
6.7.2 Setting Up Serial Mode 6-31
6.7.3 Running Utilities from a Serial Terminal 6-32
6.7.4 Running the RAID Standalone Configuration Utility 6-34
6.8 Setting SRM Security 6-35
6.9 Setting Windows NT Security 6-38
6.10 Configuring Devices 6-40
6.10.1 CPU Configuration 6-40
6.10.2 Memory Configuration 6-42
6.10.3 PCI Configuration 6-46
6.10.4 Power Supply Configurations 6-48
6.11 Switching Between Operating Systems 6-50
6.11.1 Switching from UNIX or OpenVMS to Windows NT 6-50
6.11.2 Switching from Windows NT to UNIX or OpenVMS 6-52

Chapter 7 Using the Remote Management Console

7.1 RMC Overview 7-2
7.2 Operating Modes 7-4
7.2.1 Bypass Modes 7-6
7.3 Terminal Setup 7-9
7.4 Connecting to the RMC CLI 7-10
7.5 SRM Environment Variables for COM1 7-12
7.6 RMC Command-Line Interface 7-13
7.6.1 Defining the COM1 Data Flow 7-15
7.6.2 Displaying the System Status 7-16
7.6.3 Displaying the System Environment 7-18
7.6.4 Dumping DPR Data 7-20
7.6.5 Power On and Off, Reset, and Halt 7-22
7.6.6 Configuring Remote Dial-In 7-24
7.6.7 Configuring Dial-Out Alert 7-26
7.6.8 Resetting the Escape Sequence 7-29
7.7 Resetting the RMC to Factory Defaults 7-30
7.8 Troubleshooting Tips 7-32

Chapter 8 FRU Removal and Replacement

8.1 FRUs 8-2
8.1.1 Power Cords 8-5
8.1.2 FRU Locations 8-6
8.1.3 Important Information Before Replacing FRUs 8-8
8.2 Removing Enclosure Panels on a Tower or Pedestal 8-10
8.3 Accessing the System Chassis in a Cabinet 8-14
8.4 Removing Covers from the System Chassis 8-16
8.5 Power Supply 8-20
8.6 Fans 8-22
8.7 Hard Disk Drives 8-24
8.8 CPUs 8-26
8.9 Memory DIMMs 8-28
8.10 PCI Cards 8-32
8.11 OCP Assembly 8-34
8.12 Removable Media 8-36
8.13 Floppy Drive 8-38
8.14 I/O Connector Assembly 8-40
8.15 PCI Backplane 8-42
8.16 System Motherboard 8-46
8.17 Power Harness 8-50

Appendix A SRM Console Commands

Appendix B Jumpers and Switches

B.1 RMC and SPC Jumpers on System Motherboard B-2
B.2 TIG/SROM Jumpers on System Motherboard B-4
B.3 Clock Generator Switch Settings B-6
B.4 Jumpers on PCI Board B-8
B.5 Setting Jumpers B-10

Appendix C DPR Address Layout

C.1 DPR Address Layout C-2

Appendix D Registers

D.1 Ibox Status Register (I_STAT) D-2
D.2 Memory Management Status Register (MM_STAT) D-4
D.3 Dcache Status Register (DC_STAT) D-6

D.4 Cbox Read Register D-8
D.5 Exception Address Register (EXC_ADDR) D-10
D.6 Interrupt Enable and Current Processor Mode Register (IER_CM).. D-12
D.7 Interrupt Summary Register (ISUM) D-14
D.8 PAL Base Register (PAL_BASE) D-16
D.9 Ibox Control Register (I_CTL) D-18
D.10 Process Context Register (PCTX) D-23
D.11 21272-CA Cchip Miscellaneous Register (MISC) D-26
D.12 21272-CA Cchip CPU Device Interrupt Request Register (DIRn, n=0,1,2,3) D-29
D.13 21272-CA Pchip Error Register (PERROR) D-31
D.14 21272-CA Array Address Registers (AAR0–AAR3) D-35
D.15 DPR Registers for 680 Correctable Machine Check Logout Frames . D-37
D.16 DPR Power Supply Status Registers D-40
D.17 DPR 680 Fatal Registers D-41
D.18 CPU and System Uncorrectable Machine Check Logout Frame D-42
D.19 Console Data Log Event Environmental Error Logout Frame (680 Uncorrectable) D-43
D.20 CPU and System Correctable Machine Check Logout Frame D-44
D.21 Environmental Error Logout Frame (680 Correctable) D-45
D.22 Platform Logout Frame Register Translation D-46

Appendix E Isolating Failing DIMMs

E.1 Information for Isolating Failures E-2
E.2 DIMM Isolation Procedure E-3
E.3 EV6 Single-Bit Errors E-16

Index

Examples

3–1 Sample SROM Power-Up Display 3-8
3–2 SRM Power-Up Display 3-10
3–3 Memory Resize Crash/Reboot Cycle 3-15
3–4 Sample Console Event Log 3-19
3–5 AlphaBIOS Initialization Screen 3-20
3–6 AlphaBIOS Boot Screen 3-21
3–7 Checksum Error and Fail-Safe Load 3-24
4–1 buildfru 4-4
4–2 more el 4-8

4–3	clear_error	4-10
4–4	deposit and examine	4-12
4–5	exer	4-16
4–6	floppy_write	4-21
4–7	grep	4-22
4–8	hd	4-24
4–9	info 0	4-26
4–10	info 1	4-27
4–11	info 2	4-28
4–12	info 3	4-29
4–13	info 4	4-30
4–14	kill and kill_diags	4-31
4–15	memexer	4-32
4–16	memtest	4-34
4–17	net -ic and net -s	4-39
4–18	nettest	4-41
4–19	set sys_serial_num	4-45
4–20	show error	4-46
4–21	show fru	4-49
4–22	show status	4-52
4–23	sys_exer	4-54
4–24	test -lb	4-56
5–1	Console Level Environmental Error Logout Frame	5-18
6–1	set ocp_text	6-7
6–2	set password	6-35
6–3	set secure	6-36
6–4	clear password	6-36
6–5	Advanced CMOS Setup Screen	6-38
7–1	set com1_mode	7-15
7–2	status	7-16
7–3	env	7-18
7–4	dump	7-20
7–5	power on/off	7-22
7–6	halt in/out	7-23
7–7	reset	7-23
7–8	Dial-In Configuration	7-24
7–9	Dial-Out Alert Configuration	7-26
7–10	set escape	7-29

Figures

1–1 System Block Diagram ... 1-2
1–2 Compaq AlphaServer ES40 Systems ... 1-4
1–3 Components Top/Front View (Pedestal/Rackmount Orientation) ... 1-6
1–4 Rear Components (Pedestal/Rackmount Orientation) ... 1-7
1–5 Rear Connectors ... 1-8
1–6 Control Panel ... 1-10
1–7 Component and Connector Locations ... 1-12
1–8 CPU Card ... 1-14
1–9 Memory Architecture ... 1-16
1–10 I/O Control Logic ... 1-18
1–11 Remote System Management Logic Diagram ... 1-20
1–12 Power Supplies ... 1-24
1–13 System Fans ... 1-26
1–14 Removable Media Drive Area ... 1-28
1–15 Hard Disk Storage Cage with Drives (Tower View) ... 1-29
1–16 System Lock and Key ... 1-30
1–17 Console Terminal Connections (Local) ... 1-32
3–1 Power-Up Sequence ... 3-4
3–2 Function Jumpers ... 3-32
5–1 Compaq Analyze GUI ... 5-4
5–2 Compaq Analyze Event Screen ... 5-5
5–3 Problem Found Report ... 5-6
5–4 FRU List Designator ... 5-8
5–5 Evidence Designator ... 5-10
5–6 New Error Frame Was Detected Window ... 5-20
5–7 Display Error Frames Screen ... 5-22
5–8 View by Formatted Text Style ... 5-24
5–9 Browsing Error Logs ... 5-25
5–10 Binary Dump of Error Frame ... 5-26
5–11 Save to the Floppy ... 5-27
5–12 Formatted Text File ... 5-28
5–13 Deleting a New Error Frame ... 5-30
5–14 Deleting an Old Error Frame ... 5-31
6–1 AlphaBIOS Setup Screen ... 6-2
6–2 Invoking SRM from AlphaBIOS ... 6-4
6–3 Display System Configuration Screen ... 6-9
6–4 CMOS Setup Screen ... 6-21
6–5 Hard Disk Setup Screen ... 6-22
6–6 Advanced CMOS Setup Screen ... 6-23

6–7 AlphaBIOS Utilities Menu 6-29
6–8 Run Maintenance Program Dialog Box 6-30
6–9 CPU Slot Locations (Pedestal/Rack) 6-40
6–10 CPU Slot Locations (Tower) 6-41
6–11 Stacked and Unstacked DIMMs 6-43
6–12 Memory Configuration (Pedestal/Rack) 6-44
6–13 Memory Configuration (Tower) 6-45
6–14 PCI Slot Locations (Pedestal/Rack) 6-46
6–15 PCI Slot Locations (Tower) 6-47
6–16 Power Supply Locations 6-48
7–1 Data Flow in Through Mode 7-4
7–2 Data Flow in Bypass Mode 7-6
7–3 Terminal Setup for RMC (Tower View) 7-9
7–4 RMC Jumpers (Default Positions) 7-30
8–1 FRUs — Front/Top (Pedestal/Rack View) 8-6
8–2 FRUs — Rear (Pedestal/Rack View) 8-7
8–3 Enclosure Panel Removal (Tower) 8-10
8–4 Enclosure Panel Removal (Pedestal) 8-12
8–5 Accessing the Chassis in a Cab 8-14
8–6 H9A10 Overhang Bezel 8-15
8–7 Covers on the System Chassis (Tower) 8-18
8–8 Covers on the System Chassis (Pedestal/Rack) 8-19
8–9 Removing a Power Supply 8-20
8–10 Replacing Fans 8-22
8–11 Removing a Hard Drive 8-24
8–12 Removing CPU Cards 8-26
8–13 Removing MMBs and DIMMs 8-28
8–14 Aligning DIMM in MMB 8-30
8–15 Installing or Replacing a PCI Card 8-32
8–16 Removing the OCP Assembly 8-34
8–17 Removing a 5.25-Inch Device 8-36
8–18 Removing the Floppy Drive 8-38
8–19 Removing the I/O Connector Assembly 8-40
8–20 Cables Connected to PCI Backplane 8-42
8–21 Removing the PCI Backplane 8-44
8–22 Removing the System Motherboard 8-46
8–23 Removing the Power Harness 8-50
B–1 RMC and SPC Jumpers B-2
B–2 TIG/SROM Jumpers B-4
B–3 CSB Switchpack E16 .... B-6
B–4 PCI Board Jumpers B-8

Tables

1 Compaq AlphaServer ES40 Documentation xix
1–1 Fan Descriptions 1-27
2–1 Power Problems.. 2-4
2–2 Problems Getting to Console Mode 2-5
2–3 Problems Reported by the Console.. 2-6
2–4 Boot Problems 2-7
2–5 Errors Reported by the Operating System.............................. 2-8
3–1 Error Beep Codes 3-22
3–2 RMC Fatal Error Messages.. 3-28
3–3 RMC Warning Messages 3-29
3–4 SROM Error Messages.. 3-30
4–1 Summary of Diagnostic and Related Commands................... 4-2
4–2 Show Error Message Translation 4-48
4–3 Bit Assignments for Error Field.. 4-51
5–1 Compaq AlphaServer ES40 Fault Detection and Correction............. 5-13
5–2 Machine Checks/Interrupts 5-14
5–3 Sample Error Log Event Structure Map (ES40 with 10 PCI Slots).... 5-17
6–1 SRM Environment Variables Used on ES40 Systems 6-12
6–2 AlphaBIOS Option Key Mapping... 6-32
7–1 Status Command Fields... 7-17
7–2 Elements of Dial String and Alert String 7-28
7–3 RMC Troubleshooting 7-32
8–1 FRU List.. 8-2
8–2 Country-Specific Power Cords 8-5
A–1 SRM Commands Used on ES40 Systems....................................A-1
B–1 RMC/SPC Jumper Settings..B-3
B–2 TIG/SROM Jumper Descriptions...B-5
B–3 Clock Generator Settings...B-7
B–4 PCI Board Jumper Descriptions B-9
C–1 DPR Address Layout...C-2
D–1 Ibox Status Register Fields D-3
D–2 Memory Management Status Register Fields D-5
D–3 Dcache Status Register Fields D-7
D–4 Cbox Read Register Fields D-8
D–5 IER_CM Register Fields D-13
D–6 ISUM Register Fields.. D-15
D–7 PAL_BASE Register Fields... D-17
D–8 I_CTL Register Fields D-19
D–9 PCTX Register Fields.. D-25
D–10 21272-CA Cchip Miscellaneous Register Fields.................... D-27

D–11 21272-CA Device Interrupt Request Register Fields D-30
D–12 21272-CA Pchip Error Register Fields D-33
D–13 21272-CA Array Address Register (AAR) D-35
D–14 DPR Locations A0:A9 .... D-37
D–15 Nine Bytes Read from Power Supply D-40
D–16 DPR 680 Fatal Registers D-41
D–17 CPU and System Uncorrectable Machine Check Logout Frame D-42
D–18 Console Data Log Event Environmental Error Logout Frame (680 Uncorrectable) D-43
D–19 CPU and System Correctable Machine Check Logout Frame D-44
D–20 Environmental Error Logout Frame D-45
D–21 Bit Definition of Logout Frame Registers D-47
E–1 Information Needed to Isolate Failing DIMMs E-2
E–2 Determining the Real Failed Array E-3
E–3 Description of DPR Locations 80, 82, 84, and 86 ....E-4
E–4 Failing DIMM Lookup Table...E-6
E–5 Syndrome to Data Check Bits Table E-16

Preface

Intended Audience

This manual is for service providers and self-maintenance customers who are responsible for servicing *Compaq AlphaServer* ES40 systems.

WARNING: To prevent injury, access is limited to persons who have appropriate technical training and experience. Such persons are expected to understand the hazards of working within this equipment and take measures to minimize danger to themselves or others. These measures include:
1. Remove any jewelry that may conduct electricity.
2. If accessing the system card cage, power down the system and wait 2 minutes to allow components to cool.
3. Wear an anti-static wrist strap when handling internal components.

Document Structure

This manual uses a structured documentation design. Topics are organized into small sections, usually consisting of two facing pages. Most topics begin with an abstract that provides an overview of the section, followed by an illustration or example. The facing page contains descriptions, procedures, and syntax definitions.

This manual has eight chapters and five appendixes.

- **Chapter 1, System Overview**, gives an overview of the system and describes the components.
- **Chapter 2, Troubleshooting**, describes the troubleshooting strategy, lists service tools, utilities, and information services, and gives diagnostic tables for problem categories.
- **Chapter 3, Power-Up Diagnostics and Display**, explains the power-up process and RMC, SROM, and SRM power-up diagnostics.
- **Chapter 4, SRM Console Diagnostics**, describes SRM console diagnostic commands.
- **Chapter 5, Error Logs,** describes error analysis with Compaq Analyze.
- **Chapter 6, System Configuration and Setup,** explains how to set up the system, configure devices, and ensure system security.
- **Chapter 7, Using the Remote Management Console,** explains the operation and use of the RMC.
- **Chapter 8**, **FRU Removal and Replacement**, gives procedures for removing and replacing FRUs.
- **Appendix A, SRM Console Commands**, lists the SRM commands used most frequently on ES40 systems.
- **Appendix B, Jumpers and Switches**, shows the jumpers and switches on the system motherboard and PCI backplane and explains their settings.
- **Appendix C, DPR Address Layout,** shows the address layout of the dual-port RAM (DPR).
- **Appendix D, Registers,** describes 21264 (EV6) internal processor registers; 21272 (Tsunami/Typhoon) system support chipset registers; and dual-port RAM (DPR) registers that are related to general logout frame errors. It also provides error state bit definitions of all the platform logout frame registers.
- **Appendix E, Isolating Failing DIMMs,** explains how to manually isolate a failing DIMM from the failing address and failing data bits. It also covers how to isolate single-bit errors.

Documentation Titles

1 Compaq AlphaServer ES40 Documentation

Title	Order Number
User Documentation Kit	**QA-6E88A-G8**
Owner's Guide	EK-ES240-UG
User Interface Guide	EK-ES240-UI
Basic Installation	EK-ES240-PD
Release Notes	EK-ES240-RN
Documentation CD (6 languages)	AG-RF9HA-BE
Maintenance Kit	**QZ-01BAB-GZ**
Service Guide	EK-ES240-SV
Service Guide HTML Help	AK-RFXDA-CA
Illustrated Parts Breakdown	EK-ES240-IP
Loose Piece Items	
Rackmount Installation Guide	EK-ES240-RG
Rackmount Installation Template	EK-ES4RM-TP
Model 1 to Model 2 Upgrade	EK-ES4M2-UP
ES40 DIMM Information Sheet	EK-MS610-DM

Information on the Internet

You can access service tools and more information about the ES40 from Compaq Web sites. See Chapter 2.

Chapter 1
System Overview

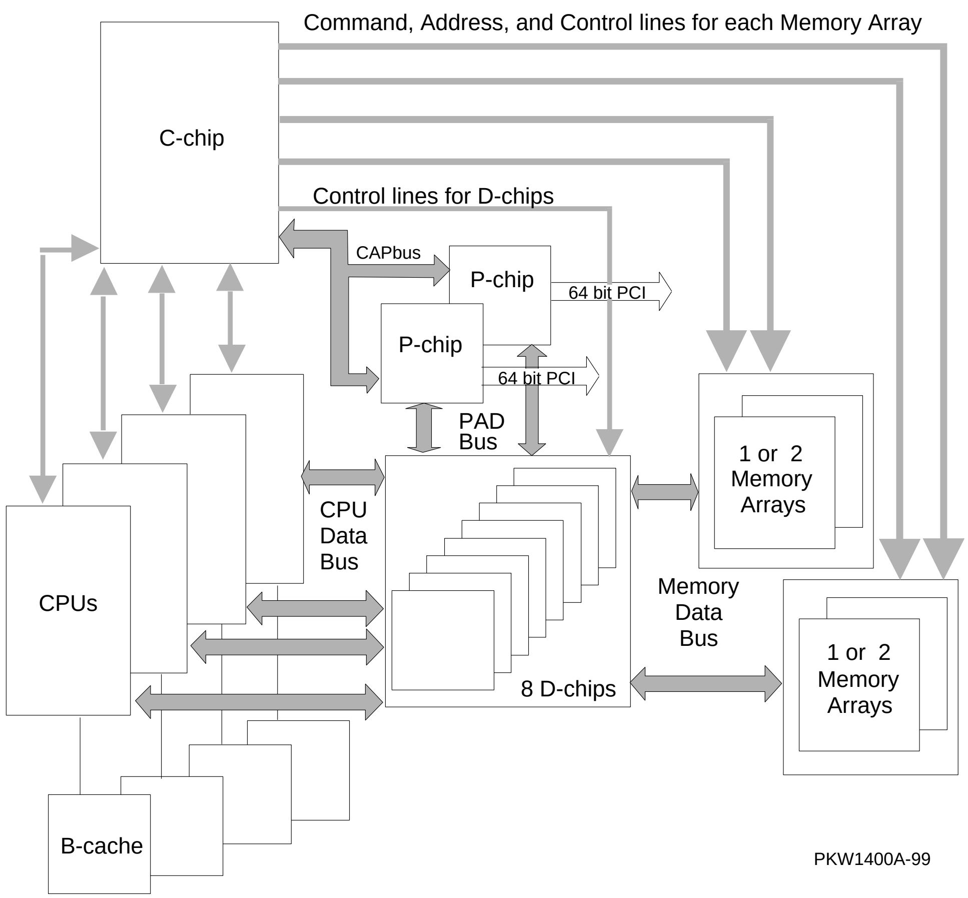
This chapter provides an overview of the system in these sections:

- System Architecture
- System Enclosures
- System Chassis—Front View/Top View
- System Chassis—Rear View
- I/O Ports and Slots
- Control Panel
- System Motherboard
- CPU Card
- Memory Architecture and Options
- PCI Backplane
- Remote System Management Logic
- Power Supplies
- Fans
- Removable Media Storage
- Hard Disk Drive Storage
- System Access
- Console Terminal

1.1 System Architecture

The system uses a switch-based interconnect system that maintains constant performance even as the number of transactions multiplies.

Figure 1–1 System Block Diagram



This system is designed to fully exploit the potential of the Alpha 21264 chip by using a switch-based (or point-to-point) interconnect system. With a traditional bus design, the processors, memory, and I/O modules share the bus. As the number of bus users increases, the transactions interfere with one another, increasing latency and decreasing aggregate bandwidth. With a switch-based system, speed is maintained and little degradation in performance occurs as the number of CPUs, memory, and I/O users increases.

The switched system interconnect uses a set of complex microprocessor 21272 support chips that route the traffic over multiple paths. This chipset consists of one C-chip, two P-chips, and eight D-chips.

- **C-chip.** Provides the command interface from the CPUs and main memory. The C-chip allows each CPU to do transactions simultaneously.
- **D-chips.** Provide the data path for the CPUs, main memory, and I/O.
- **P-chips.** Provide the interface to two independent 64-bit, 33 MHz PCI buses.

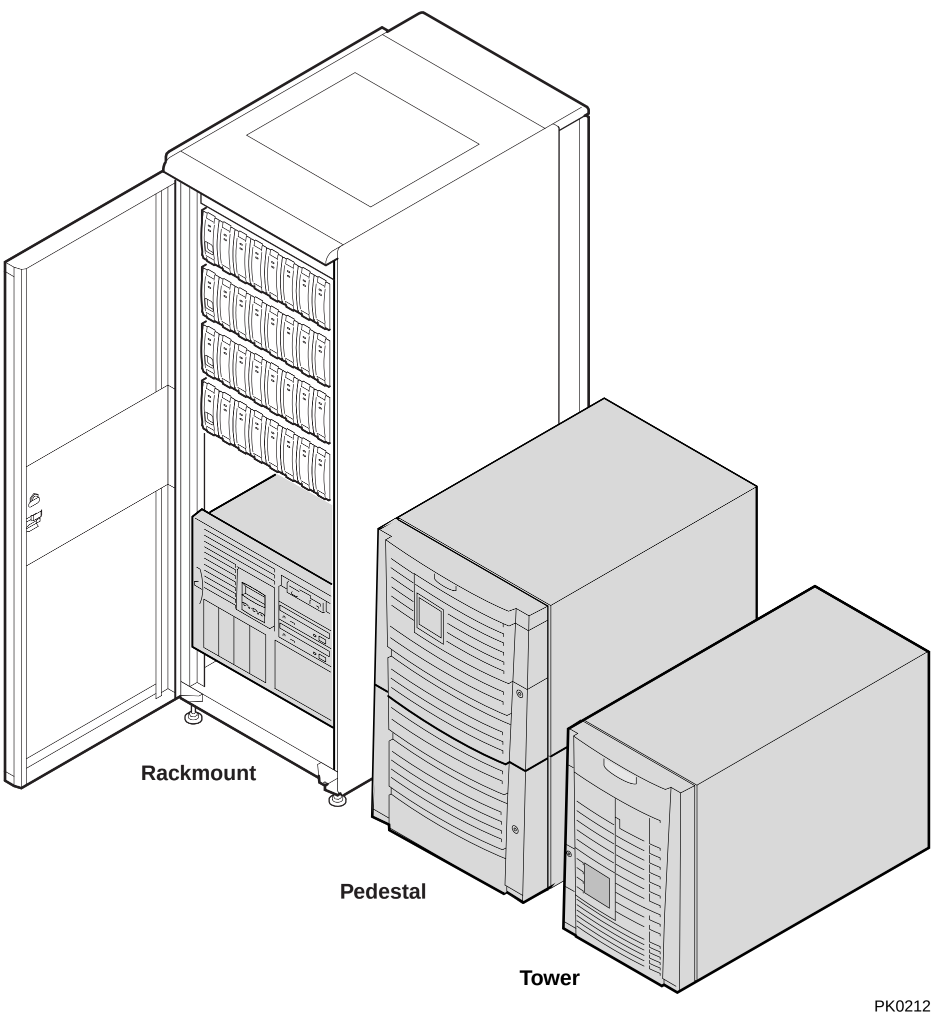
The chipset supports up to four CPUs and up to 32 Gbytes of memory. Interleaving occurs when at least two sibling or nonsibling memory arrays are used.

Two 256-bit memory buses support four memory arrays, yielding a maximum 5.2 Gbytes/sec system bandwidth. Transactions are ECC protected. Upon the receipt of data, the receiver checks for data integrity and corrects any errors.

1.2 System Enclosures

The *Compaq AlphaServer* ES40 family consists of a standalone tower, a pedestal with expanded storage capacity, and a cabinet.

Figure 1–2 Compaq AlphaServer ES40 Systems



Model Variants

AlphaServer ES40 systems are offered in two models. The entry-level model provides connectors for four DIMMs on each of the memory motherboards (MMBs) and connectors for six PCI options on the PCI backplane. To upgrade from Model 1 to Model 2, you replace the PCI backplane and the four memory motherboards.

Model 1	**Model 2**
1–4 CPUs	1–4 CPUs
Up to 16 DIMMs (4 DIMMs on each MMB	Up to 32 DIMMs (8 DIMMs on each MMB)
6 PCI slots	10 PCI slots

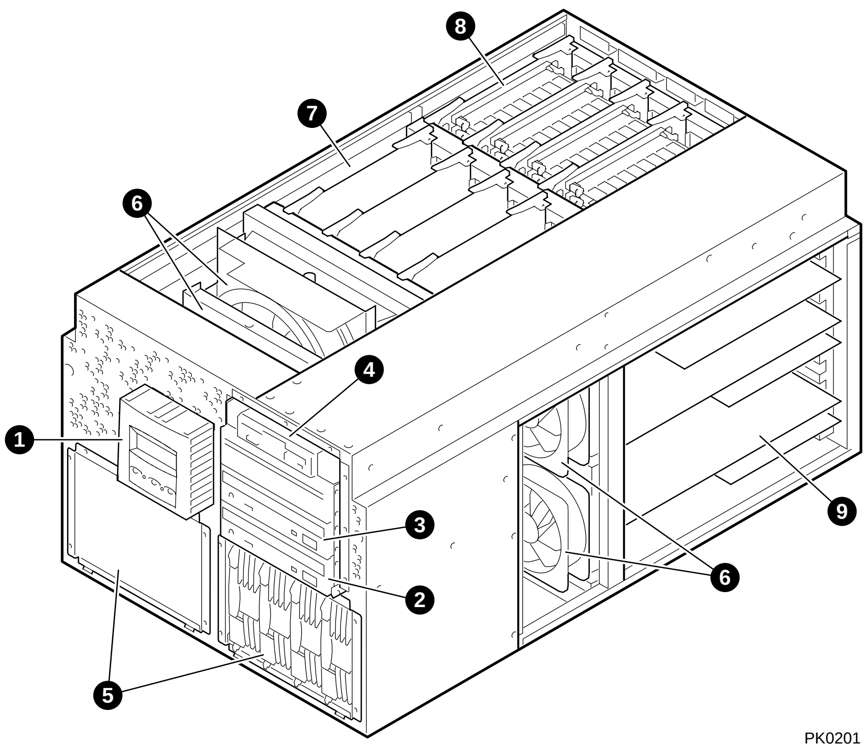
Common Components

The following components are common to all ES40 systems:

- Up to four CPUs, based on the 21264 Alpha chip
- Memory DIMMs (200-pin)
- Floppy diskette drive (3.5-inch, high density)
- CD-ROM drive
- Two half-height or one full-height removable media bays
- Up to two storage drive cages that house up to four 1.6-inch drives per cage
- Up to three 735-watt power supplies, offering N+1 power
- A 25-pin parallel port, two 9-pin serial ports, two universal serial bus (USB) ports, mouse and keyboard ports, and one MMJ connector for a local console terminal
- An operator control panel with a 16-character back-lit display and a Power button, Halt button, and Reset button

1.3 System Chassis—Front View/Top View

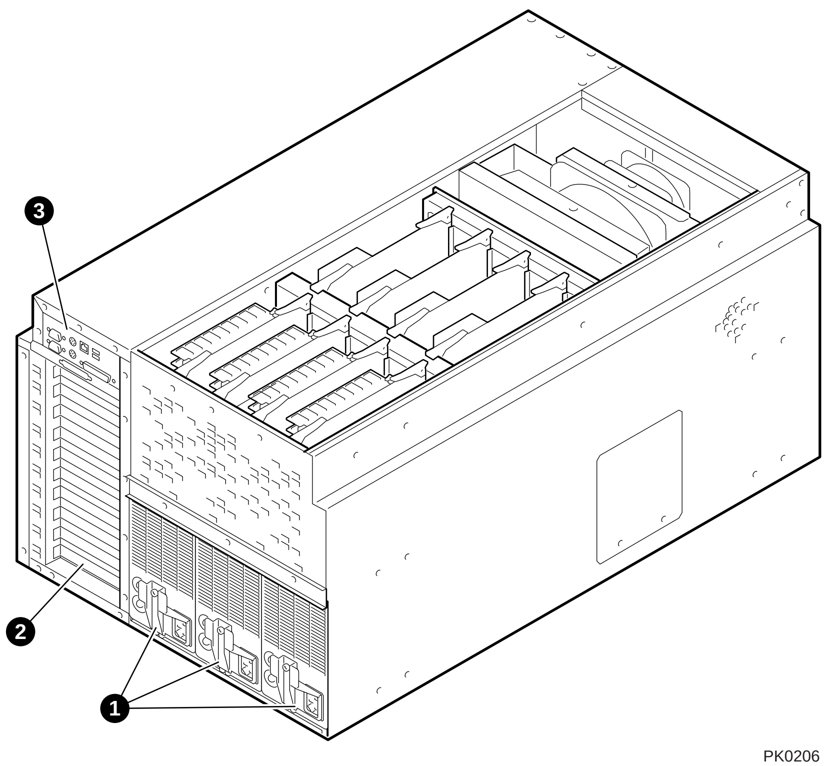
Figure 1–3 Components Top/Front View (Pedestal/Rackmount Orientation)



❶ Operator control panel
❷ CD-ROM drive
❸ Removable media bays
❹ Floppy diskette drive
❺ Storage drive bays
❻ Fans
❼ CPUs
❽ Memory
❾ PCI cards

1.4 System Chassis—Rear View

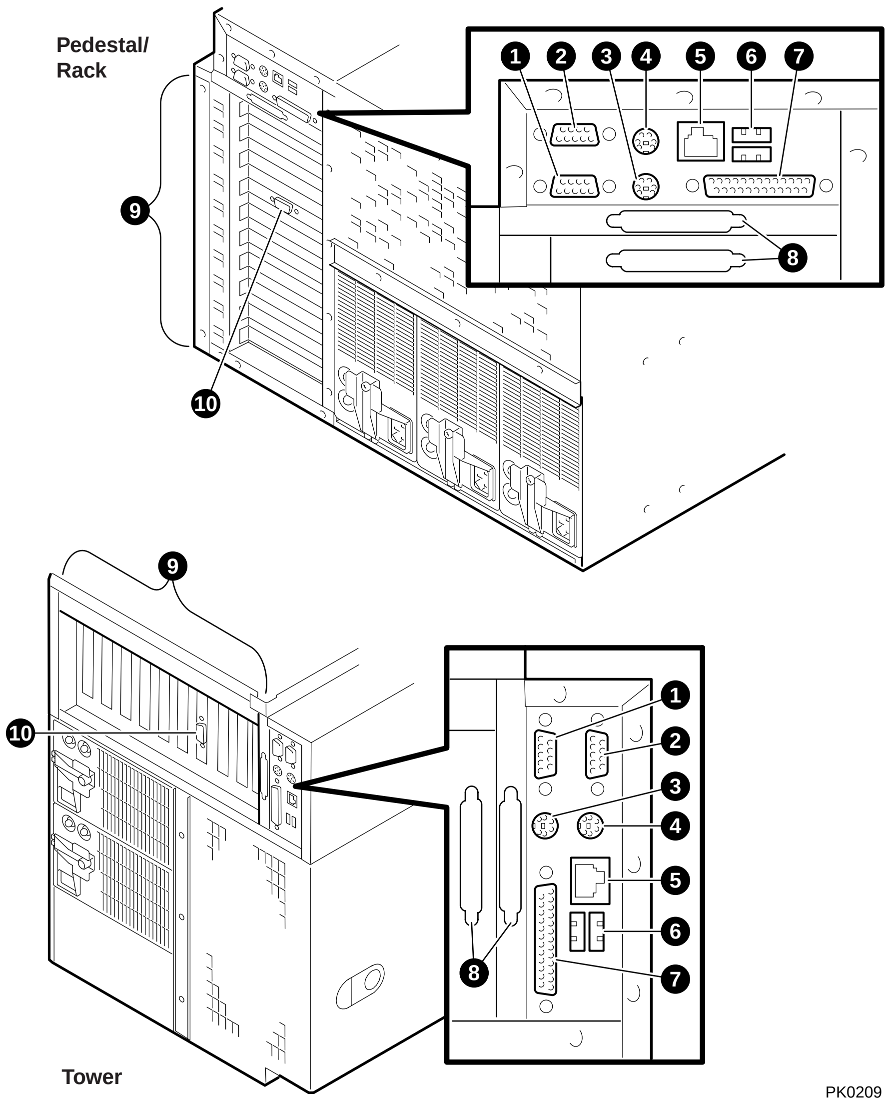
Figure 1–4 Rear Components (Pedestal/Rackmount Orientation)



❶ Power supplies
❷ PCI bulkhead
❸ I/O ports

1.5 I/O Ports and Slots

Figure 1–5 Rear Connectors



Rear Panel Connections

❶ Modem port—Dedicated 9-pin port for connection by modem to remote management console.

❷ COM2 serial port—Extra port to modem or any serial device.

❸ Keyboard port—To PS/2-compatible keyboard.

❹ Mouse port—To PS/2-compatible mouse.

❺ COM1 MMJ-type serial port/terminal port —For connecting a console terminal.

❻ USB ports.

❼ Parallel port—To parallel device such as a printer.

❽ SCSI breakouts.

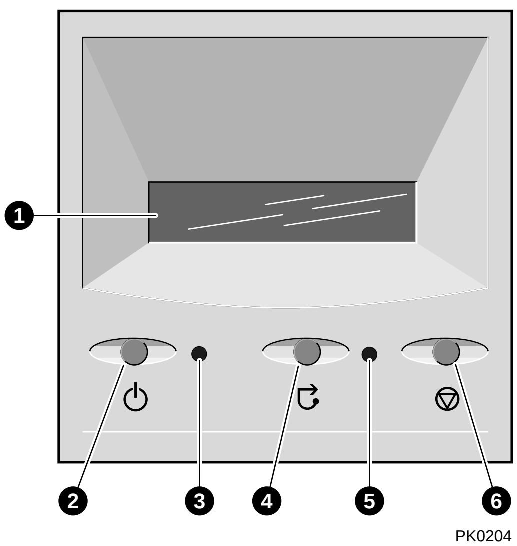
❾ PCI slots—For option cards for high-performance network, video, or disk controllers.

❿ PCI slot for VGA controller, if installed.

1.5 Control Panel

The control panel provides system controls and status indicators. The controls are the Power, Halt, and Reset buttons. A 16-character back-lit alphanumeric display indicates system state. The panel has two LEDs: a green Power OK indicator and an amber Halt indicator.

Figure 1–6 Control Panel



❶ Control panel display. A one-line, 16-character alphanumeric display that indicates system status during power-up and testing. During operation, the control panel is back lit.

❷ Power button. Powers the system on and off.

If a failure occurs that causes the system to shut down, pressing the power button off and then on clears the shutdown condition and attempts to power the system back on. Conditions that prevent the system from powering on can be determined by entering the **env** command from the remote management console (RMC) command line. The RMC is powered separately from the rest of the system and can operate as long as one power supply is plugged in. (See Chapter 7.)

❸ Power LED (green). Lights when the power button is depressed and system power passes initial checks.

❹ Reset button. A momentary contact switch that restarts the system and reinitializes the console firmware. Power-up messages are displayed, and then the console prompt is displayed or the operating system boot messages are displayed, depending on how the startup sequence has been defined.

❺ Halt LED (amber). Lights when you press the Halt button.

❻ Halt button. Halts the system.

- If Tru64 UNIX or OpenVMS is running, pressing the Halt button halts the operating system and returns to the SRM console. Pressing the Halt button does not halt the Windows NT operating system.
- If the Halt button is latched when the system is reset or powered up, the system halts in the SRM console, regardless of the operating system. UNIX and OpenVMS systems that are configured to autoboot cannot boot until the Halt button is unlatched.

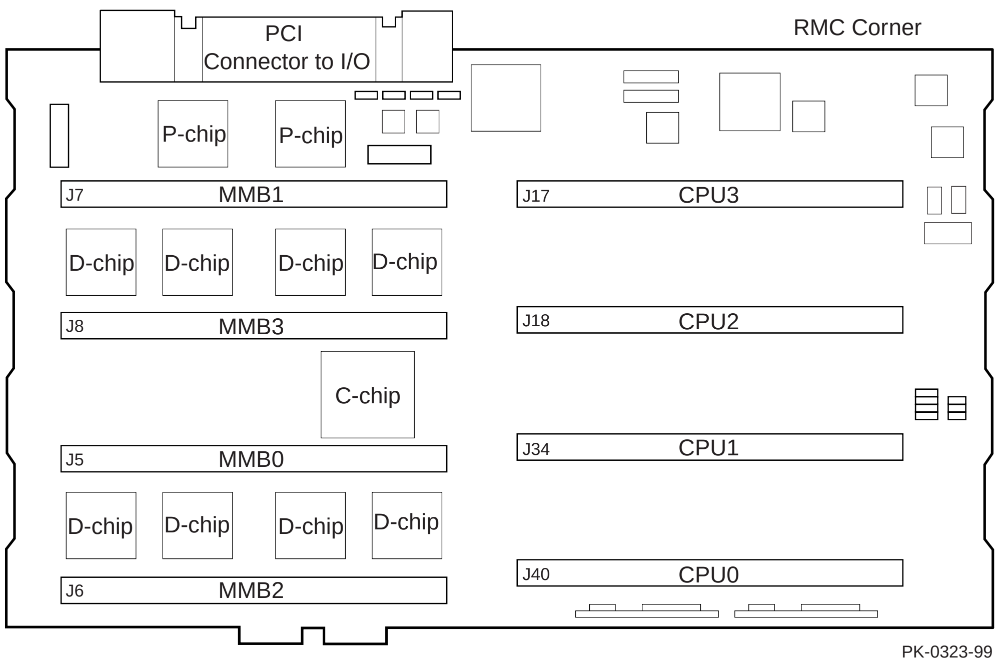
Commands issued from the remote management console (RMC) can be used to reset, halt, and power the system on or off.

RMC Command	Function
Power {off, on}	Equivalent to pressing the Power button on the system. If the Power button is in the off position, the RMC **power on** command has no effect.
Halt {in, out}	Equivalent to pressing the Halt button on the control panel to cause a halt (**halt in**) or releasing it from the latched position to deassert the halt (**halt out**).
Reset	Equivalent to pressing the Reset button on the control panel.

1.6 System Motherboard

The system motherboard is located on the floor of the system card cage. It has slots for the CPUs and memory motherboards (MMBs) and has the PCI backplane interconnect.

Figure 1–7 Component and Connector Locations

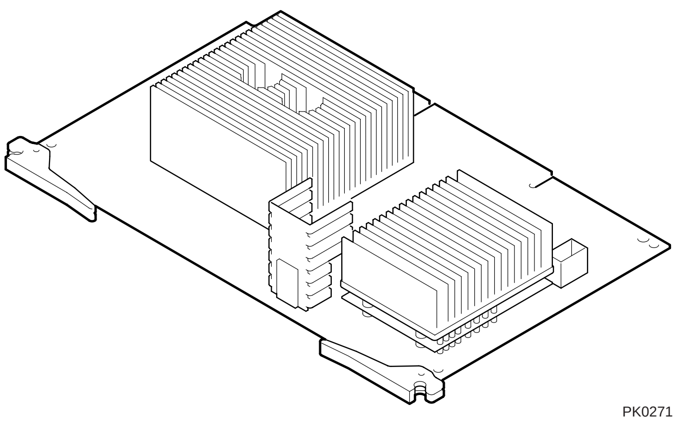


The system motherboard has the majority of the logic for the system, including the CPU, MMB connectors, the PCI connector to I/O, the D-chips and P-chips, the logic for the remote management console (RMC), and the jumpers for the fail-safe loader (FSL). Figure 1–7 shows the location of components and connectors on the system motherboard.

1.7 CPU Card

An *AlphaServer* ES40 can have up to four CPU cards. In addition to the Alpha 21264 chip, the CPU card has a 4-Mbyte second-level cache and a 2.2V DC-to-DC converter with heatsink that provides the required voltage to the Alpha chip. Power-up diagnostics are stored in a flash SROM on the card.

Figure 1–8 CPU Card



The 21264 microprocessor is a superscalar CPU with out-of-order execution and speculative execution to maximize speed and performance. It contains four integer execution units and dedicated execution units for floating-point add, multiply, and divide. It has an instruction cache and a data cache on the chip. Each cache is a 64 KB, two-way, set associative, virtually addressed cache that has 64-byte blocks. The data cache is a physically tagged, write-back cache.

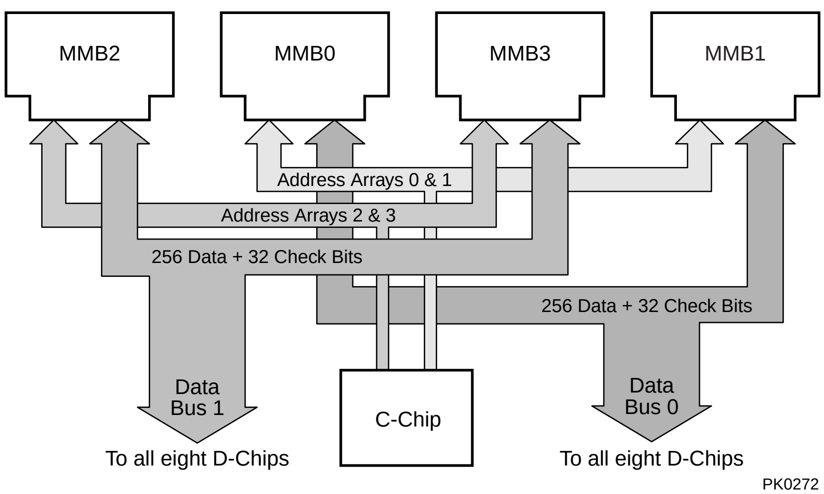
Each CPU card has a 4 MB secondary B-cache (backup cache) consisting of late-write synchronous static RAMs (SRAMs) that provide low latency and high bandwidth. Each CPU card also has a 5 ->2.2 volt power regulator that supplies up to 100 watts at 2.2 volts to the CPU.

See Chapter 6 for CPU configuration.

1.8 Memory Architecture and Options

The system has two 256-bit wide memory data buses, which can move large amounts of data simultaneously.

Figure 1–9 Memory Architecture



Memory Architecture

Memory throughput in this system is maximized by the following features:

- Two independent, wide memory data buses
- Very low memory latency (120 ns) and high bandwidth with 12 ns clock
- ECC memory

Each data bus is 256 bits wide (32 bytes). The memory bus speed is 83 MHz. This yields 2.6 GB/sec bandwidth per bus (32 x 83 MHz = 2.6 GB/sec). The maximum bandwidth is 5.2 GB/sec.

The switch interconnect design takes full advantage of the capabilities of the two wide data buses. The 256 data bits are distributed equally over two memory motherboards (MMBs). Simultaneously, in a read operation, 128 bits come from one MMB and the other 128 bits come from another MMB, to make one 256-bit read. Another 256-bit read operation can occur at the same time on the other independent data bus.

In addition, two address buses per MMB (one for each array) allow overlapping/pipelined accesses to maximize use of each data bus. When all arrays are identical (same size and speed), the memory is interleaved; that is, sequential blocks of memory are distributed across all four arrays.

Memory Options

Each memory option consists of four 100 MHz, 200-pin industry-standard DIMMs. The DIMMs are synchronous DRAMs. The Model 1 system supports up to four memory options (16 DIMMs), and the Model 2 system supports up to eight options (32 DIMMs). Memory options are available in the following sizes:

- 256 Mbytes (64 MB DIMMs)
- 512 Mbytes (128 MB DIMMs)
- 1 Gbyte (256 MB DIMMs)
- 2 Gbytes (512 MB DIMMs)

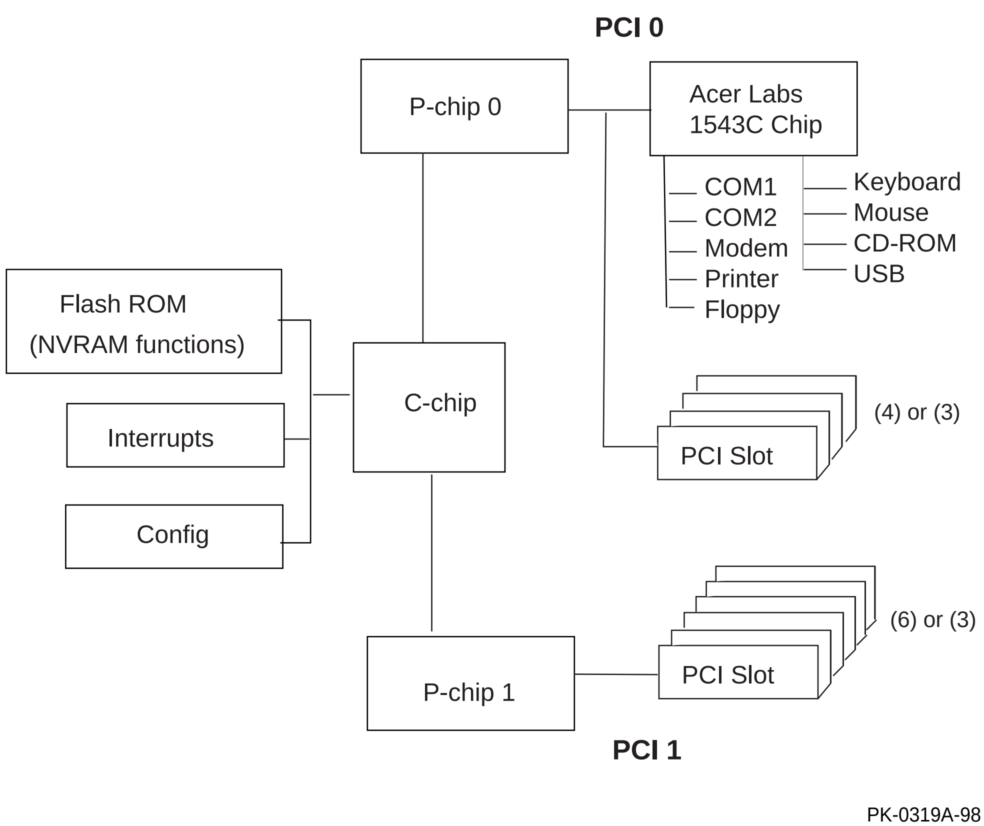
Memory options are installed into memory motherboards (MMBs) located on the system motherboard (see Figure 1–7). There are four MMBs. The MMBs have either four or eight slots for installing DIMMs.

See Chapter 6 for memory configuration.

1.9 PCI Backplane

The PCI backplane has two independent 64-bit, 33 MHz PCI buses that support 64-bit PCI slots. The 64-bit PCI slots are split across the two buses. The PCI buses support 3.3 V and 5 V options.

Figure 1–10 I/O Control Logic



NOTE: *No USB options are currently supported.*

PCI Bus Implementation

- Is fully compliant with the PCI Version 2.1 Specification
- Operates at 33 MHz, delivering a peak bandwidth of 500 MB/sec; over 250 Mbytes/sec for each PCI bus
- Has six option slots (Model 1) or ten option slots (Model 2)
- Supports three address spaces: PCI I/O, PCI memory, and PCI configuration space
- Supports byte/word, tri-byte, quadword, and longword operations
- Exists in noncached address space only

I/O Implementation

In a system with 10 I/O slots, PCI 0 has 4 slots, and PCI 1 has 6 slots. In a system with 6 slots, each PCI has 3 slots; the middle four connectors are not present.

The Acer Labs 1543C chip provides the bridge from PCI 0 to ISA. The C-chip controls accesses to memory on behalf of both P-chips.

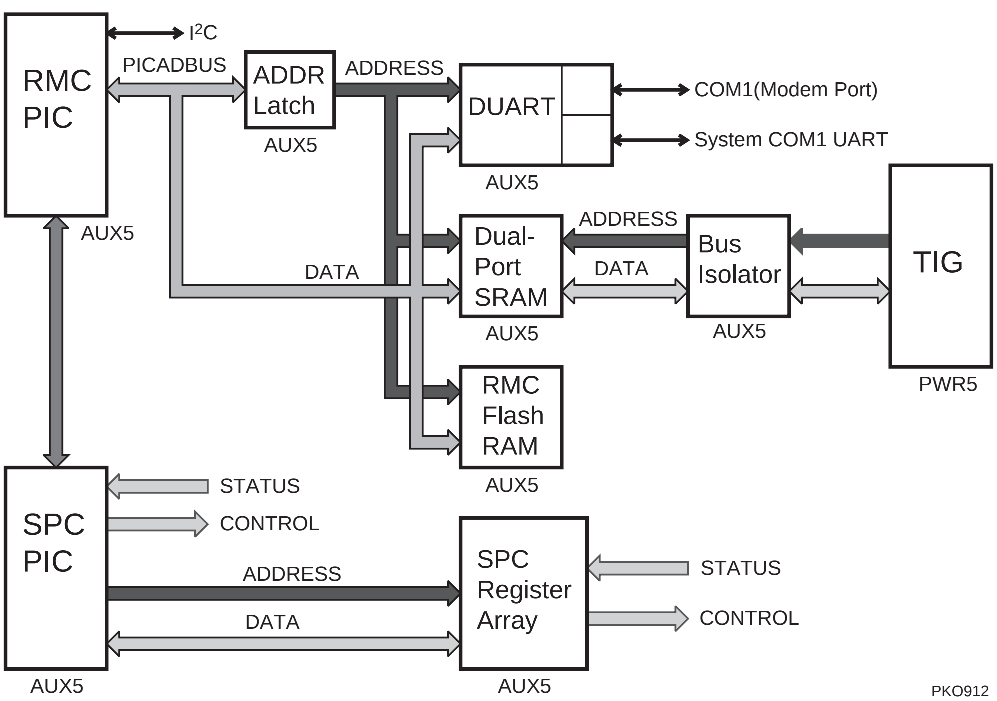
I/O Ports

The I/O ports are shown in Section 1.5.

1.10 Remote System Management Logic

The remote system management logic consists of two major elements: the System Power Controller (SPC), used to monitor and control system power supplies, regulators, and cooling apparatus; and the Remote Management Console (RMC), which facilitates remote interrogation and control of the system. The components used within the remote system management logic are powered by the AUX_5V supply, which is always present whenever AC input power is available to the system.

Figure 1–11 Remote System Management Logic Diagram



Dual-Port RAM (DPR)

The ES40 system features a dual-port RAM—RAM that is shared between the RMC and the system motherboard logic—to ease communication between the system and the RMC. This book refers to the dual-port RAM as the DPR.

The RMC reads 256 bytes of data from each FRU EEPROM at power-up and stores it in the DPR. This data contains configuration and possibly error log information. The data is accessible via the TIG chip to the firmware for configuration information during start-up. Remote or local applications can read the error log and configuration information. The error log information is written to the DPR by Compaq Analyze (see Chapter 5) and then written back to the EEPROMs by the RMC. This ensures that the error log is available on a FRU after power has been lost.

- Section 1.10.1 describes the SPC logic.
- Section 1.10.2 describes the RMC logic.

1.10.1 System Power Controller (SPC)

The System Power Controller (SPC) is responsible for sequencing the turn-on/turn-off of all power supplies and regulators, monitoring all system power supplies and regulators, generating hardware resets to all logic elements, and generating power system status signals for use by other functional units within the system. Additionally, it is responsible for emergency shutdown if the internal system temperature exceeds permissible limits.

An 8-bit CMOS microprocessor (PIC 17C44) with associated programming controls the functions of the SPC. The PIC processor receives inputs from:

- Operator control panel (power-on, reset)
- Power supplies and DC/DC regulators (Power-OK)
- Thermal sensors (temperature failure)
- TIG chip (command bus from the firmware)
- Remote management console logic (remote power up/down, reset)

It provides outputs to:

- Power supplies and DC/DC regulators (power supply enables)
- Processors (DC_OK, reset)
- TIG bus chip (handshake)
- Remote management console (power status)

1.10.2 Remote Management Console (RMC)

The remote management console (RMC) provides a mechanism for remotely monitoring a system and manipulating it on a very low level. It also provides access to the repository for all error information in the system. This provides the operator, either remotely or locally, with the ability to monitor the system (voltages, temperature, fans, error status) and manipulate it (reset, power on/off, halt) without any interaction on the part of the operating system.

The RMC can also detect alert conditions such as overtemperature, fan failure, and power supply failure and automatically dial a user-defined pager phone number or another computer system to make the remote operator aware of the alert condition.

The RMC logic is implemented using an 8-bit microprocessor (PIC 17C44) as the primary control device. Support devices include:

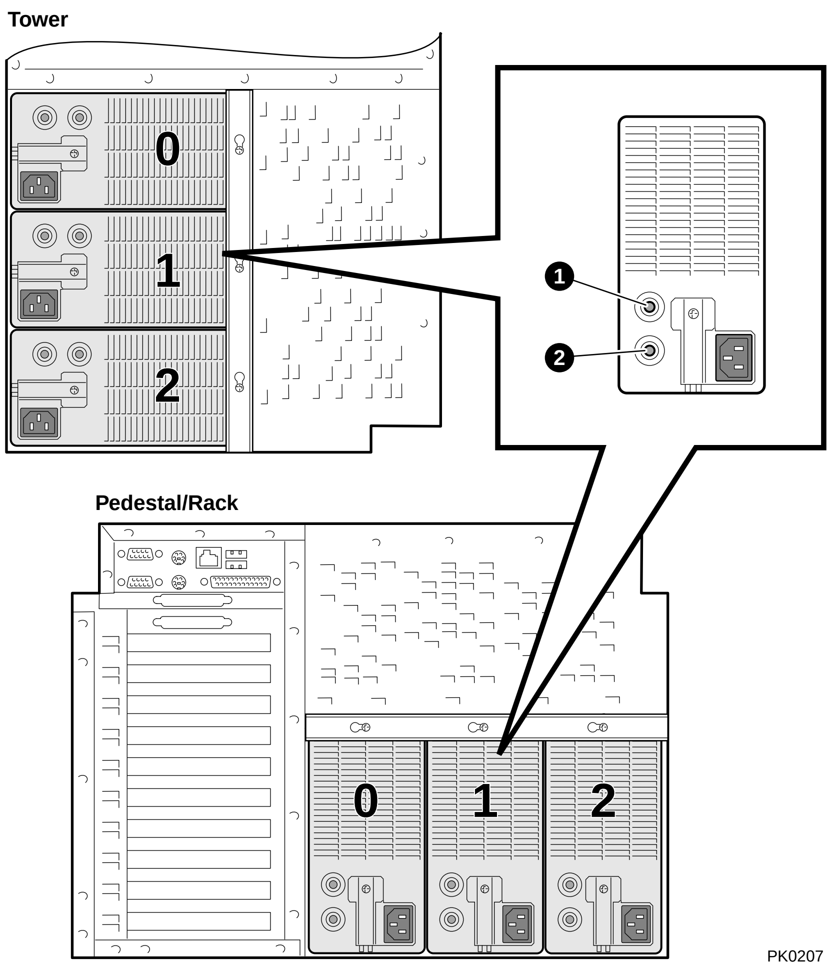
- Flash RAM (for code storage)
- Address latch
- Dual universal asynchronous receiver/transmitter (DUART)
- 8-bit I^2C port expanders
- I^2C temperature sensors
- I^2C nonvolatile memories (NVRAM)
- Programmable array logic (PAL)
- Dual-port RAM (DPR)
- RS232 drivers and receivers

Chapter 7 describes the operation and use of the RMC.

1.11 Power Supplies

The power supplies provide power to components in the system box. The number of power supplies required depends on the system configuration.

Figure 1–12 Power Supplies



One to three power supplies provide power to components in the system box. The system supports redundant power configurations to ensure continued system operation if a power supply fails. See Chapter 6 for power supply configurations.

When more than one power supply is installed, the supplies share the load. The power supplies select line voltage automatically (120V or 240V and 50 Hz or 60 Hz).

Power Supply LEDs

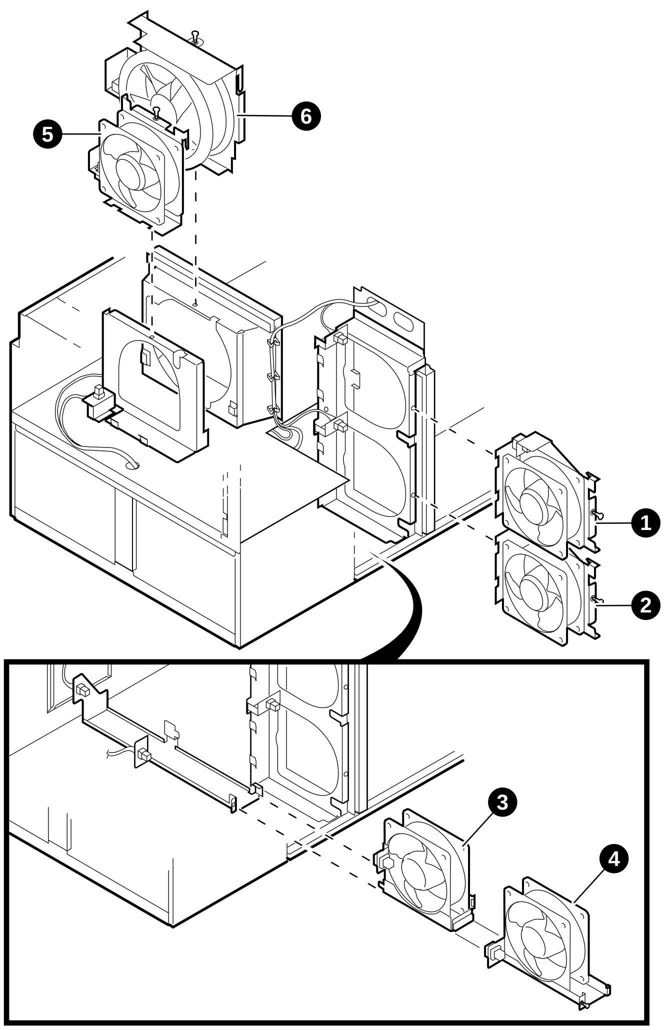
Each power supply has two green LEDs that indicate the state of power to the system.

❶ POK (Power OK)	Indicates that the power supply is providing power. The POK LED is on when the system is running. When the system power is on and a POK LED is off, that supply is not contributing to powering the system.
❷ +5 V Auxiliary	Indicates that AC power is flowing from the wall outlet. As long as the power supply cord is plugged into the wall outlet, the +5V Aux LED is always on, even when the system power is off.

1.12 Fans

The system has six hot-plug fans that provide front-to-back airflow.

Figure 1–13 System Fans



The system fans are shown in Figure 1–13 and described in Table 1–1.

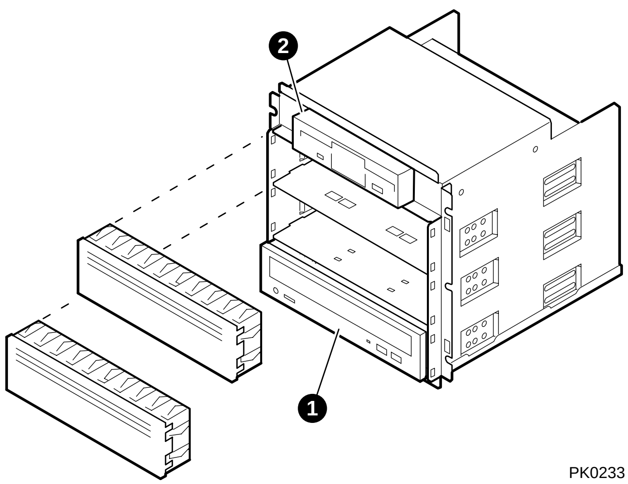
Table 1–1 Fan Descriptions

Fan Number	Area Cooled	Fan Failure Scenario
❶, ❷ 4.5-in.	PCI card cage Removable media Right drive cage	Both fans are powered at all times. If one fan fails, all other system fans speed up to provide adequate cooling. You can replace either fan while the system is running.
❸, ❹ 4.5-in.	Power supplies Left drive cage	Both fans are powered at all times. If one fan fails, all other system fans speed up to provide adequate cooling. You can replace either fan while the system is running.
❺ 4.5-in. redundant	CPU and memory card cage	Not powered unless the main fan fails. If the main fan fails, fan 5 runs at maximum speed to provide adequate cooling.
❻ 6.75-in. main fan	CPU and memory card cage	Fan 6 or fan 5 must always be running or the system will shut down. You can replace fan 6 as long as fan 5 is running.

1.13 Removable Media Storage

The system box houses a CD-ROM drive ❶ and a high-density 3.5-inch floppy diskette drive ❷ and supports two additional 5.25-inch half-height drives or one additional full-height drive. The 5.25-inch half height area has a divider that can be removed to mount one full-height 5.25-inch device.

Figure 1–14 Removable Media Drive Area

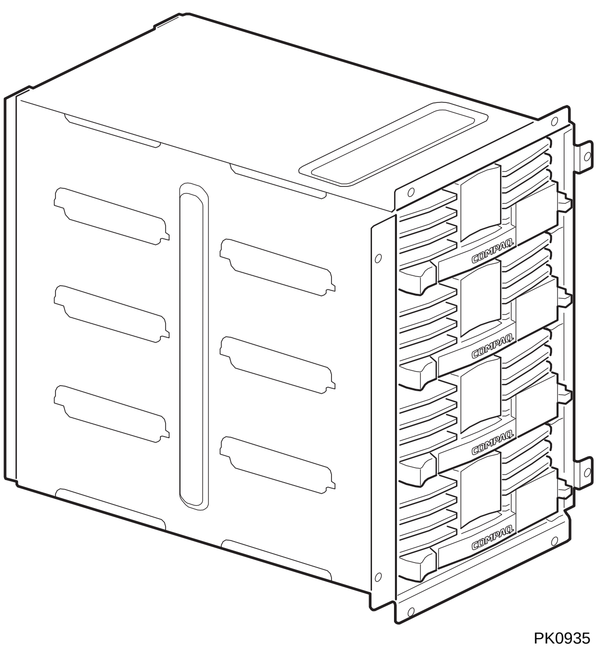


1.14 Hard Disk Drive Storage

The system chassis can have either one or two storage disk cages.

You can install four 1.6-inch hard drives in each storage disk cage. See Chapter 8 for information on replacing hard disk drives.

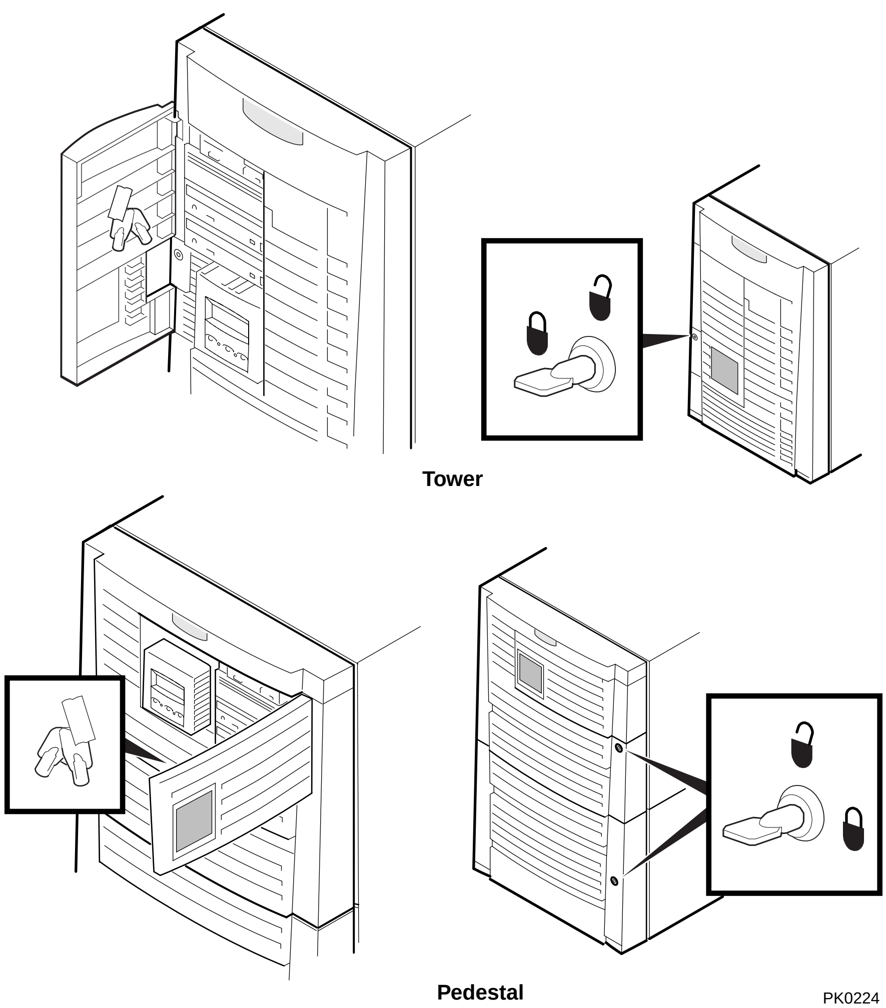
Figure 1–15 Hard Disk Storage Cage with Drives (Tower View)



1.15 System Access

At the time of delivery, the system keys are taped inside the small front door that provides access to the operator control panel and removable media devices.

Figure 1–16 System Lock and Key



Both the tower and pedestal systems have a small front door through which the control panel and removable media devices are accessible. At the time of delivery, the system keys are taped inside this door.

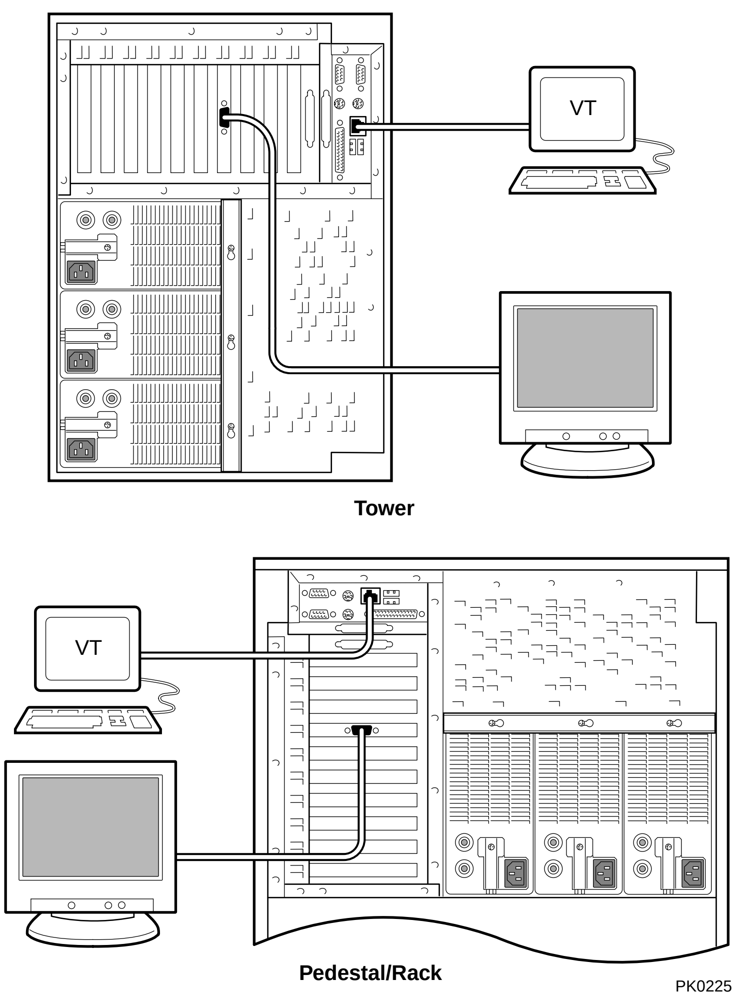
The tower front door has a lock that lets you secure access to the disk drives and to the rest of the system.

The pedestal has two front doors, both of which can be locked. The upper door secures the disk drives and access to the rest of the system, and the lower door secures the expanded storage.

1.16 Console Terminal

The console terminal can be a serial (character cell) terminal connected to the COM1 or COM2 port or a VGA monitor connected to a VGA adapter on PCI 0. A VGA monitor requires a keyboard and mouse.

Figure 1–17 Console Terminal Connections (Local)



Chapter 2
Troubleshooting

This chapter describes the starting points for diagnosing problems on *Compaq AlphaServer* ES40 systems. The chapter also provides information resources.

- Questions to Consider
- Diagnostic Tables
- Service Tools and Utilities
- Information Resources

2.1 Questions to Consider

Before troubleshooting any system problem, first check the site maintenance log for the system's service history.

Be sure to ask the system manager the following questions:

- Has the system been used and did it work correctly?
- Have changes to hardware or updates to firmware or software been made to the system recently? If so, are the revision numbers compatible for the system? (Refer to the hardware and operating system release notes.)
- What is the current state of the system?
  - If the operating system is down, but you are able to access the SRM console, use the console environment diagnostic tools, including the OCP display, power-up display, and SRM commands.
  - If you are unable to access the SRM console, enter the RMC CLI and issue commands to determine the hardware status. See Chapter 7.
  - If the operating system has crashed and rebooted, the CCAT (Compaq Crash Analysis Tool), the Compaq Analyze service tools (to interpret error logs), the SRM **crash** command, operating system exercisers, and DEC VET can be used to diagnose system problems.

2.2 Diagnostic Tables

System problems can be classified into the following five categories. Using these categories, you can quickly determine a starting point for diagnosis and eliminate the unlikely sources of the problem.

1. Power problems—Table 2–1
2. No access to console mode—Table 2–2
3. Console-reported failures—Table 2–3
4. Boot problems—Table 2–4
5. Errors reported by the operating system—Table 2–5

Table 2–1 Power Problems

Symptom	Action	Reference
System does not power on.	• Check error messages on the OCP. • Check that AC power is plugged in. • Check that the ambient room temperature is within environmental specifications (10–40° C, 50–104° F). • Check the Power setting on the control panel. Toggle the Power button to off, then back on to clear a remote power disable. • Check that internal power supply cables are plugged in at the system motherboard.	
Power supply shuts down after a few seconds	The system may be powered off by one of the following: Loss of AC power RMC **power off** command System software Multiple fan failure Overtemperature condition Power supply failure (If N+1 config. multiple power supply failure Faulty CPU (CPU DC/DC converter failure)	
	If AC power is present, use the RMC **env** command to check environmental status.	Chapter 7
	Check jumper J26. If the system must be kept running, this jumper can be positioned to override an overtemperature condition.	Appendix B

Table 2–2 Problems Getting to Console Mode

Symptom	Action	Reference
Power-up screen is not displayed at system console.	Note any error beep codes and observe the OCP display for a failure detected during self-tests.	Chapter 3
	Check keyboard and monitor connections.	Chapter 1
	Press the Return key. If the system enters console mode, check that the **console** environment variable is set correctly.	
	If the console terminal is a VGA monitor, the **console** variable should be set to **graphics**. If it is a serial terminal, the console environment variable should be set to **serial**.	Chapter 6
	If **console** is set to **serial**, the power-up screen is routed to the COM1 serial communication port or MMJ port and cannot be viewed from the VGA monitor.	
	Try connecting a console terminal to the COM1 serial communication port. When using the COM1 port set the **console** environment variable to **serial**.	Chapter 6
	Use RMC commands to determine status.	Chapter 7

Table 2–3 Problems Reported by the Console

Symptom	Action	Reference
No SRM messages are displayed after the "jump to console" message.	Console firmware is corrupted. Load new firmware with fail-safe loader.	Chapter 3
The system attempts to boot from the floppy drive after a checksum error is reported.	The system automatically reverts to the fail-safe loader to load new SRM and AlphaBIOS firmware. If the fail-safe load does not work, replace the system motherboard.	Chapter 3 and Chapter 8
Console program reports error:		
• Error beep codes report an error at power-up.	Use the error beep codes and OCP messages to determine the error.	Chapter 3
• Power-up screen includes error messages.	Examine the console event log (**more el** command).	Chapter 4
• Power-up screen or console event log indicates problems with mass storage devices.	Check cables and seating of drives. Check power to an external storage box.	
• Storage devices are missing from the **show config** display.	Check cables and seating of drives. Check power to an external storage box.	
• PCI devices are missing from the **show config** display.	Checking seating of modules.	

Table 2–4 Boot Problems

Symptom	Action	Reference
System cannot find boot device.	Check the system configuration for the correct device parameters (node ID, device name, and so on). • For UNIX and OpenVMS, use the **show config** and **show device** commands. • For Windows NT, use the AlphaBIOS Display System Configuration menu and the CMOS Setup menus.	Chapter 6
	Check the system configuration for the correct environment variable settings. • For UNIX and OpenVMS, examine the **auto_action**, **bootdef_dev**, **boot_osflags**, and **os_type** environment variables. • For network boots, make sure **ei*0_protocols** or **ew*0_protocols** is set to **bootp** for UNIX or **mop** for OpenVMS. • For Windows NT, examine the Auto Start and Auto Start Count options on the CMOS Setup menu.	Chapter 6
Device does not boot.	For problems booting over a network, make sure **ei*0_protocols** or **ew*0_protocols** is set to **bootp** for UNIX or **mop** for OpenVMS.	Chapter 6
	Run the **test** command to see if the boot device is operating.	Chapter 4

Table 2–5 Errors Reported by the Operating System

Symptom	Action	Reference
System is hung, but SRM console is operating	Press the Halt button and enter the **crash** command to provide a crash dump file for analysis (OpenVMS and UNIX only).	Chapter 4
	Refer to *OpenVMS Alpha System Dump Analyzer Utility Manual* for information on how to interpret OpenVMS crash dump files.	
	Refer to the *Guide to Kernel Debugging* for information on using the UNIX Krash Utility.	
	Use the SRM **info** command to display registers and data structures.	Chapter 4
	If the problem is intermittent, run the SRM **test** and **sys_exer** commands.	Chapter 4
System is hung and SRM console is not operating.	Invoke the RMC CLI and enter the **dump** command to access DPR locations.	Chapter 7
Operating system has crashed and rebooted.	Examine the operating system error log files to isolate the problem.	
	If the problem is intermittent, ensure that Compaq Analyze has been installed and is running in background mode (GUI does not have to be running) to determine the defective FRU.	Chapter 5

2.3 Service Tools and Utilities

This section lists some of the tools and utilities available for acceptance testing and diagnosis and gives recommendations for their use.

2.3.1 Error Handling/Logging Tools (Compaq Analyze)

The Tru64 UNIX, OpenVMS, and Microsoft Windows NT operating systems provide fault management error detection, handling, notification, and logging.

The primary tool for error handling is Compaq Analyze, a fault analysis utility designed to analyze both single and multiple error/fault events. Compaq Analyze uses error/fault data sources other than the traditional binary error log. See Chapter 5.

2.3.2 Loopback Tests

Internal and external loopback tests are used to test the components on the I/O connector assembly ("junk I/O") and to test Ethernet cards. The loopback tests are a subset of the SRM diagnostics.

Use loopback tests to isolate problems with the COM2 serial port, the parallel port, and Ethernet controllers. See the **test** command in Chapter 4 for instructions on performing loopback tests.

2.3.3 SRM Console Commands

SRM console commands are used on systems running Tru64 UNIX or OpenVMS to set and examine environment variables and device parameters. For example, the **show configuration** and **show device** commands are used to examine the configuration, and the **set *envar*** command is used to set environment variables.

SRM commands are also used to invoke ROM-based diagnostics and to run native exercisers. For example, the **test** and **sys_exer** commands are used to test the system.

See Chapter 6 for information on configuration-related console commands and environment variables. See Chapter 4 for information on running console exercisers. See Appendix A for a list of console commands used most often on ES40 systems.

2.3.4 AlphaBIOS Menus

The AlphaBIOS Standard CMOS Setup menu and the Advanced CMOS Setup menu are used to configure Windows NT systems.

Standard CMOS Setup	Advanced CMOS Setup
Enable/disable Auto Start	Set length of memory test
Set date and time	Enable/disable PCI parity
Configure floppies	Enable/disable password protection
Configure keyboard	Enable/disable SCSI BIOS

You can view the hardware configuration for a system running Windows NT by selecting items on the **Display System Configuration** menu accessed from the AlphaBIOS Setup screen.

Use AlphaBIOS menus for viewing system configuration and configuring systems running Windows NT. You can also set the length of the memory test done at power-up from Advanced CMOS Setup.

The AlphaBIOS Utilities menu has a Display Error Frames selection that allows you to view hardware error reports on fatal error halts or double error halts. See Chapter 5.

2.3.5 Remote Management Console (RMC)

The remote management console (RMC) is used for managing the server either locally or remotely. It also plays a key role in error analysis by passing error log information to the dual-port RAM (DPR), which is shared between the RMC and the system motherboard logic, so that this information can be accessed by the system. RMC also controls the control panel display. RMC has a command-line interface from which you can enter a few diagnostic commands.

RMC can be accessed as long as the power cord for a working supply is plugged into the AC wall outlet and a console terminal is attached to the system. This feature ensures that you can gather information when the operating system is down and the SRM console is not accessible. See Chapter 7.

2.3.6 Operating System Exercisers (DEC VET)

The Verifier and Exerciser Tool (DEC VET) is supported by the Tru64 UNIX, OpenVMS, and Windows NT operating systems. DEC VET is an on-line diagnostic tool used to ensure the proper installation and operation of hardware and base operating system software. Use DEC VET as part of acceptance testing to ensure that the CPU, memory, disk, tape, file system, and network are interacting properly.

2.3.7 Crash Dumps

For fatal errors, the Tru64 UNIX and OpenVMS operating systems save the contents of memory to a crash dump file. This file can be used to determine why the system crashed.

CCAT, the Compaq Crash Analysis Tool, is the primary crash dump analysis tool for analyzing crash dumps on Alpha systems running Tru64 UNIX or OpenVMS. CCAT compares the results of a crash dump with a set of rules. If the results match one or more rules, CCAT notifies the system user of the cause of the crash and provides information to avoid similar crashes in the future. CCAT does not currently support AlphaServer systems running Windows NT.

Windows NT provides the Windows NT Crash Dump Collector, a client/server application that automatically transfers crash information from the client machine to a centralized server. A control panel application is included, which allows the customer to control the transfer of crash information.

2.3.8 Revision and Configuration Management Tool (RCM)

RCM is a tool to assist with revision and configuration management for hardware, firmware, operating system, and software products. It collects configuration and revision data from a system and stores it. A report generator produces configuration, change, and comparison reports that are useful in finding revision incompatibilities. RCM also helps you verify service actions. For example, if a new board was supposed to be installed, you can use RCM to verify that the installation was done.

RCM is accessible from the following Web site:

http://smsat-www.ilo.dec.com/products/rcm/service/index.htm

2.3.9 StorageWorks Command Console (SWCC)

The StorageWorks Command Console (SWCC) is a storage management software tool that allows you to configure and monitor storage graphically from a single management console. It also has distributed capabilities that let you view multiple servers at the same time in a Microsoft Explorer-like navigation pane.

The StorageWorks Command Console's client is a graphical user interface (GUI) that can configure and monitor StorageWorks RAID Array solutions. The client runs on Windows NT (Intel only) or Windows 95. The Command Console agent runs on the host system and communicates with the client over a TCP/IP network connection, a SCSI connection, or a serial connection.

You can download the Command Console from the following Web site:

http://www.storage.digital.com/homepage/support/swcc/

2.4 Information Resources

Many information resources are available, including tools that can be downloaded from the Internet, firmware updates, a supported options list, and more.

2.4.1 Compaq Service Tools CD

The Compaq Service Tools CD-ROM enables field engineers to upgrade customer systems with the latest version of software when the customer does not have access to Compaq Web pages. The CD-ROM Web site is:

http://caspian1.zko.dec.com/service_tools/

2.4.2 AlphaServer ES40 Service HTML Help File

The information contained in this guide, including the FRU procedures and illustrations, is available in HTML Help format as part of the Maintenance Kit (QZ-01BAB-GZ).

2.4.3 Alpha Systems Firmware Updates

The AlphaBIOS firmware for Windows NT and the SRM firmware for Tru64 UNIX and OpenVMS reside in the flash ROM on the system motherboard. You can obtain the latest system firmware from CD-ROM or over the network.

Quarterly Update Service

The Alpha Systems Firmware Update Kit CD-ROM is available by subscription.

Alpha Firmware Internet Access

- You can obtain Alpha Firmware updates from the World Wide Web from the following Web site:

  http://ftp.digital.com/pub/Digital/Alpha/firmware/readme.html

  The README file describes the firmware directory structure and how to download and use the files.

- If you do not have a Web browser, you can download the files using anonymous ftp:

  ftp.digital.com/pub/Digital/Alpha/firmware

- Individual Alpha system firmware releases that occur between releases of the firmware CD are located in the interim directory:

  ftp.digital.com/pub/Digital/Alpha/firmware/interim

AlphaBIOS Firmware

The AlphaBIOS firmware is included in the Alpha Systems Firmware Update Kit CD-ROM.

2.4.4 Fail-Safe Loader

The fail-safe loader (FSL) allows you to boot a firmware update utility diskette in an attempt to repair corrupted console files that reside within the flash ROMs on the system motherboard. You can download the fail-safe loader from the Internet (using the firmware update URL above) to create your own fail-safe loader diskette. See Chapter 3 for information on forcing a fail-safe floppy load.

2.4.5 Software Patches

Software patches for the supported operating systems are available from the World Wide Web as follows:

http://www.digital.com/alphaserver/support.html

2.4.6 Late-Breaking Technical Information

You can download up-to-date files and late-breaking technical information from the Internet.

The information includes firmware updates, the latest configuration utilities, software patches, lists of supported options, and more.

http://www.digital.com/alphaserver/es40/es40.html

2.4.7 Supported Options

A list of options supported on the system is available on the Internet:

http://www.digital.com/alphaserver/es40/es40_sol.pdf

Chapter 3
Power-Up Diagnostics and Display

This chapter describes the power-up process and RMC, SROM, and SRM power-up diagnostics. The following topics are covered:

- Overview of Power-Up Diagnostics
- System Power-Up Sequence
- Power-Up Displays
- Power-Up Error Messages
- Forcing a Fail-Safe Floppy Load
- Updating the RMC

3.1 Overview of Power-Up Diagnostics

The power-up process begins with the power-on of the power supplies. After the AC and DC power-up sequences are completed, the remote management console (RMC) reads EEROM information and deposits it into the DPR. The SROM minimally tests the CPUs, initializes and tests backup cache, and minimally tests memory. Finally, the SROM loads the SRM console program into memory and jumps to the first instruction in the console program.

There are three distinct sets of power-up diagnostics:

1. System power controller and remote management console diagnostics—These diagnostics check the power regulators, temperature, and fans. Failures are reported in the dual-port RAM (DPR) and on the OCP display. Certain failures may prevent the system from powering on.
2. Serial ROM (SROM) diagnostics—SROM tests check the basic functionality of the system and load the console code from the FEPROM on the system motherboard into system memory. Failures during SROM tests are indicated by error beep codes and messages on the serial console terminal and the OCP.
3. Console firmware diagnostics—These tests are executed by the SRM console code. They test the core system, including boot path devices. Failures during these tests are reported to the console terminal through the power-up screen or console event log.

3.2 System Power-Up Sequence

The power-up sequence is described below and illustrated in Figure 3–1.

1. When the power cord is plugged into the wall outlet, 5V auxiliary AC voltage is enabled. The 5 V AUX LEDs on the power supplies are lit, and the system power controller and RMC are initialized.
2. Pressing the Power button on the control panel or subsequently issuing the **power-on** command from the RMC turns on power to the power supplies, CPU converters, and VTERM regulators. The POK LEDs on the power supplies are lit and the power supplies are tested. If all power supplies are bad, power-up stops. All DC/DC converters and regulators are then tested. If any converter or regulator is bad, power-up stops.
3. CPU_DCOK and SYS_DC_OK are set to "true," which means that DC power on the CPUs and system is okay. All CPUs load the initial Y divisor (clock multiplier). The OCP power LED is lit.
4. SYS_RESET is set to "false." This setting releases the system motherboard logic and PCI backplane logic from the Reset state.
5. The primary CPU is selected and CPU_(P)_RESET is set to "false." This allows the primary CPU to attempt to load flash SROM code.
6. If the primary CPU is good, it loads flash SROM. If bad, the system tries the next available CPU and if that CPU is good, it becomes the primary. The remaining CPUs load flash SROM. The SROM power-up then continues, as described in Section 3.3.

Figure 3–1 Power-Up Sequence

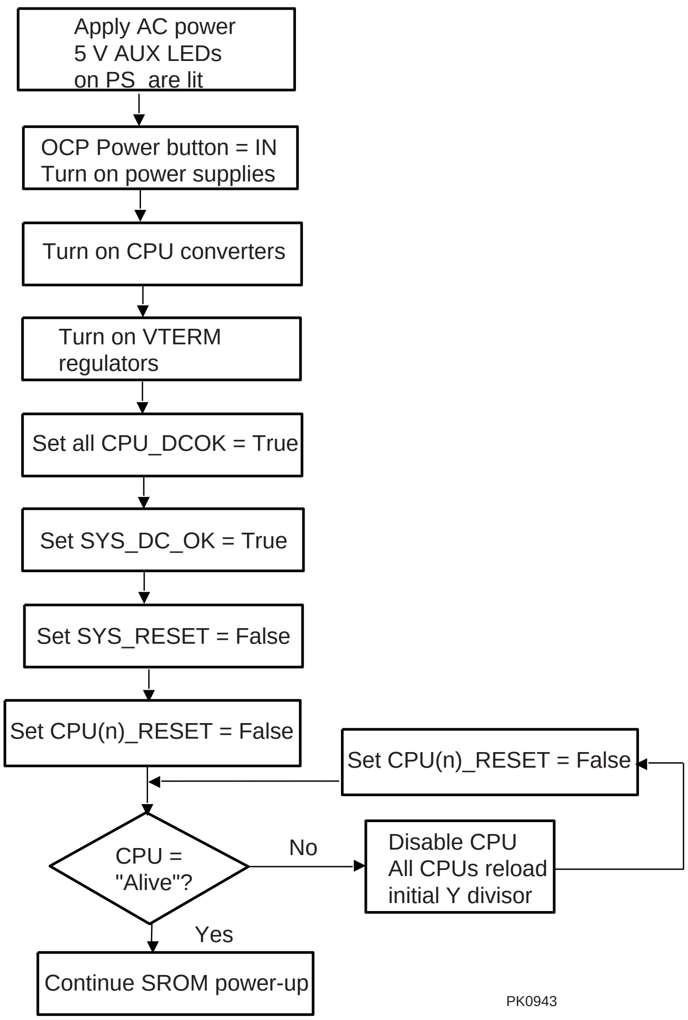
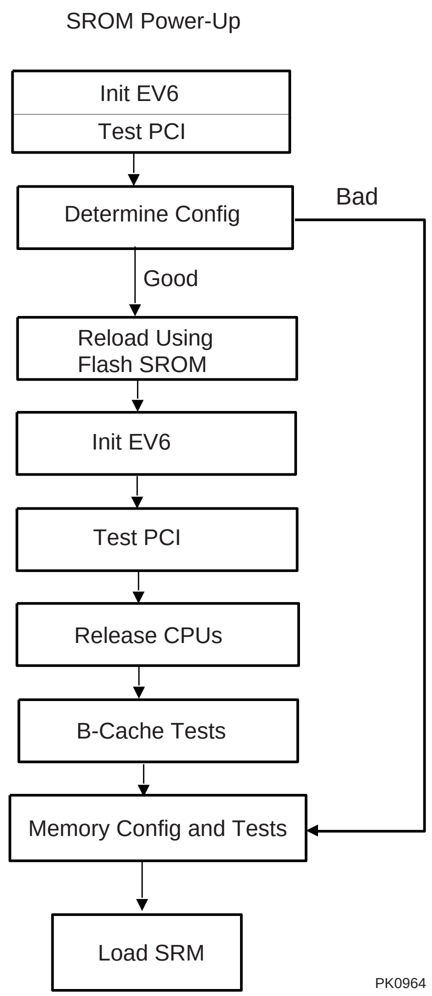


Figure 3–1 Power-Up Sequence (Continued)



3.3 Power-Up Displays

Power-up information is displayed on the operator control panel and on the console terminal startup screen. Messages sent from the RMC and SROM programs are displayed first, followed by messages from the SRM console.

NOTE: *The power-up text that is displayed on the screen depends on what kind of terminal is connected as the console terminal: VT or VGA.*

*If the SRM **console** environment variable is set to **serial**, the entire power-up display, consisting of the SROM and SRM power-up messages, is displayed on the VT terminal screen. If **console** is set to **graphics**, no SROM messages are displayed, and the SRM messages are delayed until VGA initialization has been completed.*

- Section 3.3.1 describes the SROM power-up sequence and shows the SROM power-up messages and corresponding OCP messages.
- Section 3.3.2 shows the messages that are displayed once the SROM has transferred control to the SRM console.

3.3.1 SROM Power-Up Display

Example 3–1 Sample SROM Power-Up Display

SROM Power-Up Display	OCP Message	
	PCI Test	❶
SROM V1.00 CPU #00 @ 0500 MHz	Power on	❷
SROM program starting	Reload	
Reloading SROM		
SROM T1.5-F CPU # 00 @ 0500 MHz		
SROM program starting		
Starting secondary on CPU #1	RelCPU1	❸
Starting secondary on CPU #2	RelCPU2	
Starting secondary on CPU #3	RelCPU3	
Bcache data tests in progress	BC Data	❹
Bcache address test in progress	BC Addr	
CPU parity and ECC detection in progress		
Bcache ECC data tests in progress		
Bcache TAG lines tests in progress		
Memory sizing in progress	Size Mem	❺
Memory configuration in progress	Cfg Mem	
Memory data test in progress		
Memory address test in progress		
Memory pattern test in progress		
Memory thrashing test in progress		
Memory initialization		
Loading console	Load ROM	❻
Code execution complete (transfer control)	Jump to Console	

SROM Power-Up Sequence

❶ When the system powers up, the SROM code is loaded into the I-cache (instruction cache) on the first available CPU, which becomes the primary CPU. The order of precedence is CPU0, CPU1, and so on. The primary CPU attempts to access the PCI bus. If it cannot, either a hang or a failure occurs, and this is the only message displayed.

❷ The primary CPU interrogates the I^2C EEROM as stored in the DPR. The primary CPU determines the optimum CPU and system configuration to jump to.

The primary CPU next checks the SROM checksum to determine the validity of the flash SROM sectors.

If flash SROM is invalid, the primary CPU reports the error and continues the execution of the SROM code. Invalid flash SROM must be reprogrammed.

If flash SROM is good, the primary CPU programs appropriate registers with the values from the flash data and selects itself as the target CPU to be loaded.

❸ The primary CPU (usually CPU0) initializes and then loads the flash SROM code to the next CPU. That CPU then initializes the EV6 (21264 chip) and marks itself as a secondary CPU. Once the primary CPU sees the secondary, it loads the flash SROM code to the next CPU until all remaining CPUs are loaded.

❹ The flash SROM performs B-cache tests. For example, the ECC data test verifies the detection logic for single- and double-bit errors.

❺ The primary CPU sizes memory and initiates all memory tests. The memory is tested for address and data errors for the first 32 MB of memory. It also initializes all the “sized” memory in the system.

If a memory failure occurs, an error is reported. An untested memory array is assigned to address 0 and the failed memory array is deassigned. The memory tests are re-run on the first 32 MB of memory. If all memory fails, the “No Memory Available” message is reported and the system halts.

❻ If all memory passes, the primary CPU loads the console and transfers control to it.

3.3.2 SRM Console Power-Up Display

When SROM power-up is complete, the primary CPU transfers control to the SRM console program. The console program continues the system initialization. Failures are reported to the console terminal through the power-up screen and a console event log.

Example 3–2 SRM Power-Up Display

```
OpenVMS PALcode V1.50-0,  Tru64 UNIX PALcode V1.45-5

starting console on CPU 0                                  ❶
initialized idle PCB
initializing semaphores
initializing heap
initial heap 200c0
memory low limit = 144000
heap = 200c0, 17fc0
initializing driver structures
initializing idle process PID
initializing file system
initializing hardware
initializing timer data structures
lowering IPL
CPU 0 speed is 2.00 ns (500MHz)
create dead_eater
create poll
create timer
create powerup
access NVRAM
Memory size 2048 MB
testing memory                                             ❷
...
probe I/O subsystem                                        ❸
probing hose 1, PCI
bus 0, slot 2, function 0 -- pka -- NCR 53C895
bus 0, slot 3, function 0 -- pkb -- NCR 53C895
bus 0, slot 4 -- ewa -- DE500-AA Network Controller
probing hose 0, PCI
probing PCI-to-ISA bridge, bus 1
bus 0, slot 2 -- vga -- DEC PowerStorm
bus 0, slot 15 -- dqa -- Acer Labs M1543C IDE
bus 0, slot 15 -- dqb -- Acer Labs M1543C IDE
starting drivers                                           ❹
```

SRM Power-Up Sequence

❶ The primary CPU prints a message indicating that it is running the console. Starting with this message, the power-up display is sent to any console terminal, regardless of the state of the **console** environment variable.

If console is set to **graphics**, the display from this point on is saved in a memory buffer and displayed on the VGA monitor after the PCI buses are sized and the VGA device is initialized.

❷ The memory size is determined and memory is tested.

❸ The I/O subsystem is probed and I/O devices are reported. I/O adapters are configured.

❹ Device drivers are started.

Continued on next page

Example 3–2 SRM Power-Up Display (Continued)

```
entering idle loop
initializing keyboard
starting console on CPU 1                                ❺
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 1 speed is 2.00 ns (500MHz)
create powerup
entering idle loop
starting console on CPU 2
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 2 speed is 2.00 ns (500MHz)
create powerup
starting console on CPU 3
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 3 speed is 2.00 ns (500MHz)
create powerup
Memory Testing and Configuration Status                  ❻
  Array       Size       Base Address
---------  ----------  ----------------
    0       256Mb     0000000060000000
    1       512Mb     0000000040000000
    2       256Mb     0000000070000000
    3      1024Mb     0000000000000000

     2048 MB of System Memory
Testing the System
Testing the Disks (read only)
Testing the Network
initializing GCT/FRU at offset 192000
AlphaServer ES40 Console V5.4-5528, built on Feb  1 1999 at
01:43:35                                                 ❼
P00>>>
```

SRM Power-Up Sequence (Continued)

❺ The console is started on the secondary CPUs. The example shows a four-processor system.

❻ Various diagnostics are performed.

❼ Systems running UNIX or OpenVMS display the SRM console banner and the prompt, P*nn*>>>. The number *n* indicates the primary processor. In a multiprocessor system, the prompt could be P00>>>, P01>>>, P02>>>, or P03>>>. From the SRM prompt, you can boot the UNIX or OpenVMS operating system.

NOTE: *If the console requires the heap to be expanded, it restarts. See Section 3.3.3.*

3.3.3 Resizing SRM Console Heap

The SRM console allocates enough memory for most configurations. If options were installed that require more memory than the SRM console has allocated, the console dynamically resizes itself to provide additional memory to support the configuration.

The following crash/reboot cycle can occur several times until the console has allocated enough memory. An abbreviated example of the output to a serial console screen is shown in Example 3–3.

1. The console powers up.
2. Drivers try to allocate more “heap space” (space for more memory) but cannot.
3. The console displays a message similar to the following:

```
CPU0: insufficient dynamic memory for a request of 4592 bytes
Console heap space will be automatically increased in size by
64KB
```

4. The console takes an exception.
5. The console allocates more heap space and restarts with memory set to the required size.

After the console completes its final reinitialization, the console banner is displayed, followed by the console prompt. Enter the **show heap_expand** command to verify that the console has allocated more memory. You can then boot the operating system. No other action is required, and the crash/reboot cycle should not occur again.

If the configuration is subsequently changed, enter the following command to reset the heap space to its default before you boot the system:

```
P00>>> set heap_expand none
```

Resizing may or may not occur again, depending on whether the console requires additional heap space.

Example 3–3 Memory Resize Crash/Reboot Cycle

```
initialized idle PCB
initializing semaphores
initializing heap
initial heap 200c0
memory low limit = 15e000
heap = 200c0, 17fc0
initializing driver structures
initializing idle process PID
initializing file system
initializing hardware
initializing timer data structures
lowering IPL
CPU 0 speed is 500 MHz
create dead_eater
create poll
create timer
create powerup
access NVRAM
Memory size 2048 MB
testing memory
......
probe I/O subsystem
probing hose 1, PCI
bus 0, slot 1 -- pka—NCR 53C895
bus 0, slot 3 -- mca—DEC PCI MC
bus 0, slot 4 -- mcb—DEC PCI MC
.
.
.
starting drivers
entering idle loop
initializing keyboard
starting console on CPU 1
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 1 speed is 500 MHz
create powerup
.
.
.
Memory Testing and Configuration Status
  Array       Size       Base Address
---------  ----------  ----------------
    0        512Mb     0000000040000000
    1       1024Mb     0000000000000000
    2        256Mb     0000000060000000
    3        256Mb     0000000070000000
```

```
     2048 MB of System Memory
Testing the System
CPU0: insufficient dynamic memory for a request of 4592 bytes
Console heap space will be automatically increased in size by 64KB

   PID       bytes  name
-------- ---------- ----
00000000      27360 ????
00000001      23424 idle
00000002        800 dead_eater
00000003        800 poll
00000004        800 timer
00000005     499584 powerup
00000031     129536 pwrup_diag
00000013        896 ????
00000016       1056 ????
00000026        128 ????
00000017        512 ????
00000006       2880 tt_control
00000007        800 mscp_poll
00000008        800 dup_poll
00000012       2336 shell_0
0000000A      13920 ????
0000000D      13920 ????
00000010      13920 ????
0000000B       2336 shell_1
0000000E       2336 shell_2
00000011       2336 shell_3
00000029        128 ????
00000014        992 rx_ewa0
00000018        512 ????
0000001F        992 rx_eib0
0000001C        992 rx_eia0
0000001D        160 ????
00000025       1024 rx_eie0
00000021        992 rx_eic0
0000002C        160 ????
00000023        992 rx_eid0
0000002F        160 ????
00000024        128 ????
00000028        992 rx_eif0
00000027        160 ????
0000002B       1024 rx_eig0
0000002E        992 rx_eih0
0000002D        160 ????
0000002A        128 ????
00000030        128 ????
00000038       2080 ????
0000003D      22848 sh_cmdsub
00000040       5696 show
00000041        800 setmode
```

```
SYSFAULT CPU0 - pc = 0014faac
exception context saved starting at 001FD7B0
GPRs:
  0: 00000000 00048FF8  16: 00000000 0000001E
  1: 00000000 00150C80  17: 00000000 EFEFEFC8
  2: 00000000 001202D0  18: 00000000 001FD2F8
  3: 00000000 000011F0  19: 00000000 00000025
  4: 00000000 0010C7B8  20: 00000801 FC000000
  5: 00000000 00000020  21: 00000000 0008A8B0
  6: 00000000 00000000  22: 00000000 0010ACB8
  7: 00000000 00038340  23: 00000000 00000001
  8: 00000000 00000000  24: 00000000 00000000
  9: 00000000 00000000  25: 00000000 00000001
 10: 00000000 00000000  26: 00000000 0014FAAC
 11: 00000000 3FFFF520  27: 00000000 00150C90
 12: 00000000 001254D0  28: 00000000 00038518
 13: 00000000 0013BB20  29: 00000000 001FD8F0
 14: 00000000 0010C7C0  30: 00000000 001FD8F0
 15: 00000000 00000001
dump of active call frames:
PC  =  0014FAAC
PD  =  001202D0
FP  =  001FD8F0
SP  =  001FD7B0

.
.
.
initialized idle PCB
initializing semaphores
initializing heap
initial heap 200c0
memory low limit = 15e000
heap = 200c0, 17fc0
initializing driver structures
initializing idle process PID
initializing file system
initializing hardware
initializing timer data structures
lowering IPL
CPU 0 speed is 500 MHz
create dead_eater
create poll
create timer
create powerup
access NVRAM
Memory size 2048 MB
testing memory
......
probe I/O subsystem
probing hose 1, PCI
bus 0, slot 1 -- pka—NCR 53C895
bus 0, slot 3 -- mca—DEC PCI MC
```

```
.
.
.
bus 0, slot 15 -- dqb—Acer Labs M1543C IDE
starting drivers
entering idle loop
initializing keyboard
starting console on CPU 1
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 1 speed is 500 MHz
create powerup
.
.
.
Memory Testing and Configuration Status
  Array       Size       Base Address
---------  ----------  ----------------
    0        512Mb     0000000040000000
    1       1024Mb     0000000000000000
    2        256Mb     0000000060000000
    3        256Mb     0000000070000000

     2048 MB of System Memory
Testing the System
Testing the Disks (read only)
Testing the Network
Partition 0, Memory base: 000000000, size: 080000000
initializing GCT/FRU at offset 1dc000
AlphaServer ES40 Console V5.5-3059, built on May 14 1999 at 01:57:42

P00>>>show heap_expand
heap_expand         64KB
P00>>>
```

3.3.4 SRM Console Event Log

The SRM console event log helps you troubleshoot problems that do not prevent the system from coming up to the SRM console. The console event log consists of status messages received during power-up self-tests.

Example 3–4 Sample Console Event Log

```
>>> more el
*** Error - CPU 1 failed powerup diagnostics ***
  Secondary start error
EV6 BIST         = 1
STR status       = 1
CSC status       = 1
PChip0 status    = 1
PChip1 status    = 1
DIMx status      = 0
TIG Bus status   = 1
DPR status       = 0
CPU speed status = 0
CPU speed        = 0
Powerup time     = 00-00-00 00:00:00
CPU SROM sync    = 0

*** Error - Fan 1 failed ***

*** Error - Fan 2 failed ***
```

If problems occur during power-up, error messages indicated by asterisks (***) may be embedded in the console event log. To display the console event log one screen at a time, use the **more el** command.

Example 3–4 shows a console event log that shows errors. The console reported that CPU 1 did not power up and fans 1 and 2 failed.

3.3.5 AlphaBIOS Startup Screens

If the system is running the Windows NT operating system, the SRM console loads and starts the AlphaBIOS console. An initialization screen similar to Example 3–5 is displayed on the VGA monitor. The initialization includes a memory test that is displayed to the screen. Once AlphaBIOS initialization is complete, an AlphaBIOS boot screen similar to Example 3–6 is displayed.

Example 3–5 AlphaBIOS Initialization Screen

```
                      AlphaBIOS 5.68

  Alpha Processor and System Information:
       System:       AlphaServer ES40
       Processor:    Alpha 21264, 500 MHz
       Memory:       256 MB

  Alpha Processor(s) Status:
       Processor 0 Running
       Processors 1, 2, 3 Ready

  SCSI Controller Initialization...

  Initialize ATAPI #0...
       Device: CD-ROM SCSI ID:0 TOSHIBA CD-ROM XM62028 1110

 F2=Setup   PAUSE=Pause Display   ESC=Bypass Network Init
```

PKO950

Example 3–6 AlphaBIOS Boot Screen

```
                          AlphaBIOS 5.68

Please select the operating system to start:

   Windows NT Server 4.00


Use ↓ and ↑ to move the highlight to your choice.
Press Enter to choose.



   COMPAQ      AlphaServer


                                         Press <F2> to enter SETUP
```

PK0949

3.4 Power-Up Error Messages

Error messages at power-up may be displayed by the RMC, SROM, and SRM. A few SROM messages are announced by beep codes.

3.4.1 SROM Messages with Beep Codes

Table 3–1 Error Beep Codes

Beep Code	Associated Messages	Meaning
1	Jump to Console	SROM code has completed execution. System jumps to SRM console. SRM messages should start to be displayed. If no SRM messages are displayed, it may indicate corrupted firmware. See Section 3.4.2.
1-3		VGA monitor not plugged in. The first beep is a long beep.
1-1-4	ROM err	The ROM err message is displayed briefly, then a single beep is emitted, and Jump to Console is displayed. The SROM code is unable to load the console code; a flash ROM header area or checksum error has been detected. See Section 3.4.2.
2-1-2	Cfg ERR *n* Cfg ERR s	Configuration error on CPU *n* (*n* is 0, 1, 2, or 3) or a system configuration error. The system will still power up.
1-2-4	BC error CPU error BC bad	Backup cache (B-cache) error. Indicates a bad CPU.
1-3-3	No mem	No usable memory detected. Some memory DIMMs may not be properly seated or some DIMM sets may be faulty. See Section 3.4.3.

A few SROM error messages that appear on the operator control panel are announced by audible error beep codes, an indicated in Table 3–1. For example, a 1-1-4 beep code consists of one beep, a pause (indicated by the hyphen), one beep, a pause, and a burst of four beeps. This beep code is accompanied by the message “ROM err.”

Related messages are also displayed on the console terminal if the console device is connected to the serial line and the SRM **console** environment variable is set to **serial**.

3.4.2 Checksum Error

If Jump to Console is the last message displayed on the OCP, the console firmware may have become corrupted. When the system detects the error, it attempts to load the fail-safe loader (FSL) program so that you can load new console firmware images.

Example 3–7 Checksum Error and Fail-Safe Load

```
Loading console
Console ROM checksum error                                    ❶
Expect: 00000000.000000FE
Actual: 00000000.000000FF
XORval: 00000000.00000001
Loading program from floppy                                   ❷
Code execution complete (transfer control)

OpenVMS PALcode V1.3-3, Digital UNIX PALcode V1.4-2

starting console on CPU 0                                     ❸
.
.
starting drivers
entering idle loop
.
.
.
P00>>> Boot update_cd                                         ❹

OpenVMS PALcode V1.3-3, Digital UNIX PALcode V1.4-2

starting console on CPU 0                                     ❺
.
.
starting drivers
entering idle loop
.
.
.
```

```
          ***** Loadable Firmware Update Utility *****      ❻
-------------------------------------------------------------
Function    Description
------------------------------------------------------------
Display     Displays the system's configuration table.
Exit        Done exit LFU (reset).
List        Lists the device, revision, firmware name, and
            update revision.
Readme      Lists important release information.
Update      Replaces current firmware with loadable data
            image.
Verify      Compares loadable and hardware images.
? or Help   Scrolls this function table.
--------------------------------------------------------------
UPD> update                                                 ❼
```

The sequence shown in Example 3–7 is as follows:

❶ The system detects the checksum error and writes a message to the console screen.

❷ The system attempts to automatically load the FSL program from the floppy drive.

❸ As the FSL program is initialized, messages similar to the console power-up messages are displayed. This example shows the beginning and ending messages.

❹ At the P00>>> console prompt, boot the Loadable Firmware Update Utility (LFU) from the Alpha Systems Firmware CD (shown in the example as the variable *update_cd*).

❺ As the LFU program is initialized, messages similar to the console power-up messages are displayed. This example shows a few of the messages.

❻ After the "entering idle loop" message, the banner for the Loadable Firmware Update Utility is displayed.

❼ At the UPD> prompt, enter the **update** command to load the new console firmware images.

NOTE: *For more information on the LFU, see the Firmware Updates Web site: http://ftp.digital.com/pub/digital/Alpha/firmware/*

3.4.3 No MEM Error

If the SROM code cannot find any usable memory, a 1-3-3 beep code is issued (one beep, a pause, a burst of three beeps, a pause, and another burst of three beeps), and the message "No MEM" is displayed on the OCP. The system does not come up to the console program. This error indicates missing or bad DIMMs.

The OCP and console terminal display text similar to the following:

```
Failed  M:1 D:2               ❶
Failed  M:1 D:1
Failed  M:0 D:2
Failed  M:0 D:1
Incmpat M:1 D:4               ❷
Incmpat M:1 D:3
Incmpat M:0 D:4
Incmpat M:0 D:3
Missing M:3 D:2               ❸
Illegal M:2 D:2               ❹
No usable memory detected
```

❶ Indicates failed DIMMs. M identifies the MMB; D identifies the DIMM. In this line, DIMM 2 on MMB1 failed.

❷ Indicates that some DIMMs in this array are mismatched. All DIMMs in the affected array are marked as incompatible (incmpat).

❸ Indicates that a DIMM in this array is missing. All missing DIMMs in the affected array are marked as missing.

❹ Indicates that the DIMM data for this array is unreadable. All unreadable DIMMs in the affected array are marked as illegal.

See Chapter 6 for memory configuration rules.

3.4.4 RMC Error Messages

Table 3–2 lists the fatal error messages that could potentially be displayed on the OCP by the remote management console during power-up. Most fatal error messages prevent the system from completing power-up. The warning messages listed in Table 3–3 require prompt attention but might not prevent the system from completing power-up or booting the operating system.

Table 3–2 RMC Fatal Error Messages

Message	Meaning
AC loss	No AC power to the system.
CPU*n* failed	CPU failed. "n" is 0, 1, 2, or 3.
VTERM failed	No VTERM voltage to CPUs.
CTERM failed	No CTERM voltage to CPUs.
Fan5, 6 failed	Main fan (6) and redundant fan (5) failed.
OverTemp failure	System temperature has passed the high threshold.
No CPU in slot 0	Configuration requires that a CPU be installed in slot 0.
CPU door opened	System card cage cover off. Reinstall cover.
TIG error	Code essential to system operation is not loaded and/or running or TIG flash is corrupt..
Mixed CPU types	Different types of CPU are installed. Configuration requires that all CPUs be the same type.
Bad CPU ROM data	Invalid data in EEROM on the CPU.

NOTE: *The"* CPUn *failed" message does not necessarily prevent the completion of power-up. If the system finds a good CPU, it continues the power-up process.*

Table 3–3 RMC Warning Messages

Message	Meaning
PS*n* failed	Power supply failed. “n” is 0, 1, or 2.
OverTemp Warning	System temperature is near the high threshold.
Fan*n* failed	Fan failed. “n” is 0 through 6.
PCI door opened	Cover to PCI card cage is off. Reinstall cover.
Fan door opened	Cover to main fan area (fans 5 and 6) is off. Reinstall cover.
3.3V bulk warn	Power supply voltage over or under threshold.
5V bulk warn	Power supply voltage over or under threshold.
12V bulk warn	Power supply voltage over or under threshold.
–12V bulk warn	Power supply voltage over or under threshold.
VTERM warn	Voltage regulator over or under threshold.
CTERM warn	Voltage regulator over or under threshold.
CPU*n* VCORE warn	CPU core voltage over or under threshold. “n” is 0, 1, 2, or 3.
CPU*n* VIO warn	I/O voltage on CPU over or under threshold. “n” is 0, 1, 2, or 3.

3.4.5 SROM Error Messages

The SROM power-up identifies errors that may or may not prevent the system from coming up to the console. It is possible that these errors may prevent the system from successfully booting the operating system. Errors encountered during SROM power-up are displayed on the OCP. Some errors are also displayed on the console terminal screen if the console output is set to serial.

Table 3–4 lists the SROM error messages.

Table 3–4 SROM Error Messages

Code	SROM Message	OCP Message
FD	PCI data path error	PCI Err
FA	No usable memory detected	No Mem
EF	Bcache data lines test error	BC Error
EE	Bcache data march test error	BC Error
ED	Bcache address test error	BC Error
EC	CPU parity detection error	CPU Err
EB	CPU ECC detection error	CPU Err
EA	Bcache ECC data lines test error	BC Error
E9	Bcache ECC data march test error	BC Error
E8	Bcache TAG lines test error	BC Error
E7	Bcache TAG march test error	BC Error
E6	Console ROM checksum error	ROM Err
E5	Floppy driver error	Flpy Err
E4	No real-time clock (TOY)	TOY Err
E3	Memory data path error	Mem Err
E2	Memory address line error	Mem Err
E1	Memory pattern error	Mem Err
E0	Memory pattern ECC error	Mem Err
7F	Configuration error on CPU #3	CfgERR 3

Table 3–4 SROM Error Messages (Continued)

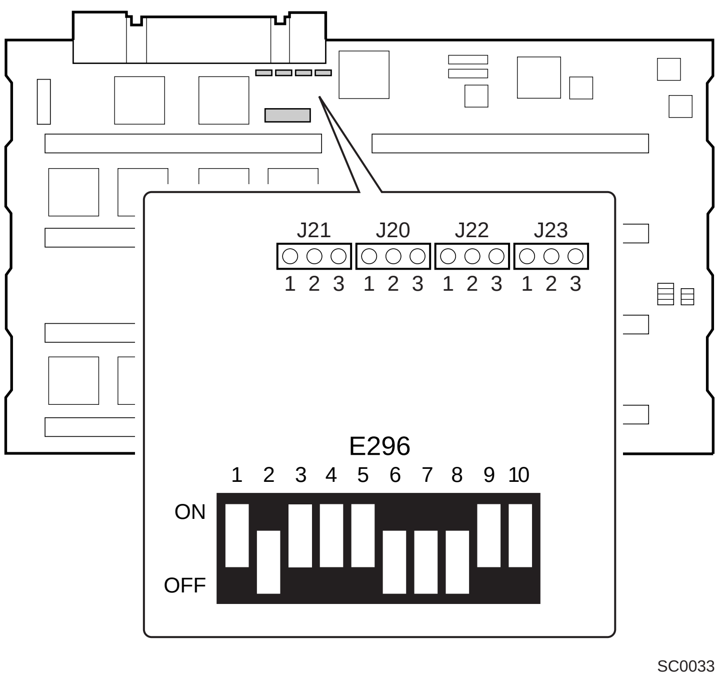
Code	SROM Message	OCP Message
7E	Configuration error on CPU #2	CfgERR 2
7D	Configuration error on CPU #1	CfgERR 1
7C	Configuration error on CPU #0	CfgERR 0
7B	Bcache failed on CPU #3 error	BC Bad 3
7A	Bcache failed on CPU #2 error	BC Bad 2
79	Bcache failed on CPU #1 error	BC Bad 1
78	Bcache failed on CPU #0 error	BC Bad 0
77	Memory thrash error on CPU #3	MtrERR 3
76	Memory thrash error on CPU #2	MtrERR 2
75	Memory thrash error on CPU #1	MtrERR 1
74	Memory thrash error on CPU #0	MtrERR 0
73	Starting secondary on CPU #3 error	RCPU 3 E
72	Starting secondary on CPU #2 error	RCPU 2 E
71	Starting secondary on CPU #1 error	RCPU 1 E
70	Starting secondary on CPU #0 error	RCPU 0 E
6F	Configuration error with system	CfgERR S

NOTE: *The code numbers shown in the Code column are displayed in place of OCP or SROM messages if the SROM flash is invalid.*

3.5 Forcing a Fail-Safe Floppy Load

Under some circumstances, you may need to force the activation of the FSL. For example, if you install a system motherboard that has an older version of the firmware than your system requires, you may not be able to bring up the SRM console. In that case you need to force a floppy load so that you can update the SRM firmware.

Figure 3–2 Function Jumpers



1. Turn off the system. Unplug the power cord from each power supply and wait for the 5V AUX indicators to extinguish.
2. Remove enclosure covers (tower and pedestal) or the front bezel (rackmount) to access the system chassis. See Chapter 8 for illustrations.
3. Remove the fan cover and the system card cage cover to gain access to the system motherboard. See Chapter 8 for illustrations.
4. Remove MMB 1 (closest to the PCI backplane) so that you can access the function jumpers.
5. Locate the J22 function jumper on the system motherboard. See Figure 3–2.
6. Enable the fail-safe loader by moving the J22 jumper from pins 1 and 2 to pins 2 and 3.

NOTE: *The J20 and J23 function jumpers must be in their default positions over pins 1 and 2.*

---

7. Replace the chassis covers and enclosure covers. Plug in the power supplies.
8. Insert the Firmware Update Utility diskette into the floppy drive, and insert the update CD into the CD-ROM drive.
9. Power up the system and check the control panel display for progress messages.
10. At the P00>>> prompt, boot the update CD. Enter **update** at the UPD> prompt and press Return. Enter **yes** at the "Confirm update" prompt.
11. After the update is complete, turn off the system and unplug the power supplies.
12. Place J22 over pins 1 and 2.
13. Replace MMB 1.
14. Replace the chassis covers and enclosure covers, plug in the power supplies, and power up the system.

NOTE: *For more information on the LFU, see the Firmware Updates Web site: http://ftp.digital.com/pub/digital/Alpha/firmware/*

3.6 Updating the RMC

Under certain circumstances, the RMC will not function. If the problem is caused by corrupted RMC flash ROM, you need to update RMC firmware.

The RMC will not function if:

- No AC power is provided to any of the power supplies.
- DPR does not pass its self-test (DPR is corrupted).
- RMC flash ROM is corrupted.

If the RMC is not working, the control panel displays the following message:

```
Bad RMC flash
```

The SRM console also sends a message to the terminal screen:

```
*** Error - RMC detected power up error - RMC Flash corrupted ***
```

You can update the remote management console firmware from flash ROM using the LFU.

1. Load the update medium.

2. At the UPD> prompt, exit from the update utility, and answer **y** to the manual update prompt. Enter **update RMC** to update the firmware.

```
UPD> exit

Do you want to do a manual update [y/(n)] y

***** Loadable Firmware Update Utility *****
-------------------------------------------------------------
Function    Description
-------------------------------------------------------------
Display     Displays the system’s configuration table.
Exit        Done exit LFU (reset).
List        Lists the device, revision, firmware name, and
            update revision.
Readme      Lists important release information.
Update      Replaces current firmware with loadable data
            image.
Verify      Compares loadable and hardware images.
? or Help   Scrolls this function table.
-----------------------------------------------------------

UPD> update RMC
.
.
.
```

NOTE: *For more information on the LFU, see the Firmware Updates Web site: http://ftp.digital.com/pub/digital/Alpha/firmware/*

Chapter 4
SRM Console Diagnostics

This chapter describes troubleshooting with the SRM console.

The SRM console firmware contains ROM-based diagnostics that allow you to run system-specific or device-specific exercisers. The exercisers run concurrently to provide maximum bus interaction between the console drivers and the target devices.

Run the diagnostics by using commands from the SRM console. To run the diagnostics in the background, use the background operator "&" at the end of the command. Errors are reported to the console terminal, the console event log, or both.

If you are not familiar with the SRM console, see the *Compaq AlphaServer ES40 User Interface Guide.*

NOTE: *If you are running a Windows NT system, you need to switch from AlphaBIOS to SRM to run SRM console firmware diagnostics.*

4.1 Diagnostic Command Summary

Diagnostic commands are used to test the system and help diagnose failures. Table 4–1 gives a summary of the SRM diagnostic commands and related commands. See Chapter 6 for a list of SRM environment variables, and see Appendix A for a list of SRM commands most commonly used for the ES40 system.

Table 4–1 Summary of Diagnostic and Related Commands

Command	Function
buildfru	Initializes I^2Cbus EEPROM data structures for the named FRU.
cat el	Displays the console event log. Same as **more el**, but scrolls rapidly. The most recent errors are at the end of the event log and are visible on the terminal screen.
clear_error	Clear errors logged in the FRU EEPROMs as reported by the **show error** command.
crash	Forces a crash dump at the operating system level.
deposit	Writes data to the specified address of a memory location, register, or device.
examine	Displays the contents of a memory location, register, or device.
exer	Exercises one or more devices by performing specified read, write, and compare operations.
floppy_write	Runs a write test on the floppy drive to determine whether you can write on the diskette.
grep	Searches for “regular expressions”—specific strings of characters—and prints any lines containing occurrences of the strings.
hd	Dumps the contents of a file (byte stream) in hexadecimal and ASCII.
info	Displays registers and data structures.

Table 4–1 Summary of Diagnostic and Related Commands (Continued)

Command	Function
kill	Terminates a specified process.
kill_diags	Terminates all executing diagnostics.
more el	Same as **cat el**, but displays the console event log one screen at a time.
memexer	Runs a requested number of memory tests in the background.
memtest	Tests a specified section of memory.
net -ic	Initializes the MOP counters for the specified Ethernet port.
net -s	Displays the MOP counters for the specified Ethernet port.
nettest	Runs loopback tests for PCI-based Ethernet ports. Also used to test a port on a "live" network.
set sys_serial_ num	Sets the system serial number, which is then propagated to all FRUs that have EEPROMs.
show error	Reports errors logged in the FRU EEPROMs.
show fru	Displays information about field replaceable units (FRUs), including CPUs, memory DIMMs, and PCI cards.
show_status	Displays the progress of diagnostic tests. Reports one line of information for each executing diagnostic.
sys_exer	Exercises the devices displayed with the **show config** command
sys_exer -lb	Runs console loopback tests for the COM2 serial port and the parallel port during the **sys_exer** test sequence.
test	Verifies the configuration of the devices in the system.
test -lb	Runs loopback tests for the COM2 serial port and the parallel port in addition to verifying the configuration of devices.

4.2 buildfru

The buildfru command initializes I^2C bus EEPROM descriptive data structures for the named FRU and initializes its SDD and TDD error logs. This command uses data supplied on the command line to build the FRU descriptor. Buildfru is used by Manufacturing, FRU repair operations, or Field Service.

Example 4–1 buildfru

```
P00>>> buildfru smb0.mmb0.dim1 54-24941-EA NI90200100        ❶
P00>>> buildfru smb0.cpu0 30-30158-05.AX05 NI94060554 Compaq❷
P00>>> buildfru -s smb0.mmb0.dim1 80 45                      ❸
P00>>> buildfru -s smb0.mmb0.dim1 80 47 46 45 44 43 42 41    ❹
```

❶ Building of the FRU descriptor on a DIMM, passing a part number and a serial number

❷ Building of the FRU descriptor on a CPU, passing a part number, serial number, and miscellaneous string

❸ Building of the FRU descriptor on a DIMM with the **-s** qualifier, pass offset 80, and value of 45

❹ Building of the FRU descriptor on a DIMM with the **-s** qualifier, pass offset 80, and many sequential data bytes

The **buildfru** command is used for several purposes:

- By Manufacturing to build a FRU table containing a description of each FRU in the system
- By FRU repair operations for initializing good stocking spares
- By Field Service to make any FRU descriptor adjustments required by the customer.

The information supplied on the **buildfru** command line includes the console name for the FRU, part number, serial number, model number, and optional information. The **buildfru** command facilitates writing the FRU information to the EEPROM on the device.

Use the **show fru** command to display the FRU table created with **buildfru**. Use the **show error** command to display FRUs that have errors logged to them.

Typically, you only need to use **buildfru** in Field Service if you replace a device for which the information displayed with the **show fru** command is inaccurate or missing. After replacing the device, use **buildfru** to build the new FRU descriptor.

NOTE: *Be sure to enter the FRU information carefully. If you enter incorrect information, the callout used by Compaq Analyze will not be accurate.*

Three areas of the EEPROM can be initialized: the FRU generic data, the FRU specific data, and the system specific data. Each area has its own checksum, which is recalculated any time that segment of the EEPROM is written.

When the **buildfru** command is executed, the FRU EEPROM is first flooded with zeros and then the generic data, the system specific data, and EEPROM format version information are written and checksums are updated. For certain FRUs, such as CPU modules, additional FRU "specific" data can be entered using the **-s** option. This data is written to the appropriate region, and its corresponding checksum is updated.

FRU Assembly Hierarchy

AlphaServer systems can be decomposed into a collection of FRUs. Some FRUs carry various levels of nested FRUs. For instance, the system motherboard is a FRU that carries a number of "child" FRUs. A child, such as a memory motherboard (MMB), may carry a number of its own children, DIMMs. The naming convention for FRUs represents the assembly hierarchy.

The following is the general form of a FRU name:

<fru*n*>[.<fru*n*>[.<fru*n*>]]

The *fru* is a placeholder for the appropriate FRU type at that level and *n* is the number of that FRU instance on that branch of the system hierarchy.

Continued on next page

The ES40 FRU assembly hierarchy has three levels. The FRU types from the top to the bottom of the hierarchy are as follows:

Level	FRU Type	Meaning
First Level	SMB	System motherboard
	JIO	I/O connector module (junk I/O)
	OCP	Operator control panel
	PWR (0–2)	Power supplies
	FAN	Fans
Second Level	CPU (0–3)	CPUs
	MMB (0–3)	Memory motherboards
	CPB	PCI backplane
Third Level	DIM (1–8)	Memory DIMMs
	PCI (0–9)	PCI slots
	SBM (0–1)	SCSI backplane

To build a FRU descriptor for a lower level FRU, point back to the higher level FRUs to which it is associated. For example, to build a descriptor for a DIMM, point back to the MMB on which it resides and then to the system motherboard. All fields are automatically set to uppercase before writing to EEPROM. See Example 4–1.

If you enter the **buildfru** data correctly for a device that has an EEPROM to program, nothing is displayed after you enter the command. If you enter incorrect data or the device does not have an EEPROM to program, an error message similar to the following is displayed:

```
P00>>>
P00>>> buildf fan4 54-12345-01.a001 ay84412345
Device FAN4 does not support setting FRU values
P00>>>
```

Syntax

buildfru (<fru_name> <part_num> <serial_num> [<misc> [<other>]]
or
-s <fru_name> <offset> <byte> [<byte>...])

Arguments

<fru_name> Console name for this FRU. This name reflects the position of the FRU in the assembly hierarchy.

<part_num> The FRU's 2-5-2.4 part number. This ASCII string should be 16 characters (extra characters are truncated). This field should not contain any embedded spaces. If a space must be inserted, enclose the entire argument string in double quotes. This field contains the FRU revision, and in some cases an embedded space is allowed between the part number and the revision.

<serial_num> The FRU's serial number. This ASCII string must be 10 characters (extra characters are truncated). The manufacturing location and date are extracted from this field.

<misc> The FRU's model name or number or the common name for the FRU. This ASCII string may be up to 10 characters (extra characters are truncated). This field is optional, unless **<alias>** is specified.

<other> The FRU's Compaq alias number, if one exists. This ASCII string may be up to 16 characters (extras are truncated). This field is optional.

<offset> The beginning byte offset (0–255 hex) within this FRU's EEPROM, where the following supplied data bytes are to be written.

<byte>... The data bytes to be written. At least one data byte must be supplied after the offset.

Options

-s Writes raw data to the EEPROM. This option is typically used to apply any FRU specific data.

4.3 cat el and more el

The cat el and more el commands display the contents of the console event log.

In Example 4–2, the console reports that CPU 1 did not power up and fans 1 and 2 failed.

Example 4–2 more el

```
>>> more el
*** Error - CPU 1 failed powerup diagnostics ***                ➊
  Secondary start error
EV6 BIST         = 1
STR status       = 1
CSC status       = 1
PChip0 status    = 1
PChip1 status    = 1
DIMx status      = 0
TIG Bus status   = 1
DPR status       = 0
CPU speed status = 0
CPU speed        = 0
Powerup time     = 00-00-00 00:00:00
CPU SROM sync    = 0

*** Error - Fan 1 failed ***                                     ➋

*** Error - Fan 2 failed ***
```

❶ CPU 1 failed.

❷ Fan 1 and Fan 2 failed.

Status and error messages are logged to the console event log at power-up, during normal system operation, and while running system tests. Standard error messages are indicated by asterisks (***).

When **cat el** is used, the contents of the console event log scroll by. Use the Ctrl/S key combination to stop the screen from scrolling, and use Ctrl/Q to resume scrolling.

The **more el** command allows you to view the console event log one screen at a time.

Syntax

cat el or **more el**

4.4 clear_error

The clear_error command clear errors logged in the FRU EEPROMs as reported by the show error command.

Example 4–3 clear_error

```
P00>>> clear_error smb0        ➊
P00>>>

P00>>> clear_error all         ➋
P00>>>
```

➊ Clears all errors logged in the FRU EEPROM on the system motherboard (SMB0).

➋ Clears all errors logged to all FRU EEPROMs in the system

The **clear_error** command clears TDD, SDD, and checksum errors. Hardware failures and unreadable EEPROM errors are not cleared. See Table 4–2.

Syntax

clear_error <*fruname*>	Clears all errors logged to a specific FRU. *Fruname* is the name of the specified FRU. If you do not specify a FRU, you must use **clear_error all** to clear errors.
clear_error all	Clears all errors logged to all system FRUs.

See the **show error** command for information on the types of errors that might be logged to the FRU EEPROMs.

4.5 crash

The SRM crash command forces a crash dump to the selected device for UNIX and OpenVMS systems.

```
P00>>> crash

CPU 0 restarting

DUMP: 19837638 blocks available for dumping.
DUMP: 118178 wanted for a partial compressed dump.
DUMP: Allowing 2060017 of the 2064113 available on 0x800001
device string for dump = SCSI 1 1 0 0 0 0 0.
DUMP.prom: dev SCSI 1 1 0 0 0 0 0, block 2178787
DUMP: Header to 0x800001 at 2064113 (0x1f7ef1)
device string for dump = SCSI 1 1 0 0 0 0 0.
DUMP.prom: dev SCSI 1 1 0 0 0 0 0, block 2178787
DUMP: Dump to 0x800001: .......: End 0x800001
device string for dump = SCSI 1 1 0 0 0 0 0.
DUMP.prom: dev SCSI 1 1 0 0 0 0 0, block 2178787
DUMP: Header to 0x800001 at 2064113 (0x1f7ef1)
succeeded

halted CPU 0

halt code = 5
HALT instruction executed
PC = fffffc0000568704
P00>>>
```

Use the **crash** command when the system has hung and you are able to halt it with the Halt button or the RMC **halt in** command. The **crash** command restarts the operating system and forces a crash dump to the selected device.

- See the *OpenVMS Alpha System Dump Analyzer Utility Manual* for information on how to interpret OpenVMS crash dump files.
- See the *Guide to Kernel Debugging* for information on using the Tru64 UNIX Krash Utility.

4.6 deposit and examine

The deposit command writes data to the specified address of a memory location, register, or device. The examine command displays the contents of a memory location, register, or a device.

Example 4–4 deposit and examine

deposit

```
P00>>> dep -b -n 1ff pmem:0 0                                ❶
P00>>> d -l -n 3 vmem:1234 5                                 ❷
P00>>> d -n 8 r0 ffffffff                                    ❸
P00>>> d -l -n 10 -s 200 pmem:0 8                            ❹
P00>>> d -l pmem:0 0                                         ❺
P00>>> d + ff                                                ❻
P00>>> d scbb 820000                                         ❼
```

examine

```
P00>>> e dpr:34f0 -l -n 5                                    ❶
dpr:             34F0 00000000
dpr:             34F4 00000000
dpr:             34F8 00000000
dpr:             34FC 00000000
dpr:             3500 204D5253
dpr:             3504 352E3558
P00>>>
```

Deposit

The **deposit** command stores data in the location specified. If no options are given, the system uses the options from the preceding **deposit** command.

If the specified value is too large to fit in the data size listed, the console ignores the command and issues an error. If the data is smaller than the data size, the higher order bits are filled with zeros.

In Example 4–4:

❶ Clear first 512 bytes of physical memory

❷ Deposit 5 into four longwords starting at virtual memory address 1234.

❸ Load GPRs R0 through R8 with -1.

❹ Deposit 8 in the first longword of the first 17 pages in physical memory.

❺ Deposit 0 to physical memory address 0.

❻ Deposit FF to physical memory address 4.

❼ Deposit 820000 to SCBB.

Examine

The **examine** command displays the contents of a memory location, a register, or a device.

If no options are given, the system uses the options from the preceding **examine** command. If conflicting address space or data sizes are specified, the console ignores the command and issues an error.

For data lengths longer than a longword, each longword of data should be separated by a space.

In Example 4–4:

❶ Examine the DPR starting at location 34f0 and continuing through the next 5 locations, and display the data size in longwords.

Syntax

deposit [-{b,w,l,q,o,h}] [-{n *value, s value*}] [*space*:] *address data*

examine [-{b,w,l,q,o,h}] [-{n *value, s value*}] [*space:*] *address*

Continued on next page

-b	Defines data size as byte.
-w	Defines data size as word.
-l (default)	Defines data size as longword.
-q	Defines data size as quadword.
-o	Defines data size as octaword.
-h	Defines data size as hexword.
-d	Instruction decode (examine command only)
-n *value*	The number of consecutive locations to modify.
-s *value*	The address increment size. The default is the data size.
dev_name	Device name (address space) of the device to access. Device names are:

dpr	Dual-port RAM. See Appendix C for the DPR address layout.
eerom	Nonvolatile ROM used for EV storage.
fpr	Floating-point register set; name is F0 to F31. Alternatively, can be referenced by name.
gpr	General register set; name is R0 to R31. Alternatively, can be referenced by name.
ipr	Internal processor registers. Alternatively, some IPRs can be referenced by name.
pcicfg	PCI configuration space.
pciio	PCI I/O space.
pcimem	PCI memory space
pt	The PALtemp register set; name is PT0 to PT23.
pmem	Physical memory (default).
vmem	Virtual memory.

offset	Offset within a device to which data is deposited.
data	Data to be deposited.

Symbolic forms can be used for the address. They are:

pc The program counter. The address space is set to GPR.

+ The location immediately following the last location referenced in a **deposit** or **examine** command. For physical and virtual memory, the referenced location is the last location plus the size of the reference (1 for byte, 2 for word, 4 for longword). For other address spaces, the address is the last referenced address plus 1.

- The location immediately preceding the last location referenced in a **deposit** or **examine** command. Memory and other address spaces are handled as above.

***** The last location referenced in a **deposit** or **examine** command.

@ The location addressed by the last location referenced in a **deposit** or **examine** command.

4.7 exer

The exer command exercises one or more devices by performing specified read, write, and compare operations. Typically exer is run from the built-in console script. Advanced users may want to use the specific options described here. Note that running exer on disks can be destructive.

Optionally, **exer** reports performance statistics:

- A read operation reads from a device that you specify into a buffer.
- A write operation writes from a buffer to a device that you specify.
- A compare operation compares the contents of the two buffers.

The **exer** command uses two buffers, buffer1 and buffer2, to carry out the operations. A read or write operation can be performed using either buffer. A compare operation uses both buffers.

Example 4–5 exer

```
P00>>> exer dk*.* -p 0 -secs 36000
```

Read SCSI disks for the entire length of each disk. Repeat this until 36000 seconds, 10 hours, have elapsed. All disks will be read concurrently. Each block read will occur at a random block number on each disk.

```
P00>>> exer -l 2 dka0
```

Read block numbers 0 and 1 from device dka0.

```
P00>>> exer -sb 1 -eb 3 -bc 4 -a ’w’ -d1 ’0x5a’ dka0
```

Write hex 5a’s to every byte of blocks 1, 2, and 3. The packet size is bc * bs, 4 * 512, 2048 for all writes.

```
P00>>> ls -l dk*.*
r---   dk                  0/0              0   dka0.0.0.0.0
P00>>> exer dk*.* -bc 10 -sec 20 -m -a 'r'
dka0.0.0.0.0 exer completed
packet                                    IOs             elapsed idle
8192  3325    27238400              0    166   1360288     20    19

P00>>> exer -eb 64 -bc 4 -a '?w-Rc' dka0
```

A destructive write test over block numbers 0 through 100 on disk dka0. The packet size is 2048 bytes. The action string specifies the following sequence of operations:

1. Set the current block address to a random block number on the disk between 0 and 97. A four block packet starting at block numbers 98, 99, or 100 would access blocks beyond the end of the length to be processed so 97 is the largest possible starting block address of a packet.
2. Write a packet of hex 5a's from buffer1 to the current block address.
3. Set the current block address to what it was just prior to the previous write operation.
4. From the current block address read a packet into buffer2.
5. Compare buffer1 with buffer2 and report any discrepancies.
6. Repeat steps 1 through 5 until enough packets have been written to satisfy the length requirement of 101 blocks.

```
P00>>> exer -a '?r-w-Rc' dka0
```

A nondestructive write test with packet sizes of 512 bytes. Use this test only if the customer has a current backup of any disks being tested. The action string specifies the following sequence of operations:

1. Set the current block address to a random block number on the disk.
2. From the current block address on the disk, read a packet into buffer1.
3. Set the current block address to the device address where it was just before the previous read operation occurred.
4. Write the contents of buffer1 back to the current block address.
5. Set the current block address to what it was just prior to the previous write operation.
6. From the current block address on the disk, read a packet into buffer2.

7. Compare buffer1 with buffer2 and report any discrepancies.
8. Repeat the above steps until each block on the disk has been written once and read twice.

You can tailor the behavior of **exer** by using options to specify the following:

- An address range to test within the test device(s)
- The packet size, also known as the I/O size, which is the number of bytes read or written in one I/O operation
- The number of passes to run
- How many seconds to run
- A sequence of individual operations performed on the test devices. The qualifier is called the action string qualifier.

Syntax

exer ([-sb start_block>] [-eb end_block>] [-p pass_count>]
[-l blocks>] [-bs block_size>] [-bc block_per_io>]
[-d1 buf1_string>] [-d2 buf2_string>] [-a action_string>]
[-sec seconds>] [-m] [-v] [-delay milliseconds>]
device_name>...)

Arguments

device_name	Specifies the names of the devices or filestreams to be exercised.

Options

-sb <start_block>	Specifies the starting block number (hex) within filestream. The default is 0.
-eb <end_block>	Specifies the ending block number (hex) within filestream. The default is 0.
-p <pass_count>	Specifies the number of passes to run the exerciser. If 0, then run forever or until Ctrl/C. The default is 1.
-l <blocks>	Specifies the number of blocks (hex) to exercise. **-l** has precedence over **-eb**. If only reading, then specifying neither **-l** nor **-eb** defaults to read till eof. If writing, and neither **-l** nor **-eb** are specified then exer will write for the size of device. The default is 1.

-bs <block_size> Specifies the block size (hex) in bytes. The default is 200 (hex).

-bc <block_per_io> Specifies the number of blocks (hex) per I/O. On devices without length (tape), use the specified packet size or default to 2048. The maximum block size allowed with variable length block reads is 2048 bytes. The default is 1.

-d1 <buf1_string> String argument for eval to generate buffer1 data pattern from. Buffer1 is initialized only once before any I/O occurs. Default = all bytes set to hex 5A's.

-d2 <buf2_string> String argument for eval to generate buffer2 data pattern from. Buffer2 is initialized only once before any I/O occurs. Default = all bytes set to hex 5A's.

-a <action_string> Specifies an exerciser action string, which determines the sequence of reads, writes, and compares to various buffers. The default action string is ?r. The action string characters are:

- r Read into buffer1.
- W Write from buffer1.
- R Read into buffer2.
- W Write from buffer2.
- N Write without lock from buffer1.
- N Write without lock from buffer2.
- c Compare buffer1 with buffer2.
- \- Seek to file offset prior to last read or write.

Continued on next page

-a <action_string> (continued)	• ? Seek to a random block offset within the specified range of blocks. **exer** calls the program, random, to "deal" each of a set of numbers once. **exer** chooses a set that is a power of two and is greater than or equal to the block range. Each call to random results in a number that is then mapped to the set of numbers that are in the block range and **exer** seeks to that location in the filestream. Since **exer** starts with the same random number seed, the set of random numbers generated will always be over the same set of block range numbers. • s Sleep for a number of milliseconds specified by the delay qualifier. If no delay qualifier is present, sleep for 1 millisecond. Times as reported in verbose mode will not necessarily be accurate when this action character is used. • z Zero buffer 1 • Z Zero buffer 2 • b Add constant to buffer 1 • B Add constant to buffer 2
-sec <seconds>	Specifies to terminate the exercise after the number of seconds have elapsed. By default the exerciser continues until the specified number of blocks or passcount are processed.
-m	Specifies metrics mode. At the end of the exerciser a total throughput line is displayed.
-v	Specifies verbose mode. Data read is also written to stdout. This is not applicable on writes or compares. The default is verbose mode **off**.
-delay <millisecs>	Specifies the number of milliseconds to delay when **s** appears as a character in the action string.

4.8 floppy_write

The floppy_write script runs a write test on the floppy drive to determine whether or not you can write on the diskette. Use this script if a customer is unable to write data to the floppy. This is a destructive test, so use a blank floppy.

Example 4–6 floppy_write

```
P00>>> floppy_write
Destructive Test of the Floppy started
P00>>> show_status
ID        Program      Device        Pass  Hard/Soft Bytes Written  Bytes Read
-------- ------------ ------------ ------ --------- ------------- -----------
00000001  idle system               0     0    0     0             0
00000c37  exer_kid     dva0.0.0.100 0     0    0     6656          6656
```

The floppy_write script uses **exer** to run a write test on the floppy. The test runs in the background. Use the **show_status** command to display the progress of the test. Use the **kill** or **kill_diags** command to terminate the test.

4.9 grep

The grep command is very similar to the UNIX grep command. It allows you to search for "regular expressions"—specific strings of characters—and prints any lines containing occurrences of the strings. Using grep is similar to using wildcards.

Example 4–7 grep

```
P00>>> show fru | grep PCI
SMB0.CPB0.PCI1  0 DE500-BA Network Cont
SMB0.CPB0.PCI4  0 DEC PowerStorm
SMB0.CPB0.PCI5  0 NCR 53C895
P00>>>
```

In Example 4–7 the output of the **show fru** command is piped into **grep** (the vertical bar is the piping symbol), which filters out only lines with "PCI."

Grep supports the following metacharacters:

^	Matches beginning of line
$	Matches end of line
.	Matches any single character
[ ]	Set of characters; [ABC] matches either 'A' or 'B' or 'C'; a dash (other than first or last of the set) denotes a range of characters: [A-Z] matches any uppercase letter; if the first character of the set is '^' then the sense of match is reversed: [^0-9] matches any non-digit; several characters need to be quoted with backslash (\) if they occur in a set: '\', ']', '-', and '^'
*****	Repeated matching; when placed after a pattern, indicates that the pattern should match any number of times. For example, '[a-z][0-9]*' matches a lowercase letter followed by zero or more digits.
+	Repeated matching; when placed after a pattern, indicates that the pattern should match one or more times '[0-9]+' matches any non-empty sequence of digits.
?	Optional matching; indicates that the pattern can match zero or one times. '[a-z][0-9]?' matches lowercase letter alone or followed by a single digit.
****	Quote character; prevent the character that follows from having special meaning.

Syntax

grep ([-{c|i|n|v}] [-f <file>] [<expression>] [<file>...])

Arguments

<expression>	Specifies the target regular expression. If any regular expression metacharacters are present, the expression should be enclosed with quotes to avoid interpretation by the shell.
<file>...	Specifies the files to be searched. If none are present, then standard input is searched.

Options

-c	Print only the number of lines matched.
-i	Ignore case. By default **grep** is case sensitive.
-n	Print the line numbers of the matching lines.
-v	Print all lines that do not contain the expression.
-f <file>	Take regular expressions from a file, instead of command.

4.10 hd

The hd command dumps the contents of a file (byte stream) in hexadecimal and ASCII.

Example 4–8 hd

```
P00>>> hd -eb 0 dpr:2b00                                          ❶
block 0
00000000  48 45 4C 4C 4F FF FF FF FF FF FF FF FF FF FF FF   HELLO...........
00000010  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000020  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000030  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 3A   ...............:
00000040  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000050  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000060  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000070  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000080  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000090  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000000a0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000000b0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000000c0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000000d0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000000e0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000000f0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000100  48 45 4C 4C 4F FF FF FF FF FF FF FF FF FF FF FF   HELLO...........
00000110  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000120  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000130  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 3A   ...............:
00000140  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000150  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000160  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000170  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000180  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
00000190  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000001a0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000001b0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000001c0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000001d0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000001e0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
000001f0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
P00>>>
```

❶ Example 4–8 shows a hex dump to DPR location 2b00, ending at block 0.

Syntax

hd [-{byte|word|long|quad}] [-{sb|eb} <n>] <file>[:<offset>].

Arguments

<file>[:**<offset>**]	Specifies the file (byte stream) to be displayed.

Options

-byte	Print out data in byte sizes
-word	Print out data by word
-long	Print out data by longword
-quad	Print out data by quadword
-sb <n>	Start block
-eb <n>	End block

4.11 info

The info command displays registers and data structures. You can enter the command by itself or followed by a number (0, 1, 2, 3, or 4). If you do not specify a number, a list of selections is displayed and you are prompted to enter a selection.

Example 4–9 info 0

```
P00>>> info 0

HWRPB: 2000    MEMDSC:2d40   Cluster count: 5

Cluster: 0, Usage: Console
START_PFN: 00000000  PFN_COUNT: 00000100 PFN_TESTED: 00000000
     256 pages from 0000000000000000 to 0000000000200000

Cluster: 1, Usage: System
START_PFN: 00000100  PFN_COUNT: 0001fed1 PFN_TESTED: 00000f00
BITMAP_VA: 00000000101fe000 BITMAP_PA: 000000003ffb2000
  130769 good pages from 0000000000200000 to 000000003ffa0000

Cluster: 2, Usage: Console
START_PFN: 0001ffd1  PFN_COUNT: 0000002f PFN_TESTED: 00000000
      47 pages from 000000003ffa2000 to 0000000040000000

Cluster: 3, Usage: System
START_PFN: 00020000  PFN_COUNT: 0001fffe PFN_TESTED: 00000000
BITMAP_VA: 0000000010202000 BITMAP_PA: 000000007fffc000
  131070 good pages from 0000000040000000 to 000000007fffa000

Cluster: 4, Usage: Console
START_PFN: 0003fffe  PFN_COUNT: 00000002 PFN_TESTED: 00000000
       2 pages from 000000007fffc000 to ffffffff80000000
```

For information about the data displayed by the **info** commands, see the following documents:

- For **info 0**, **info 1**, and **info 4**, see the *Alpha System Reference Manual, Third Edition* (EY-W938E-DP), available from Digital Press, an imprint of Butterworth-Heinemann.
- For **info 2,** see the *Galaxy Console and Alpha Systems V5.0 FRU Configuration Tree Specification*.
- For **info 3,** see the *Tsunami 21272 Chipset Functional Specification*.

info 0 Displays the SRM memory descriptors as described in the *Alpha System Reference Manual*.

info 1 Displays the page table entries (PTE) used by the console and operating system to map virtual to physical memory. Valid data is displayed only after a boot operation.

info 2 Dumps the Galaxy Configuration Tree (GCT) FRU table. Galaxy is a software architecture that allows multiple instances of OpenVMS to execute cooperatively on a single computer.

info 3 Dumps the contents of the system control status registers (CSRs) for the C-chip, D-chip, and P-chips.

info 4 Displays the per CPU impure area in abbreviated form. The console uses this scratch area to save processor context.

Example 4–10 shows an abbreviated info 1 display.

Example 4–10 info 1

```
P00>>> info 1
pte 000000003FFA8000 0000000100001101  va 0000000010000000  pa 0000000000002000
pte 000000003FFA8008 0000000200001101  va 0000000010002000  pa 0000000000004000
pte 000000003FFA8010 0000000300001101  va 0000000010004000  pa 0000000000006000
pte 000000003FFA8018 0000000400001101  va 0000000010006000  pa 0000000000008000
pte 000000003FFA8020 0000000500001101  va 0000000010008000  pa 000000000000A000
pte 000000003FFA8028 0000000600001101  va 000000001000A000  pa 000000000000C000
pte 000000003FFA8030 0000000700001101  va 000000001000C000  pa 000000000000E000
pte 000000003FFA8038 0000000800001101  va 000000001000E000  pa 0000000000010000
pte 000000003FFA8040 0000000900001101  va 0000000010010000  pa 0000000000012000
pte 000000003FFA8048 0000000A00001101  va 0000000010012000  pa 0000000000014000
pte 000000003FFA8050 0000000B00001101  va 0000000010014000  pa 0000000000016000
pte 000000003FFA8058 0000000C00001101  va 0000000010016000  pa 0000000000018000
pte 000000003FFA8060 0000000D00001101  va 0000000010018000  pa 000000000001A000
pte 000000003FFA8068 0000000E00001101  va 000000001001A000  pa 000000000001C000
pte 000000003FFA8070 0000000F00001101  va 000000001001C000  pa 000000000001E000
pte 000000003FFA8078 0000001000001101  va 000000001001E000  pa 0000000000020000
pte 000000003FFA8080 0000001100001101  va 0000000010020000  pa 0000000000022000
pte 000000003FFA8088 0000001200001101  va 0000000010022000  pa 0000000000024000
pte 000000003FFA8090 0000001300001101  va 0000000010024000  pa 0000000000026000
pte 000000003FFA8098 0000001400001101  va 0000000010026000  pa 0000000000028000
pte 000000003FFA80A0 0000001500001101  va 0000000010028000  pa 000000000002A000
pte 000000003FFA80A8 0000001600001101  va 000000001002A000  pa 000000000002C000
pte 000000003FFA80B0 0000001700001101  va 000000001002C000  pa 000000000002E000
pte 000000003FFA80B8 0000001800001101  va 000000001002E000  pa 0000000000030000
pte 000000003FFA80C0 0000001900001101  va 0000000010030000  pa 0000000000032000
pte 000000003FFA80C8 0000001A00001101  va 0000000010032000  pa 0000000000034000
pte 000000003FFA80D0 0000001B00001101  va 0000000010034000  pa 0000000000036000
pte 000000003FFA80D8 0000001C00001101  va 0000000010036000  pa 0000000000038000
pte 000000003FFA80E0 0000001D00001101  va 0000000010038000  pa 000000000003A000
pte 000000003FFA80E8 0000001E00001101  va 000000001003A000  pa 000000000003C000
.
.
.
```

Example 4–11 shows an abbreviated **info 2** display.

Example 4–11 info 2

```
P00>>> info 2
GCT_BUFFER_HEADER
addr of config tree       1de000
buffer_cksum              c0b531e5309ee27d
buffer_size               8000
rev_major                 5
rev_minor                 2
galaxy_enable             1
galaxy_callbacks          0

GCT_ROOT_NODE
Root->lock                ffffffff
Root->transient_level     1
Root->Current_level       1
Root->console_req         200000
Root->min_alloc           100000
Root->min_align           100000
Root->base_alloc          2000000
Root->base_align          2000000
Root->max_phys_addr       800000000
Root->mem_size            80000000
Root->platform_type       140500000022
Root->platform_name       200
Root->primary_instance    0
Root->first_free          0
Root->high_limit          7d40
Root->lookaside           0
Root->available           0
Root->max_partition       1
Root->partitions          100
Root->communities         140
Root->max_plat_partition  2
Root->max_frag            10
Root->max_desc            4
Root->galaxy_id           1de108
Root->bindings            180

GCT Depth View:
   Type 2  ID ffffffffffffff00 HdExt 40 FRU 24c0 cnt 1
      Type 16  ID ff0000ffffffffff HdExt a8 FRU 2580 cnt 1
         Type 9  ID ff0000ff00ff0000 HdExt 120 FRU 2680 cnt 1
         Type 9  ID ff0000ff00ff0001 subtyp 1 HdExt 120 FRU 2740 cnt 1
         Type 9  ID ff0000ff00ff0002 subtyp 1 HdExt 120 FRU 2800 cnt 1
         Type 9  ID ff0000ff00ff0003 subtyp 1 HdExt 120 FRU 28c0 cnt 1
        .
        .
        .

dump each node ? (Y/<N>)  N

dump binary ? (Y/<N>) N
P00>>>
P00>>>
```

Example 4–12 shows an abbreviated **info 3** display.

Example 4–12 info 3

```
P00>>> info 3

CCHIP   CSRs:               801a0000000
CSC                    002140809A19796F : 0000
MTR                    00000F6414000125 : 0040
AAR0                   0000000040006105 : 0100
AAR1                   0000000000007105 : 0140
AAR2                   0000000060005005 : 0180
AAR3                   0000000070005005 : 01c0
.
.
.

DCHIP   CSRs:               801b0000000
DSC                    7F7F7F7F7F7F7F7F : 0800
DSC2                   7F7F7F7F7F7F7F7F : 08c0
STR                    3939393939393939 : 0840
DREV                   0101010101010101 : 0880

PCHIP 0 CSRs:               80180000000
WSBA0                  0000000000800000 : 0000
WSBA1                  0000000080000001 : 0040
WSBA2                  0000000000000000 : 0080
WSBA3                  0000000000000000 : 00c0
WSM0                   0000000000700000 : 0100
.
.
.

PCHIP 1 CSRs:               80380000000
WSBA0                  0000000000800000 : 0000
WSBA1                  0000000080000001 : 0040
WSBA2                  0000000000000000 : 0080
WSBA3                  0000000000000000 : 00c0
WSM0                   0000000000700000 : 0100
.
.
.
```

Example 4–13 shows an abbreviated **info 4** display.

Example 4–13 info 4

```
P00>>> info 4

                      cpu00    cpu01    cpu02    cpu03
per_cpu impure area  00004200 00004800 00004e00 00005400
cns$flag             00000001 00000001 00000001 00000001 : 0000
cns$flag+4           00000000 00000000 00000000 00000000 : 0004
cns$hlt              00000000 00000000 00000000 00000000 : 0008
cns$hlt+4            00000000 00000000 00000000 00000000 : 000c
cns$mchkflag         000001c8 000001c8 000001c8 000001c8 : 0210
cns$mchkflag+4       00000000 00000000 00000000 00000000 : 0214
cns$fpcr             00000000 00000000 00000000 00000000 : 0318
cns$fpcr+4           8ff00000 8ff00000 8ff00000 8ff00000 : 031c
cns$va               fffffffc 0016270c 0016270c 16333d20 : 0320
cns$va+4             ffffffff 00000000 00000000 00000000 : 0324
.
.
.
```

4.12 kill and kill_diags

The kill and kill_diags commands terminate diagnostics that are currently executing.

Example 4–14 kill and kill_diags

```
P00>>> memexer 3
P00>>> show_status
 ID       Program      Device       Pass  Hard/Soft Bytes Written  Bytes Read
-------- ------------ ------------ ------ --------- ------------- -----------
00000001         idle system          0    0    0               0           0
0000125e      memtest memory         12    0    0      6719275008  6719275008
00001261      memtest memory         12    0    0      6689914880  6689914880
00001268      memtest memory         11    0    0      6689914880  6689914880
0000126f     exer_kid dka0.0.0.2.1    0    0    0               0     8612352
00001270     exer_kid dka100.1.0.2    0    0    0               0     8649728
00001271     exer_kid dka200.2.0.2    0    0    0               0     8649728
00001278     exer_kid dqa0.0.0.15.    0    0    0               0     3544064
00001280     exer_kid dfa0.0.0.2.1   84    0    0               0     8619520
00001281     exer_kid dfb0.0.0.102 1066    0    0               0   109256192
0000128e     exer_kid dva0.0.0.100    0    0    0               0      980992
00001381      nettest ewa0.0.0.4.1  362    0    1         1018720     1018496
P00>>> kill_diags

dva0.0.0.1000.0 exer completed

packet       IOs            elapsed  idle
size         IOs     bytes read  bytes written   /sec bytes/sec seconds  secs
512          112          28672          28672      5      2748      21    16
```

The **kill** command terminates a specified process. The **kill_diags** command terminates all diagnostics.

Syntax

kill_diags

kill [PID. . .]

Arguments

[PID. . .] The process ID of the diagnostic to terminate. Use the **show_status** command to determine the process ID.

4.13 memexer

The memexer command runs a specified number of memory exercisers in the background. Nothing is displayed unless an error occurs. Each exerciser tests all available memory in twice the backup cache size blocks for each pass.

The following example shows no errors.

Example 4–15 memexer

```
P00>>> memexer 3
P00>>> show_status
 ID       Program      Device       Pass  Hard/Soft Bytes Written  Bytes Read
-------- ------------ ------------ ------ --------- ------------- -----------
00000001         idle system          0    0    0             0           0
0000125e      memtest memory         12    0    0    6719275008  6719275008
00001261      memtest memory         12    0    0    6689914880  6689914880
00001268      memtest memory         11    0    0    6689914880  6689914880
0000126f     exer_kid dka0.0.0.2.1    0    0    0             0     8612352
00001270     exer_kid dka100.1.0.2    0    0    0             0     8649728
00001271     exer_kid dka200.2.0.2    0    0    0             0     8649728
00001278     exer_kid dqa0.0.0.15.    0    0    0             0     3544064
00001280     exer_kid dfa0.0.0.2.1   84    0    0             0     8619520
00001281     exer_kid dfb0.0.0.102 1066    0    0             0   109256192
0000128e     exer_kid dva0.0.0.100    0    0    0             0      980992
00001381      nettest ewa0.0.0.4.1  362    0    1       1018720     1018496
```

The following example shows a memory compare error indicating bad DIMMs. In most cases, the failing bank and DIMM position are specified in the error message.

```
P00>>> memexer 3

*** Hard Error - Error #41 - Memory compare error

Diagnostic Name    ID         Device Pass  Test  Hard/Soft   11-FEB-1999
memtest            00000193   brd0   114   1      0          12:00:01
Expected value:    25c07
Received value     35c07
Failing addr:      a11848

*** ERROR - DIMM 1 on MMB 1 Failed ***

P00>>> kill_diags
P00>>>
```

If the memory configuration is very large, the console might not test all of the memory. The upper limit is 1 GB.

Use the **show_status** command to display the progress of the tests. Use the **kill** or **kill_diags** command to terminate the test.

Syntax

memexer [number]

Arguments

[number] Number of memory exercisers to start. The default is 1.

The number of exercisers, as well as the length of time for testing, depends on the context of the testing.

4.14 memtest

The memtest command exercises a specified section of memory. Typically memtest is run from the built-in console script. Advanced users may want to use the specific options described here.

Example 4–16 memtest

```
P00>>> sh mem                                    ➊
  Array       Size       Base Address
---------  ----------  ----------------
    0        256Mb     0000000060000000
    1        512Mb     0000000040000000
    2        256Mb     0000000070000000
    3       1024Mb     0000000000000000

     2048 MB of System Memory
                     ➋         ➌      ➍
P00>>>memtest -sa 400000 -l 2000000 -p 10&
*** Hard Error - Error #43 - Memory compare error

Diagnostic Name  ID            Device  Pass  Test  Hard/Soft   1-JAN-2066
memtest          00000118      brd0     1     1     1    0     12:00:01
Expected value:               fffffffe
Received value:               ffffffff
Failing addr:                   400004
*** Error - DIMM 3 on MMB 2 Failed ***➎
```

❶ Use the **show memory** command or an **info 0** command to see where memory is located.

❷ Starting address

❸ Length of the section to test in bytes

❹ Passcount. In this example, the test will run for 10 passes.

❺ The test detected a failure on DIMM 3, which is located on MMB 2.

Use the **show_status** command to display the progress of the test. Use the **kill** or **kill_diags** command to terminate the test.

Memtest provides a graycode memory test. The test writes to memory and then reads the previously written value for comparison. The section of memory that is tested has its data destroyed. The **-z** option allows testing outside of the main memory pool. Use caution because this option can overwrite the console.

Memtest may be run on any specified address. If the **-z** option is not included (default), the address is verified and allocated from the firmware's memory zone. If the **-z** qualifier is included, the test is started without verification of the starting address.

When a starting address is specified, the memory is allocated beginning at the starting address -32 bytes for the length specified. The extra 32 bytes that are allocated are reserved for the allocation header information. Therefore, if a starting address of 0xa00000 and a length of 0x100000 is requested, the area from 0x9fffe0 through 0xb00000 is reserved. This may be confusing if you try to begin two **memtest** processes simultaneously with one beginning at 0xa00000 for a length of 0x100000 and the other at 0xb00000 for a length of 0x100000. The second **memtest** process will send a message that it is "Unable to allocate memory of length 100000 at starting address b00000." Instead, the second process should use the starting address of 0xb00020.

Continued on next page

NOTE: *If **memtest** is used to test large sections of memory, testing may take a while to complete. If you issue a Ctrl/C or **kill PID** in the middle of testing, **memtest** may not abort right away. For speed reasons, a check for a Ctrl/C or **kill** is done outside of any test loops. If this is not satisfactory, you can run concurrent **memtest** processes in the background with shorter lengths within the target range.*

Memtest Test 1 — Graycode Test

Memtest Test 1 uses a graycode algorithm to test a specified section of memory. The graycode algorithm used is: data = (x>>1)^x, where x is an incrementing value.

Three passes are made of the memory under test.

- The first pass writes alternating graycode inverse graycode to each four longwords. This causes many data bits to toggle between each 16-byte write.

  For example graycode patterns for a 32 byte block would be:

  Graycode(0) 00000000 Graycode(1) 00000001 Graycode(2) 00000003 Graycode(3) 00000002 Inverse Graycode(4) FFFFFFF9 Inverse Graycode(5) FFFFFFF8 Inverse Graycode(6) FFFFFFFA Inverse Graycode(7) FFFFFFFB

- The second pass reads each location, verifies the data, and writes the inverse of the data, one longword at a time. This causes all data bits to be written as a one and zero.
- The third pass reads and verifies each location.

You can specify the **-f** (fast) option so that the explicit data verify sections of the second and third loops are not performed. This does not catch address shorts but stresses memory with a higher throughput. The ECC/EDC logic can be used to detect failures.

Syntax

memtest ([-sa <start_address>] [-ea <end_address>] [-l <length>] [-bs <block_size>] [-i <address_inc>] [-p <pass_count>] [-d <data_pattern>] [-rs <random_seed>] [-ba <block_address>] [-t <test_mask>] [-se <soft_error_threshold>] [-g <group_name>] [-rb] [-f] [-m] [-z] [-h] [-mb])

Options

-sa	Start address. Default is first free space in memzone.
-ea	End address. Default is start address plus length size.
-l	Length of section to test in bytes, default is the zone size with the **-rb** option and the block_size for all other tests. **-l** has precedence over **-ea**.
-bs	Block (packet) size in bytes in hex, default 8192 bytes. This is used only for the random block test. For all other tests the block size equals the length.
-i	Specifies the address increment value in longwords. This value is used to increment the address through the memory to be tested. The default is 1 (longword). This is only implemented for the graycode test. An address increment of 2 tests every other longword. This option is useful for multiple CPUs testing the same physical memory.
-p	Passcount If 0 then run indefinitely or until Ctrl/C is issued. Default = 1
-t	Test mask. Default = run all tests in selected group.
-g	Group name
-se	Soft error threshold
-f	Fast. If **-f** is included in the command line, the data compare is omitted. Detects only ECC/EDC errors.

Continued on next page

Options

-m Timer. Prints out the run time of the pass. Default = off .

-z Tests the specified memory address without allocation. Bypasses all checking but allows testing in addresses outside of the main memory heap. Also allows unaligned input.

CAUTION: *This flag can overwrite the console. If the system hangs, press the Reset button.*

-d Used only for march test (2). Uses this pattern as test pattern. Default = 5's

-h Allocates test memory from the firmware heap.

-rs Used only for random test (3). Uses this data as the random seed to vary random data patterns generated. Default = 0.

-rb Randomly allocates and tests all of the specified memory address range. Allocations are done of block_size.

-mb Memory barrier flag. Used only in the **-f** graycode test. When set an mb is done after every memory access. This guarantees serial access to memory.

-ba Used only for block test (4). Uses the data stored at this address to write to each block.

4.15 net

The net command performs maintenance operations on a specified Ethernet port. Net -ic initializes the MOP counters for the specified Ethernet port, and net -s displays the current status of the port, including the contents of the MOP counters.

Example 4–17 net -ic and net -s

```
P00>>> net -ic ewa0
P00>>> net -s ewa0
Status counts:
ti: 72 tps: 0 tu: 47 tjt: 0 unf: 0 ri: 70 ru: 0
rps: 0 rwt: 0 at: 0 fd: 0 lnf: 0 se: 0 tbf: 0
tto: 1 lkf: 1 ato: 1 nc: 71 oc: 0

MOP BLOCK:
 Network list size: 0

MOP COUNTERS:
Time since zeroed (Secs): 3

TX:
 Bytes: 0 Frames: 0
 Deferred: 0 One collision: 0 Multi collisions: 0
TX Failures:
 Excessive collisions: 0 Carrier check: 0 Short circuit: 0
 Open circuit: 0 Long frame: 0 Remote defer: 0
 Collision detect: 0

RX:
 Bytes: 0 Frames: 0
 Multicast bytes: 0 Multicast frames: 0
RX Failures:
 Block check: 0 Framing error: 0 Long frame: 0
 Unknown destination: 0 Data overrun: 0 No system buffer: 0
 No user buffers: 0
P00>>>
```

Continued on next page

Syntax

net [-ic]

net [-s]

Arguments

<port_name>	Specifies the Ethernet port on which to operate, either ei*0 or ew*0.

4.16 nettest

The nettest command tests the network ports using MOP loopback. Typically nettest is run from the built-in console script. Advanced users may want to use the specific options and environment variables described here.

Example 4–18 nettest

```
P00>>> nettest ei*                          ❶
P00>>> nettest -mode in ew*                 ❷
P00>>> nettest -mode ex -w 10  e*           ❸
```

❶ Internal loopback test on port ei*0

❷ Internal loopback test on ports ewa0/ewb0

❸ External loopback test on port eia0 or ewa0; wait 10 seconds between tests

Nettest performs a network test. It can test the ei* or ew* ports in internal loopback, external loopback, or live network loopback mode.

Nettest contains the basic options to run MOP loopback tests. Many environment variables can be set from the console to customize **nettest** before **nettest** is started. The environment variables, a brief description, and their default values are listed in the syntax table in this section. Each variable name is preceded by e*a0_ or e*b0_ to specify the desired port.

You can change other network driver characteristics by modifying the port mode. See the **-mode** option.

Use the **show_status** display to determine the process ID when terminating an individual diagnostic test. Use the **kill** or **kill_diags** command to terminate tests.

Syntax

nettest ([-f <file>] [-mode <port_mode>] [-p <pass_count>] [-sv <mop_version>] [-to <loop_time>] [-w <wait_time>] [<port>])

Arguments

<port> Specifies the Ethernet port on which to run the test.

Options

-f <file> Specifies the file containing the list of network station addresses to loop messages to. The default file name is lp_nodes_e*a0 for port e*a0. The default file name is lp_nodes_e*b0 for port e*b0. The files by default have their own station address.

-mode <port_mode> Specifies the mode to set the port adapter (TGEC). The default is ex (external loopback). Allowed values are:

df : default, use environment variable values

ex : external loopback

in : internal loopback

nm : normal mode

nf : normal filter

pr : promiscuous

mc : multicast

ip : internal loopback and promiscuous

fc : force collisions

nofc : do not force collisions

nc : do not change mode

-p <pass_count> Specifies the number of times to run the test. If 0, then run until terminated by a **kill** or **kill_diags** command The default is 1.

NOTE: *This is the number of passes for the diagnostic. Each pass will send the number of loop messages as set by the environment variable,* **eia*_loop_count** *or* **ewa*_loop_count**.

-sv <mop_version>	Specifies which MOP version protocol to use. If 3, then MOP V3 (DECNET Phase IV) packet format is used. If 4, then MOP V4 (DECNET Phase V IEEE 802.3) format is used.
-to <loop_time>	Specifies the time in seconds allowed for the loop messages to be returned. The default is 2 seconds.
-w <wait_time>	Specifies the time in seconds to wait between passes of the test. The default is 0 (no delay). The network device can be very CPU intensive. This option will allow other processes to run.

Environment Variables

e*a*_loop_count	Specifies the number (hex) of loop requests to send. The default is 0x3E8 loop packets.
e*a*_loop_inc	Specifies the number (hex) of bytes the message size is increased on successive messages. The default is 0xA bytes.
e*a*_loop_patt	Specifies the data pattern (hex) for the loop messages. The following are legitimate values. 0 : all zeros 1 : all ones 2 : all fives 3 : all 0xAs 4 : incrementing data 5 : decrementing data ffffffff : all patterns
loop_size	Specifies the size (hex) of the loop message. The default packet size is 0x2E.

4.17 set sys_serial_num

The set sys_serial_num command sets the system serial number. This command is used by Manufacturing for establishing the system serial number, which is then propagated to all FRU devices that have EEPROMs. The sys_serial_num environment variable can be read by the operating system.

Example 4–19 set sys_serial_num

```
P00>>> set sys_serial_num NI900100022
```

When the system motherboard (SMB) is replaced, you must use the **set sys_serial_num** command to restore the master setting.

Syntax

set sys_serial_num *value*

Value is the system serial number, which is printed on the system chassis.

4.18 show error

The show error command reports errors logged to the FRU EEPROMs.

Example 4–20 show error

```
P00>>> show error
 ❶
SMB0          TDD - Type: 15 Test: 15  SubTest: 15  Error: 15          ❷
001f8408  0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F   ................
SMB0          SDD - Type: 14 LastLog: 0   Overwrite: 0                  ❸
001f8408  0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F   ................
001f8418  0F 0F 0F 0F 0F 0F 0F 0F 0F 00 00 00 00 00 00 00   ................
001f8428  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001f8438  00 00 00 00 00 00 00 00 00 00 00 FF 00 00 00 00   ................
001f8448  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001f8458  00 00 00 00 00 00 00 00                           ........
SMB0          Bad checksum   0 to 64  EXP:dc  RCV:dd                    ❹
001f8408  80 08 00 01 53 00 01 00 00 00 00 00 00 00 00 00   ....S...........
001f8418  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001f8428  FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001f8438  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 DD   ...............Y
SMB0          Bad checksum  64 to 126  EXP:e1  RCV:0f
001f8408  4A FF FF FF FF FF FF FF 02 35 34 2D 31 32 33 34   J........54-1234
001f8418  35 2D 30 31 2E 41 30 30 31 20 20 00 00 09 44 91   5-01.A001  ...D.
001f8428  34 51 15 41 41 41 41 41 41 41 41 41 41 41 41 41   4Q.AAAAAAAAAAAAA
001f8438  0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F   ................
SMB0          Bad checksum 128 to 254  EXP:0c  RCV:0d
001f8408  0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F   ................
001f8418  0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F   ................
001f8428  0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F   ................
001f8438  0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 00 00   ................
001f8408  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001f8418  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001f8428  FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001f8438  00 00 00 00 00 00 00 00 00 00 00 00 00 4A 21 0D   .............J!.
SMB0          SYS_SERIAL_NUM Mismatch                                    ❺
P00>>>
```

The output of the **show error** command is based on information logged to the serial control bus EEPROMs on the system FRUs. Both the operating system and the ROM-based diagnostics log errors to the EEPROMs. This functionality allows you to generate an error log from the console environment. No errors are displayed for fans or the OCP because these components do not have an EEPROM.

Syntax

show error

All FRUs with errors are displayed. If no errors are logged, nothing is displayed and you are returned to the SRM console prompt.

Example 4–20 shows TDD, SDD, checksum, and sys_serial_num mismatch errors logged to the EEPROM on the system motherboard (SMB0). Table 4–2 shows a reference to these errors. The bit masks correspond to the bit masks that would be displayed in the E field of the **show fru** command.

❶ FRU to which errors are logged; in this example the system motherboard, SMB0.

❷ A TDD error has been logged. TDDs (test-directed diagnostics) test specific functions sequentially. Typically, nothing else is running during the test. TDDs are performed in SROM or XSROM or early in the console power-up flow.

❸ An SDD error has been logged. SDDs (symptom-directed diagnostics) are generic diagnostic exercisers that try to cause random behavior and look for failures or “symptoms.” All SDDs are logged by Compaq Analyze.

❹ Three checksum errors have been logged.

❺ There was a mismatch between the serial number on the system motherboard and the system serial number. This could occur if a motherboard from a system with a different serial number was swapped into this system.

Table 4–2 Show Error Message Translation

Bit Mask (E Field)	Text Message	Meaning and Action
01	<*fruname*> Hardware Failure	Module failure. FRUs that are known to be connected but are unreadable are considered hardware failures. An example is power supplies.
02	<*fruname*> TDD - Type:0 Test: 0 SubTest: Error: 0	Serious error. Run the Compaq Analyze GUI, if necessary, to determine what action to take. If you cannot run Compaq Analyze, replace the module.
04	<*fruname*> SDD - Type:0 LastLog: 0 Overwrite: 0	Serious error. Compaq Analyze (CA) has written a FRU callout into the SDD area and DPR global area. Follow the instructions given by Compaq Analyze.
08	<*fruname*> EEPROM Unreadable	Reserved.
10	<*fruname*> Bad checksum 0 to 64 EXP:01 RCV:02	Informational. Use the **clear_error** command to clear the error unless TDD or SDD is also set.
20	<*fruname*> Bad checksum 64 to 126 EXP:01 RCV:02	Informational. Use the **clear_error** command to clear the error unless TDD or SDD is also set.
40	<*fruname*> Bad checksum 128 to 254 EXP:01 RCV:02	Informational. Use the **clear_error** command to clear the error unless TDD or SDD is also set.
40	<*fruname*> SYS_SERIAL_NUM Mismatch	Informational. Use the **clear_error** command to clear the error unless TDD or SDD is also set.

4.19 show fru

The show fru command displays the physical configuration of FRUs. Use show fru -e to display FRUs with errors.

Example 4–21 show fru

```
P00>>> build smb0 54-25385-01.a01 ay94412345
P00>>> show fru
   ➊           ➋      ➌                ➍           ➎          ➏

FRUname         E  Part#              Serial#    Model/Other Alias/Misc
SMB0            00 54-25385-01.A01    AY94412345
SMB0.CPU0       00 54-24801-03        AY80112345 DEC         DEC
SMB0.CPU1       00 54-24801-03        AY80112345 DEC         DEC
SMB0.CPU2       00 54-24801-03        AY80112345 DEC         DEC
SMB0.CPU3       00 54-24801-03        AY80112345 DEC         DEC
SMB0.MMB0       00 54-25582-01.B02    AY90112345 CARRIER     MMB
SMB0.MMB0.DIM1  00 54-25053-BACPQ     NI90224341 COMPAQ
SMB0.MMB0.DIM2  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB0.DIM3  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB0.DIM4  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB0.DIM5  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB0.DIM6  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB1       00 54-25582-01.B02    CARRIER    CARRIER     MMB
SMB0.MMB1.DIM1  00 54-25053-BACPQ     NI90224341 COMPAQ
SMB0.MMB1.DIM2  00 54-25053-BACPQ     NI90224341 COMPAQ
SMB0.MMB1.DIM3  00 54-25053-BACPQ     NI90224341 COMPAQ
SMB0.MMB1.DIM4  00 54-25053-BACPQ     NI90224341 COMPAQ
SMB0.MMB1.DIM5  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB1.DIM6  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB2       00 54-25582-01.B02    AY90112345 CARRIER     MMB
SMB0.MMB2.DIM1  00 54-25053-BACPQ     NI90224341 COMPAQ
SMB0.MMB2.DIM2  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB2.DIM3  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB2.DIM4  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB2.DIM5  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB2.DIM6  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB3       00 54-25582-01.B02    AY90112345 CARRIER     MMB
SMB0.MMB3.DIM1  00 54-25053-BACPQ     NI90224341 COMPAQ
SMB0.MMB3.DIM2  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB3.DIM3  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB3.DIM4  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB3.DIM5  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.MMB3.DIM6  00 54-25053-BACPQ     NI90112345 COMPAQ
SMB0.CPB0       00 54-12345-01        AY80110000
SMB0.CPB0.PCI4  00 DEC PowerStorm
SMB0.CPB0.PCI5  00 NCR 53C895
SMB0.CPB0.PCI7  00 DEC PCI MC
SMB0.CPB0.PCI8  00 DEC PCI MC
SMB0.CPB0.PCIA  00 DE500-BA Network C
JIO0            00 54-25575-01        -          Junk I/O
OCP0            00 70-33894-0x        -          OCP
PWR0            00 30-49448-01. C02   2P91600482 API-7650
```

```
PWR1            00 30-49448-01. C02   2P91600530 API-7650
FAN1            00 70-40073-01        -          Fan
FAN2            00 70-40073-01        -          Fan
FAN3            00 70-40072-01        -          Fan
FAN4            00 70-40071-01        -          Fan
FAN5            00 70-40073-02        -          Fan
FAN6            00 70-40074-01        -          Fan
SMB0.CPB0.SBM0  06 54-12345-01        AY80151237
```

❶	FRUname	The FRU name recognized by the SRM console. The name also indicates the location of that FRU in the physical hierarchy. SMB = system motherboard; CPU = CPUs; MMB = memory motherboard; DIM = DIMMs; CPB = PCI backplane; PCI = PCI option; SBM = SCSI backplane; PWR = power supply; FAN = fans; JIO= I/O connector module (junk I/O).
❷	E	Error field. Indicates whether the FRU has any errors logged against it. FRUs without errors show 00 (hex). FRUs with errors have a non-zero value that represents a bit mask of possible errors. See Table 4–3.
❸	Part #	The part number of the FRU in ASCII, either a Compaq part number or a vendor part number.
❹	Serial #	The serial number. For Compaq FRUs, the serial number has the form XXYWWNNNNN. XX = manufacturing location code YWW = year and week NNNNN = sequence number. For vendor FRUs, the 4-byte sequence number is displayed in hex.
❺	Model/Other	Optional data. For Compaq FRUs, the Compaq part alias number (if one exists). For vendor FRUs, the year and week of manufacture.
❻	Alias/Misc	Miscellaneous information about the FRUs. For Compaq FRUs, a model name, number, or the common name for the entry in the Part # field. For vendor FRUs, the manufacturer's name.

Table 4–3 lists bit assignments for failures that could potentially be listed in the E (error) field of the **show fru** command. Because the E field is only two characters wide, bits are “or’ed” together if the device has multiple errors. For example, the E field for a FRU with both TDD (02) and SDD (04) errors would be 06:

010 | 100 = 110 (6)

Table 4–3 Bit Assignments for Error Field

Bit Mask (E Field)	Meaning
01	Hardware failure
02	TDD error has been logged
04	SDD error has been logged
08	Reserved
10	Checksum failure on bytes 0-62
20	Checksum failure on bytes 64-126
40	Checksum failure on bytes 128-254
80	FRU’s system serial number does not match system’s

4.20 show_status

The show_status command displays the progress of diagnostics. The command reports one line of information per executing diagnostic. Many of the diagnostics run in the background and provide information only if an error occurs.

Example 4–22 show status

```
P00>>> show_status

➊           ➋           ➌            ➍     ➎          ➏              ➐
 ID       Program      Device       Pass  Hard/Soft Bytes Written  Bytes Read
-------- ------------ ------------ ------ --------- ------------- -----------
00000001         idle system          0     0    0              0           0
0000125e      memtest memory         12     0    0     6719275008  6719275008
00001261      memtest memory         12     0    0     6689914880  6689914880
00001268      memtest memory         11     0    0     6689914880  6689914880
0000126f     exer_kid dka0.0.0.2.1    0     0    0              0     8612352
00001270     exer_kid dka100.1.0.2    0     0    0              0     8649728
00001271     exer_kid dka200.2.0.2    0     0    0              0     8649728
00001278     exer_kid dqa0.0.0.15.    0     0    0              0     3544064
00001280     exer_kid dfa0.0.0.2.1   84     0    0              0     8619520
00001281     exer_kid dfb0.0.0.102 1066     0    0              0   109256192
0000128e     exer_kid dva0.0.0.100    0     0    0              0      980992
00001381      nettest ewa0.0.0.4.1  362     0    1        1018720     1018496
P00>>>
```

❶ Process ID

❷ The SRM diagnostic for the particular device

❸ The ID of the device under test

❹ Number of diagnostic passes that have been completed

❺ Error count (hard and soft). Soft errors are not usually fatal; hard errors halt the system or prevent completion of the diagnostics.

❻ Bytes successfully written by the diagnostic.

❼ Bytes successfully read by the diagnostic.

The following command string is useful for periodically displaying diagnostic status information for diagnostics running in the background:

```
P00>>> while true;show_status;sleep n;done
```

Where *n* is the number of seconds between **show_status** displays.

Syntax

show_status

4.21 sys_exer

The sys_exer command exercises the devices displayed with the show config command. Tests are run concurrently and in the background. Nothing is displayed after the initial test startup messages unless an error occurs.

Example 4–23 sys_exer

```
P00>>> sys_exer
Default zone extended at the expense of memzone.
Use INIT before booting
Exercising the Memory
Exercising the DK* Disks(read only)
Exercising the DQ* Disks(read only)
Exercising the DF* Disks(read only)
Exercising the Floppy(read only)
Testing the VGA (Alphanumeric Mode only)
Exercising the EWA0 Network

Type "show_status" to display testing progress
Type "cat el" to redisplay recent errors
Type "init" in order to boot the operating system
P00>>> show_status
 ID       Program      Device       Pass  Hard/Soft Bytes Written  Bytes Read
-------- ------------ ------------ ------ --------- ------------- -----------
00000001         idle system          0    0    0             0           0
0000125e      memtest memory         12    0    0    6719275008  6719275008
00001261      memtest memory         12    0    0    6689914880  6689914880
00001268      memtest memory         11    0    0    6689914880  6689914880
0000126f     exer_kid dka0.0.0.2.1    0    0    0             0     8612352
00001270     exer_kid dka100.1.0.2    0    0    0             0     8649728
00001271     exer_kid dka200.2.0.2    0    0    0             0     8649728
00001278     exer_kid dqa0.0.0.15.    0    0    0             0     3544064
00001280     exer_kid dfa0.0.0.2.1   84    0    0             0     8619520
00001281     exer_kid dfb0.0.0.102 1066    0    0             0   109256192
0000128e     exer_kid dva0.0.0.100    0    0    0             0      980992
00001381      nettest ewa0.0.0.4.1  362    0    1       1018720     1018496

P00>>> init

OpenVMS PALcode V1.44-1, Tru64 UNIX PALcode V1.41-1
...
starting console on CPU 0
```

Use the **show_status** command to display the progress of diagnostic tests. The diagnostics started by the **sys_exer** command automatically reallocate memory resources, because these tests require additional resources. Use the **init** command to reconfigure memory before booting an operating system.

Because the **sys_exer** tests are run concurrently and indefinitely (until you stop them with the **init** command), they are useful in flushing out intermittent hardware problems.

When using the **sys_exer** command after shutting down an operating system, you must initialize the system to a quiescent state. Enter the following command at the SRM console:

```
P00>>> init
.
.
.
P00>>> sys_exer
```

By default, no write tests are performed on disk and tape drives. Media must be installed to test the floppy drive and tape drives. When the **-lb** argument is used, a loopback connector is required for the COM2 port (9-pin loopback connector, 12-27351-01) and parallel port (25-pin loopback connector).

Syntax

sys_exer [-lb] [-t]

Arguments

[-lb] The loopback option runs console loopback tests for the COM2 serial port and the parallel port during the test sequence.

[-t] Number of seconds to run. The default is run until terminated by a **kill** or **kill_diags** command.

4.22 test

The test command verifies all the devices in the system. This command can be used on all supported operating systems: Tru64 UNIX, OpenVMS, and Windows NT.

Example 4–24 test -lb

```
P00>>> test -lb
Testing the Memory
Testing the DK* Disks(read only)
No DU* Disks available for testing
No DR* Disks available for testing
Testing the DQ* Disks(read only)
Testing the DF* Disks(read only)
No MK* Tapes available for testing
No MU* Tapes available for testing
Testing the DV* Floppy Disks(read only)
Testing the Serial Port 1(external loopback)
Testing the parallel Port(external loopback)
Testing the VGA (Alphanumeric Mode only)
Testing the EW* Network
P00>>>
```

The **test** command also does a quick test on the system speaker. A beep is emitted as the command starts to run.

The tests are run sequentially, and the status of each subsystem test is displayed to the console terminal as the tests progress. If a particular device is not available to test, a message is displayed. The test script does no destructive testing; that is, it does not write to disk drives.

Syntax

test [*argument*]

Use the **-lb** (loopback) argument for console loopback tests.

To run a complete diagnostic test using the **test** command, the system configuration must include:

- A serial loopback connected to the COM2 port (not included)
- A parallel loopback connected to the parallel port (not included)

- A trial diskette with files installed
- A trial CD-ROM with files installed

The test script tests devices in the following order:

1. Memory tests (one pass)
2. Read-only tests: DK* disks, DR* disks, DQ* disks, MK* tapes, DV* floppy.

NOTE: *You must install media to test disks, tapes, and the floppy drive. Since no write tests are performed, it is safe to test disks and tapes that contain data.*

---

3. Console loopback tests if **-lb** argument is specified: COM2 serial port and parallel port.
4. VGA console tests: These tests are run only if the console environment variable is set to **serial**. The VGA console test displays rows of the word *compaq*.
5. Network internal loopback tests for EW* networks.

Testing a Windows NT System

To test a system running Windows NT, invoke the SRM console in one of the following ways and then enter the **test** command.

- Shut down the system from the Start button and wait for the message indicating that you can power off the system. Next, press the Reset button, and then press the Halt button.
- Alternatively, select **UNIX (SRM)** or **OpenVMS (SRM)** from the Advanced CMOS Setup screen and then reset the system.

The second method changes the **os_type** environment variable to **unix** or **openvms**, causing the SRM console to start on each subsequent reset. To restore your original setup for Windows NT, enter the following commands while still in the SRM console:

```
P00>>> set os_type nt
P00>>> init
```

Chapter 5
Error Logs

This chapter tells how to interpret error logs reported by the operating system. The following topics are covered:

- Error Log Analysis with Compaq Analyze
- Fault Detection and Reporting
- Machine Checks/Interrupts
- Environmental Errors Captured by SRM
- Windows NT Error Logs

5.1 Error Log Analysis with Compaq Analyze

Compaq Analyze (CA) is a fault management diagnostic tool that is used to determine the cause of hardware failures. Compaq Analyze performs system diagnostic processing of both single and multiple error/fault events.

Compaq Analyze may or may not be installed on the customer's system with the operating system, depending on the release cycle. If CA is installed, the Compaq Analyze Director starts automatically as part of the system start-up. CA provides automatic background analysis. When an error event occurs, it triggers the firing of an analysis rule. The analysis engine collects and processes the information and typically generates a "problem found" report, if appropriate. The report can be sent to users on a notification mailing list and, if DSNlink is installed, a call can be logged with the customer support center.

Compaq Analyze has the capability to support the Tru64 UNIX, OpenVMS, and Windows NT operating systems on AlphaServer platforms.

NOTE: *Compaq Analyze is a successor tool to DECevent and typically does not support the same systems as DECevent.*

5.1.1 WEB Enterprise Service (WEBES) Director

Compaq Analyze uses the functionality contained in the WEBES Director, a process that executes continuously on the machine. The Director manages the processing of system error events and provides analysis message routing for the system. Compaq Analyze provides the functionality for system event analysis and translation.

NOTE: *WEBES was formerly known as DESTA.*

The initial release of Compaq Analyze, V1.0, included the common WEBES code. Subsequent releases of Compaq Analyze will continue to ship with the common WEBES code.

The Director is started when the system is booted. Normally you do not need to start the Director. If the Director has stopped running, restart it by following the instructions in the WEBES documentation for the specific operating system.

Compaq Analyze includes a graphical user interface (GUI) that allows the user to interact with the Director. While only one Director process executes on the machine at any time, many GUI processes can run at the same time, connected to the single Director. Refer to the Compaq Analyze installation and user manuals for the respective operating system to launch the Compaq Analyze GUI. The Compaq Services service tools Web site available to customers is:

http://www.service.digital.com/svctools

The Compaq Analyze documentation includes the following:

- *Compaq Analyze User' s Guide*
- *Compaq Analyze Installation Guide for Tru64 UNIX*
- *Compaq Analyze Installation Guide for OpenVMS*
- *Compaq Analyze Installation Guide for Windows NT*
- *Compaq Analyze Releases Notes*

5.1.2 Invoking the GUI

When you invoke the Compaq Analyze GUI, the node "localhost" opens by default for all operating systems. The "localhost" is the system on which CA is running. If an event has occurred, it is listed under "localhost" Events. See Figure 5–1.

Figure 5–1 Compaq Analyze GUI

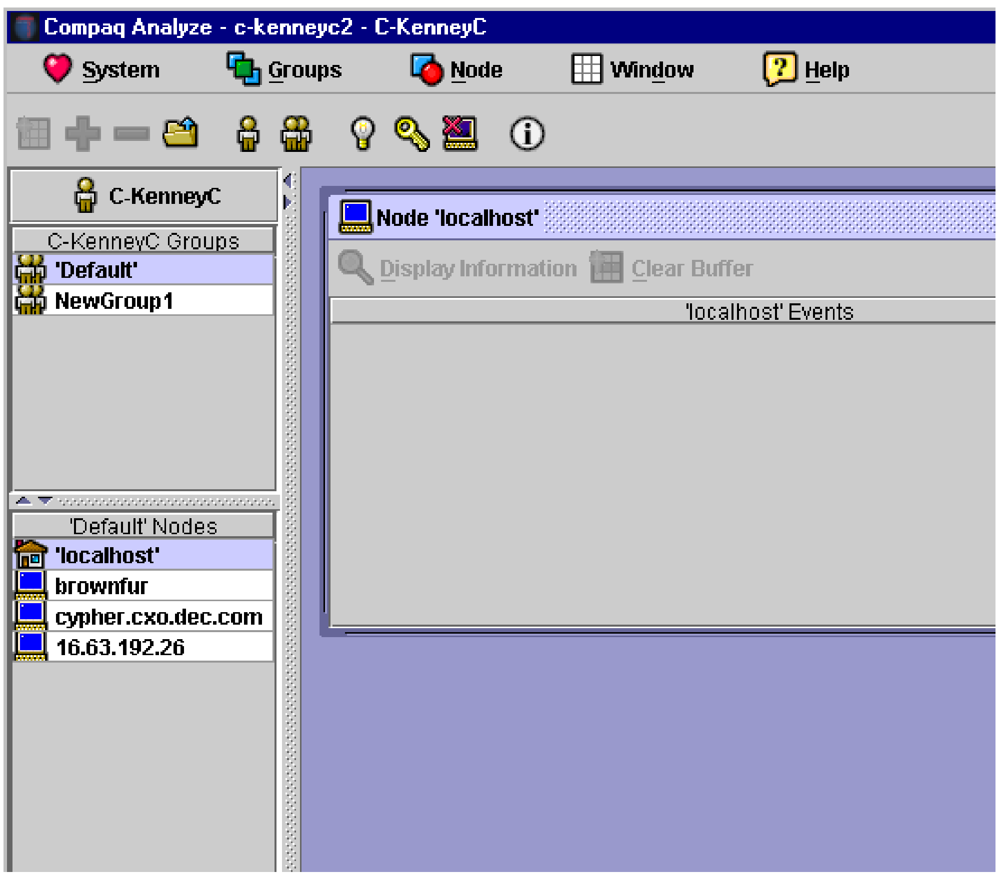
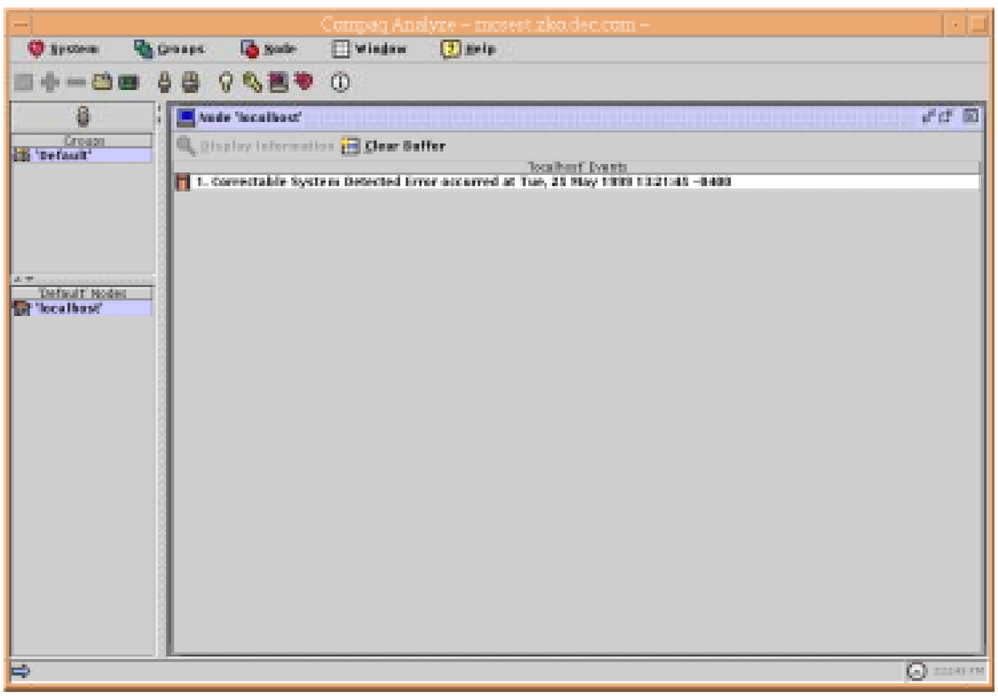


Figure 5–2 shows an example of an event screen for an ES40 system.

When an error is detected, it is reported to the console with a series of problem found statements. In this case, "Correctable System Detected Error" was logged in the event log with the date and time the event occurred.

To display an event or report, click on it to select it, then click on "Display Information." The item selected opens up in the data display window. See Figure 5–3.

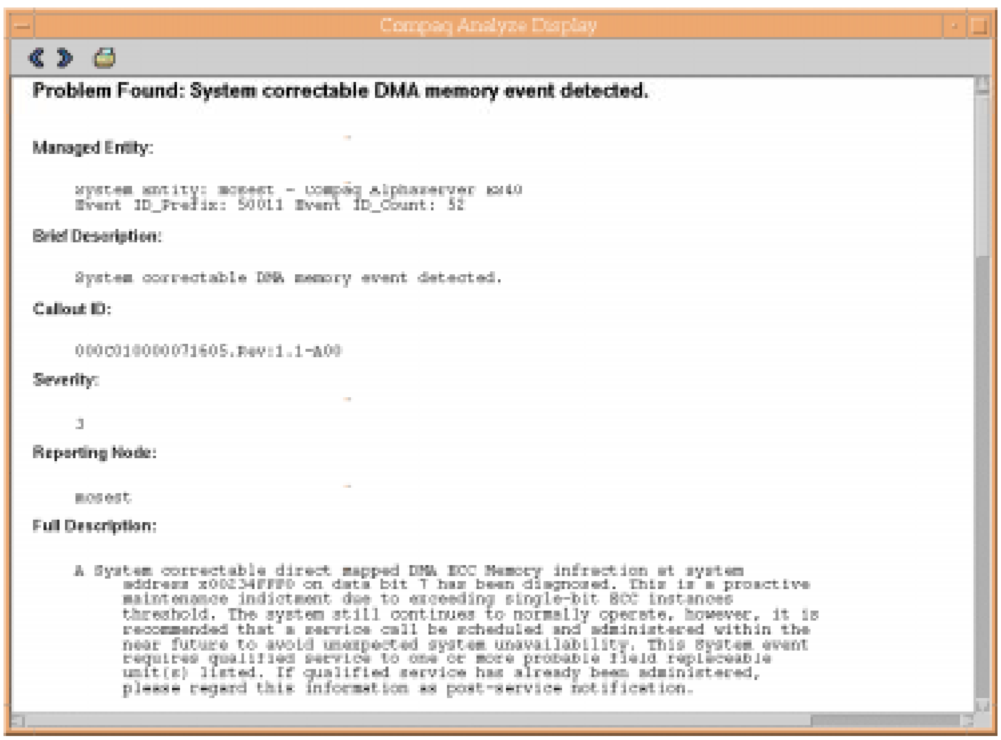
Figure 5–2 Compaq Analyze Event Screen



5.1.3 Problem Found Report

After you select the Problem Found report and click on Display Information, a full description of the error is displayed and probable FRUs and their location are called out. Figure 5–3 shows the beginning of a Compaq Analyze problem found report.

Figure 5–3 Problem Found Report



Managed Entity

The Managed Entity designator includes the system host name (typically a computer name for networking purposes), the type of computer system ("Compaq AlphaServer ES40"), and the error event identification. The error event identification uses new common event header Event_ID_Prefix and Event_ID_Count components. The Event_ID_Prefix refer to a OS specific identification for this event type. The Event_ID_Count indicates the number this event is of this event type.

Brief Description

The Brief Description designator indicates whether the error event is related to the CPU, system (PCI, storage, and so on), or environmental subsystem.

Callout ID

The last 12 characters of the Callout ID designator can be used to determine the revision level of the analysis rule-set that is being used.

Severity

The Severity designator indicates the severity of the problem.

Severity Level	Service Relevance	Comments
1	Critical	Not currently used.
2	Major	Fatal event that typically requires service.
3	Minor	Non-Fatal or Redundant warning event that typically requires future service, but system still operates normally.
4	Information	System service event such as enclosure PCI or fan door is open and requires closing.
5	Unknown	Not currently used.

Reporting Node

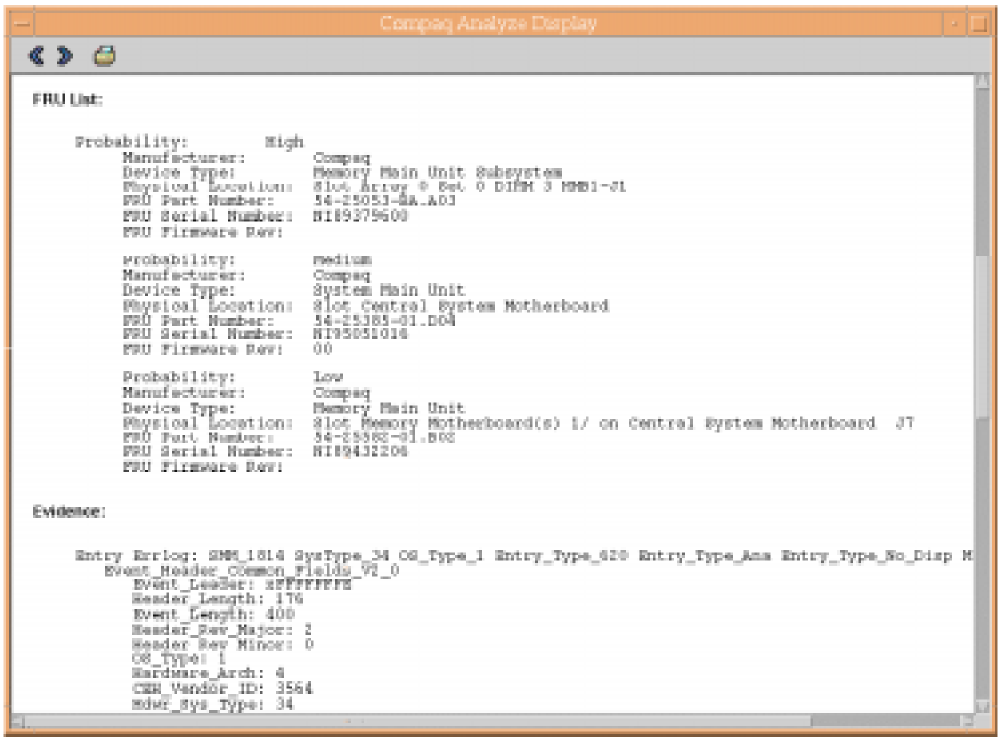
The Reporting Node designator is synonymous with the Managed Entity host name when Compaq Analysis is used to diagnose problems on the system on which it is running. For future implementations, the reporting node may be a system server reporting about a client within an enterprise computing environment.

Full Description

The Full Description designator provides detailed error information, which can include a description of the detected fault or error condition, the specific address or data bit where this fault or error occurred, the probable FRU list, and service related information.

Continued on next page

Figure 5–4 FRU List Designator



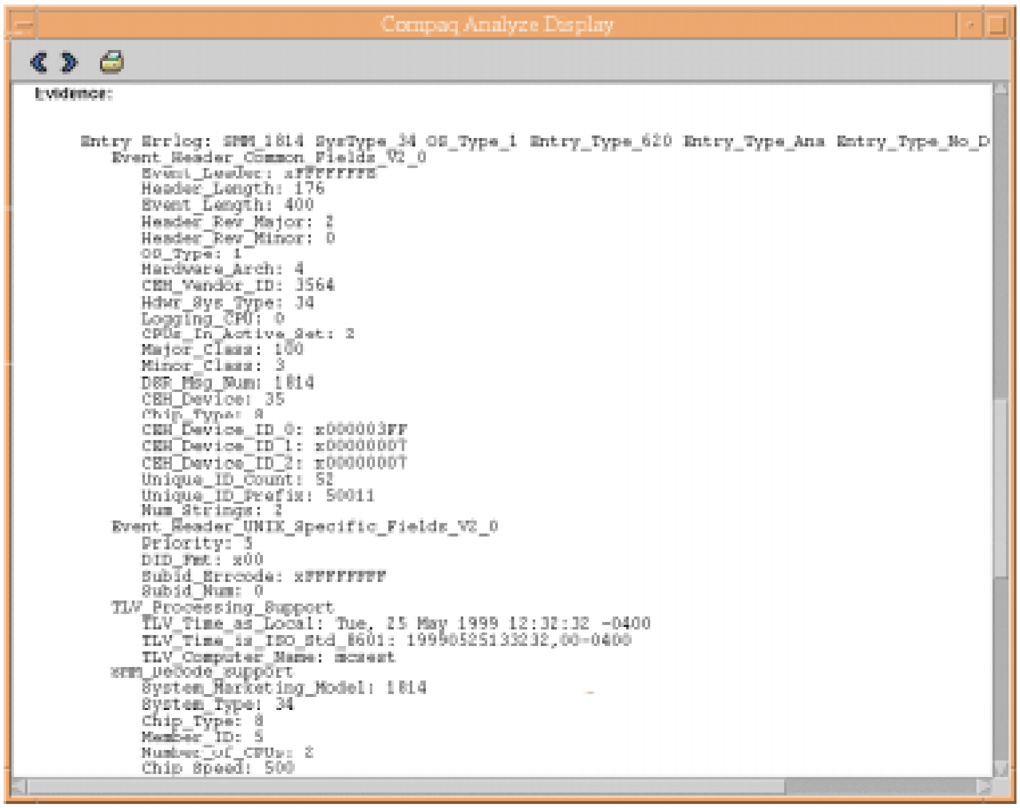
FRU List

The FRU List designator lists the most probable defective FRUs. This list indicates that service needs to be administered to one or more of these FRUs. The information typically include the FRU probability, manufacturer, system device type, system physical location, part number, serial number, and firmware revision level (if applicable).

In Figure 5–4 the most probable failing FRU is DIMM 3 on MMB1. The next less probable is the system motherboard, and the least probable is MMB1.

Continued on next page

Figure 5–5 Evidence Designator



Evidence

The Evidence designator provides information that leads Compaq Analyze to identify the failing FRU and its location. A portion of the Evidence designator is shown in Figure 5–5. The evidence provided depends on the type of error that is detected. The error types are:

CPU Correctable Error (630)
CPU Uncorrectable Error (670)
System Correctable Error (620)
System Uncorrectable Error (660)
System Correctable Environmental (680)

Brief descriptions of the errors in these categories are given in Section 5.3. See Appendix D for the source data Compaq Analyze uses to isolate to the FRUs.

The Evidence designator provides a hex dump of the error event information that triggered the indictment. The evidence is broken into segments and described as follows:

- Common Event Header—Provides information about the event as it was logged into the binary error log by the operating system.
- Logout Frame—Provides the actual system error state capture information like EV6 (21264) and System (21272 Tsunami/Typhoon).
- Appended Error Subpackets—Provides additional error state or system configuration information required for diagnostic processing.

5.2 Fault Detection and Reporting

Table 5–1 provides a summary of the fault detection and correction components of *Compaq AlphaServer* ES40 systems.

Generally, PALcode handles exceptions/interrupts as follows:

1. The PALcode determines the cause of the exception/interrupt.
2. If possible, it corrects the problem and passes control to the operating system for error notification, reporting, and logging before returning the system to normal operation.

   If PALcode is unable to correct the problem, it

   - Logs double error halt error frames into the flash ROM
   - Logs uncorrectable error logout frames to the DPR
   - For single halts, logs the uncorrectable logout frame into the DPR.
3. If error/event logging is required, control is passed through the OS Privileged Architecture Library (PAL) handler. The operating system error handler logs the error condition into the binary error log. Compaq Analyze should then diagnose the error to the defective FRU.

Table 5–1 Compaq AlphaServer ES40 Fault Detection and Correction

Component	Fault Detection/Correction Capability
Alpha 21264 (EV6) microprocessor	Contains error checking and correction (ECC) logic for data cycles. Check bits are associated with all data entering and exiting the microprocessor. A single-bit error on any of the four longwords being read can be corrected (per cycle). A double-bit error on any of the four longwords being read can be detected (per cycle).
Backup cache (B-cache)	ECC check bits on the data store, and parity on the tag address store and tag control store.
Memory DIMMs	ECC logic protects data by detecting and correcting data cycle errors. A single-bit error on any of the four longwords can be corrected (per cycle). A double-bit error on any of the four longwords being read can be detected (per cycle).
PCI SCSI controller adapter	SCSI data parity is generated.

5.3 Machine Checks/Interrupts

The exceptions that result from hardware system errors are called machine checks/interrupts. They occur when a system error is detected during the processing of a data request.

During the error-handling process, errors are first handled by the appropriate PALcode error routine and then by the associated operating system error handler. PALcode transfers control to the operating system through the PAL handler.

Table 5–2 lists the machine checks/interrupts that are related to error events. The designations — 630, 670, 620, 660, and 680 — indicate a system control block (SCB) offset to the fatal system error handler for Tru64 UNIX and OpenVMS. Windows NT does not use SCB offsets, but instead uses a self-maintained interrupt dispatch table (IDT).

Table 5–2 Machine Checks/Interrupts

Error Type	Error Descriptions
CPU Correctable Error (630) Generic Alpha 21264 (EV6) correctable errors.	B-cache probe hit single-bit ECC error D-cache tag parity error on issue I-cache tag or data parity error D-cache victim single-bit ECC error B-cache single-bit ECC fill error to I-stream or D-stream Memory single-bit ECC fill error to I-stream or D-stream
CPU Uncorrectable Error (670) Fatal microprocessor machine check errors that result in a system crash.	PAL detected bugcheck error Operating system detected bugcheck error EV6 detected second D-cache store EEC error EV6 detected D-cache tag parity error in pipeline 0 or 1 EV6 detected duplicate D-cache tag parity error EV6 detected double-bit ECC memory fill error EV6 detected double-bit probe hit EEC error EV6 detected B-cache tag parity error

Table 5–2 Machine Checks/Interrupts (Continued)

Error Type	Error Descriptions
System Correctable Error (620) ES40-specific correctable errors.	System detected ECC single-bit error
System Uncorrectable Error (660) A system-detected machine check that occurred as a result of an “off-chip” request to the system.	Uncorrectable ECC error Nonexistent memory reference PCI system bus error (SERR) PCI read data parity error (RDPE) PCI address/command parity error (APE) PCI no device select (NDS) PCI target abort (TA) Invalid scatter/gather page table entry (SGE) error PCI data parity error (PERR) Flash ROM write error PCI target delayed completion retry time-out (DCRTO) PCI master retry time-out (RTO 2**24) error PCI-ISA software NMI error
System Environmental Error (680) System-detected machine check caused by an overtemperature condition, fan failure, or power supply failure.	Overtemperature failure (>50•C) (see Note) Uncorrectable Fan 5 failure Complete power supply failure Fan failure (redundant fan) Power supply failure (redundant supply) High temperature warning (>45• C and <50• C)

NOTE: *For overtemperature failure, the position of jumper J26 determines whether the failure is fatal or nonfatal. See Appendix B.*

5.3.1 Error Logging and Event Log Entry Format

The operating system error handlers generate several entry types. Entries can be of variable length based on the number of registers within the entry.

Each entry consists of an operating system header, several device frames, and an end frame. Most entries have a PAL-generated logout frame, and may contain frames for CPU, memory, and I/O.

Table 5–3 shows an event structure map for a Windows NT system uncorrectable PCI target abort error.

NOTE: *See Appendix D for the source data Compaq Analyze uses to isolate to the FRUs.*

Table 5–3 Sample Error Log Event Structure Map (ES40 with 10 PCI Slots)

OFFSET(hex)	63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
nh0000 nh+nnnn	STANDARD MICROSOFT NT OS HEADER							
ech0000 ech+nnnn	NEW COMMON OS HEADER							
lfh0000 lfh+nnnn	STANDARD LOGOUT FRAME HEADER							
lfev60000 lfev6+nnnn	COMMON PAL EV6 SECTION (first 8 QWs Zeroed)							
lfctt_A0[u]	SESF<63:32> = Reserved(MBZ)			<39:32>= (MBZ)	SESF<31:16> = Reserved(MBZ)		SESF<15:0>= 0002(hex)	
lfctt_A8[u]	Cchip CPUx Device Interrupt Request Register (DIRx<61> = 1)							
lfctt_B0[u]	Cchip Miscellaneous Register (MISC)							
lfctt_B8[u]	Pchip0 Error Register (P0_PERROR<63:0> = 0)							
lfctt_C0[u]	Pchip1 Error Register (P1_PERROR<51>=0;<47:18>=PCI Addr;<17:16>=PCI Opn; <6>=1)							
lfett_C8[u] lfett_138[u]	Pchip1 Extended Tsunami/Typhoon System Packet							
eelcb_140	Pchip 1 PCI Slot 4 Single Device Bus Snapshot Packet							
eelcb_190	Pchip 1 PCI Slot 5 Single Device Bus Snapshot Packet							
eelcb_1E0	Pchip 1 PCI Slot 6 Single Device Bus Snapshot Packet							
eelcb_230	Pchip 1 PCI Slot 7 Single Device Bus Snapshot Packet							
eelcb_280	Pchip 1 PCI Slot 8 Single Device Bus Snapshot Packet							
eelcb_2D0	Pchip 1 PCI Slot 9 Single Device Bus Snapshot Packet							
2D8	Termination or End Packet							

5.4 Environmental Errors Captured by SRM

If an environmental error occurs while the SRM console is running, a logout frame similar to Example 5–1 is sent to the console output device. The logout frame is preceded by the message "*unexpected system event through vector 680 on CPU *n*." (usually CPU 0.) For register definitions see Appendix D.**

Example 5–1 Console Level Environmental Error Logout Frame

```
P00>>>
*** unexpected system event through vector 680 on CPU 0
os_flags        0000000000000000
cchip_dirx      0004000000000000
tig_smir        0000000000000008
tig_cpuir       000000000000000f
tig_psir        0000000000000003
lm78_isr        0000000000000000
door_open       0000000000000004                    ❶
temp_warning    0000000000000000
fan_ctrl_fault  0000000000000000
power_down_code 0000000000000000
reserved_1      0000000000000000
```

❶ This example shows a fan door open event.

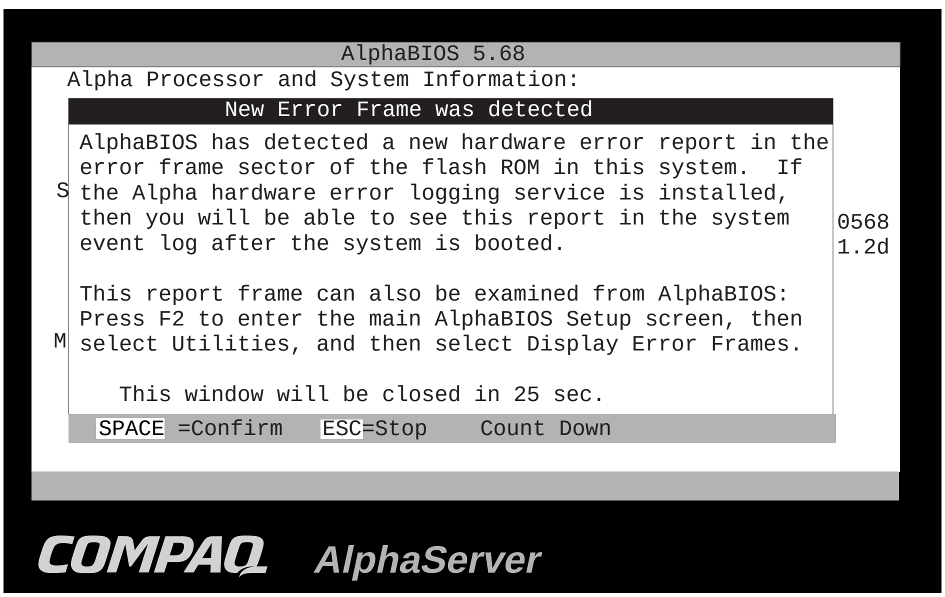
```
P00>>>
*** unexpected system event through vector 680 on CPU 0
os_flags        0000000000000000
cchip_dirx      0004000000000000
tig_smir        0000000000000008
tig_cpuir       000000000000000f
tig_psir        0000000000000003
lm78_isr        0000000000000000
door_open       0000000000000040                        ➊
temp_warning    0000000000000000
fan_ctrl_fault  0000000000000000
power_down_code 0000000000000000
reserved_1      0000000000000000
```

➊ This example shows a fan door closing event.

5.5 Windows NT Error Logs

The Display Error Frames selection of the AlphaBIOS Utilities menu allows you to view hardware error reports for systems running Windows NT. A report is generated if a fatal error or double error halt occurs. If the System Error Logging Software for Alpha kit is installed, you will be able to see the report in the system event log after the system has booted.

Figure 5–6 New Error Frame Was Detected Window



PK0955

The next time you boot the system after a fatal error or double error halt, AlphaBIOS displays the message shown in Figure 5–6 just after initialization has been completed and just before the Boot menu is displayed. The message is closed after 30 seconds. To keep the message window open, press the ESC key before the count down time has elapsed.

Fatal Error Halts

Fatal error halts are single errors that occur when the operating system is running. Only one operating system fatal (OS fatal) error at a time can exist in flash ROM. When a new OS fatal error occurs, it replaces the old error in the flash.

Double Error Halts

Double error halts are conditions in which the processing of a fatal error triggers a second error. Two varieties of double error halt errors can occur, based on which code is executing when the second error occurs: machine checks in PALmode or double errors (HAL code). Double error halts can occur on multiple processors at the same time. As a result, multiple double error halt logs can be generated at the same time and possibly in concert with another single fatal or correctable error log.

For both single and double error halts, if the System Error Logging Software for Alpha kit is installed, the next operating system boot causes the new error frame to be copied automatically to the Windows NT event log for viewing and analysis.

NOTE: *The System Error Logging Software for Alpha kit is provided on the platform OEM floppy and with the HAL updates on the World Wide Web:*

http://www.compaq.com/support/files/alphant/index.html

The software works with the operating system layer to ensure that errors and FRU table information are logged in the event log. It also provides correctable error throttling and user notification for environmental warnings. In addition, the kit provides an API for Compaq Analyze to log information to the FRU EEPROMs by means of the DPR.

Continued on next page

Figure 5–7 Display Error Frames Screen

```
                         Display Error Frames                    F1=Help

 1. Fatal Error Frame                  [NEW]: 576 bytes
 2. Double Error Halt (CPUID=01)       [OLD]: 240 bytes

                  Help: Display Error Frames           F1=Key Help

                ENTER       View the formatted error frame.

                F6          View the binary dump of the error frame.

                F10         Save the error frame to floppy disk.

                DEL         Delete the error frame from the flash ROM.

                ESC         Exit.

                 ENTER=Continue
 If the Alpha hardware error logging service is installed in this system, the
 next operating system boot will cause the error frame to be copied to the
 system event log automatically.
 Press ENTER to display the error frame.
```

Displaying an Error Frame

1. To display the error frame, enter AlphaBIOS Setup and select the **Utilities** menu.
2. From the Utilities menu, select **Display Error Frames**….

   If there is no error frame in the flash ROM, a screen with the message "No Error Frame in the flash ROM" is displayed. If there is an error frame, a screen similar to Figure 5–7 is displayed.

Figure 5–7 shows two error frames:

- "Fatal Error Frame [NEW]" is a new error frame that has not yet been copied to the system event log for analysis. If the System Error Logging Software for Alpha kit is installed, you can view the error frame in the system event log at the next operating system boot.
- "Double Error Halt [OLD]" is an old error frame that was previously copied to the system event log for analysis.

Clearing an Error Frame Log from Flash

Error frame logs remain in flash ROM and can be viewed through the AlphaBIOS error log browser until one of the following occurs:

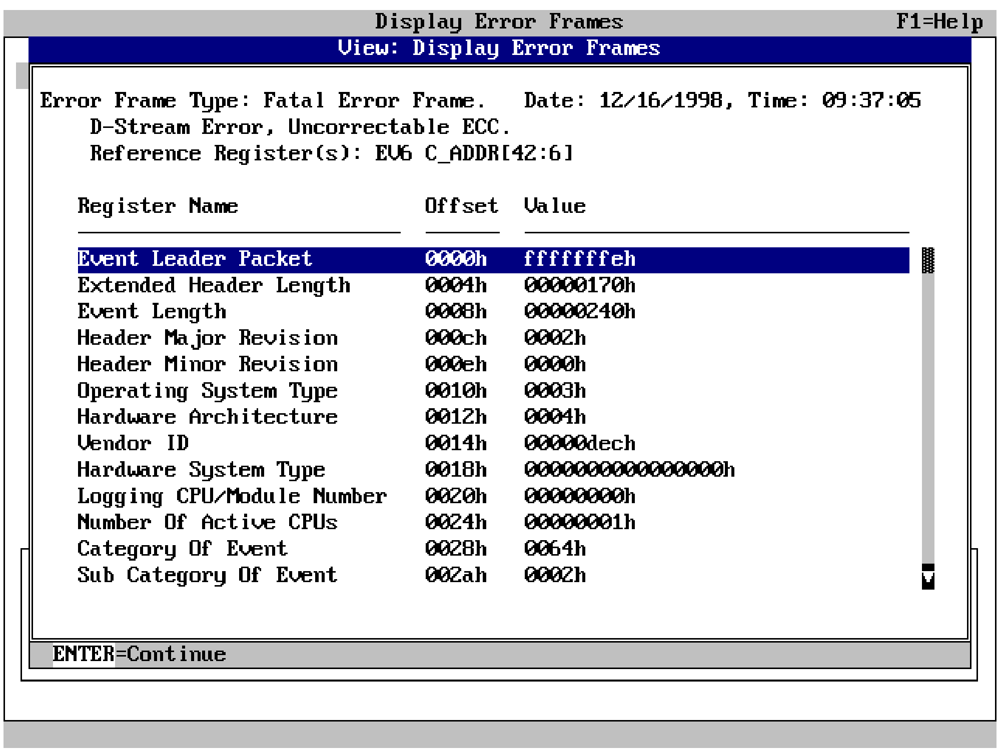
- A new error occurrence generates a new log that replaces an old one
- The user manually deletes a log

An error log might also be removed from flash if AlphaBIOS is upgraded to a newer version that has changed the error log browsing code based on an error frame version. Older error frame logs are deleted if they cannot be read by the new code.

5.5.1 Viewing a Formatted Text-Style Error Frame

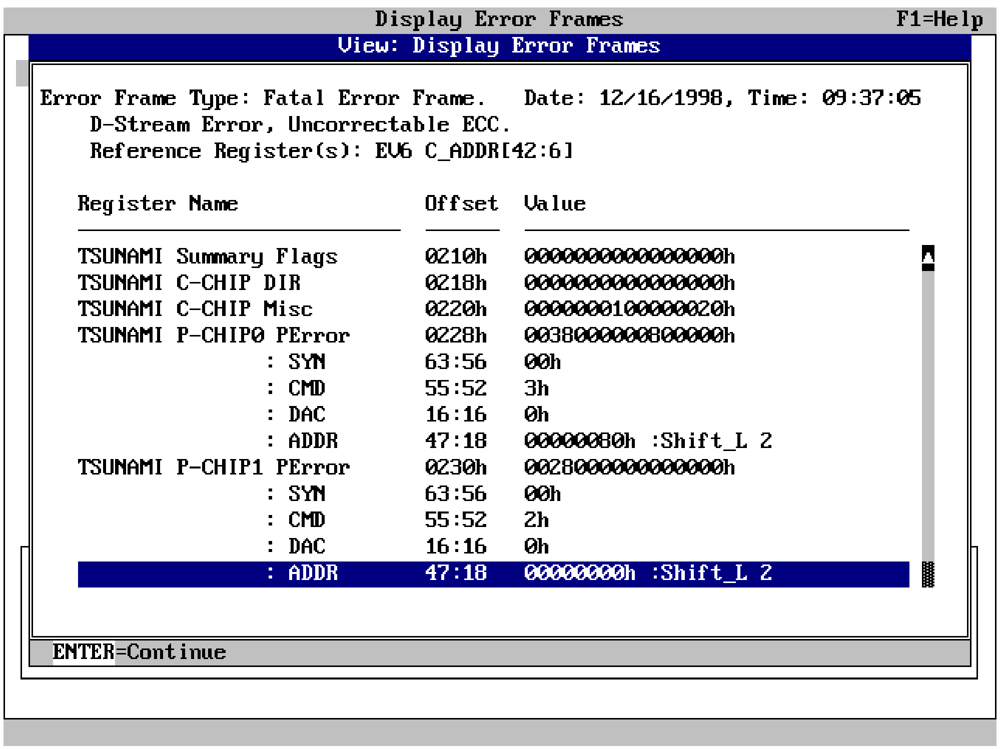
Press the Enter key to view a formatted text-style error frame. The error source is also displayed. For example, the Fatal Error Frame in Figure 5–8 reports a "D-Stream Error, Uncorrectable ECC."

Figure 5–8 View by Formatted Text Style



You can browse the entire contents of an error log by using the scroll bar, as shown in Figure 5–9.

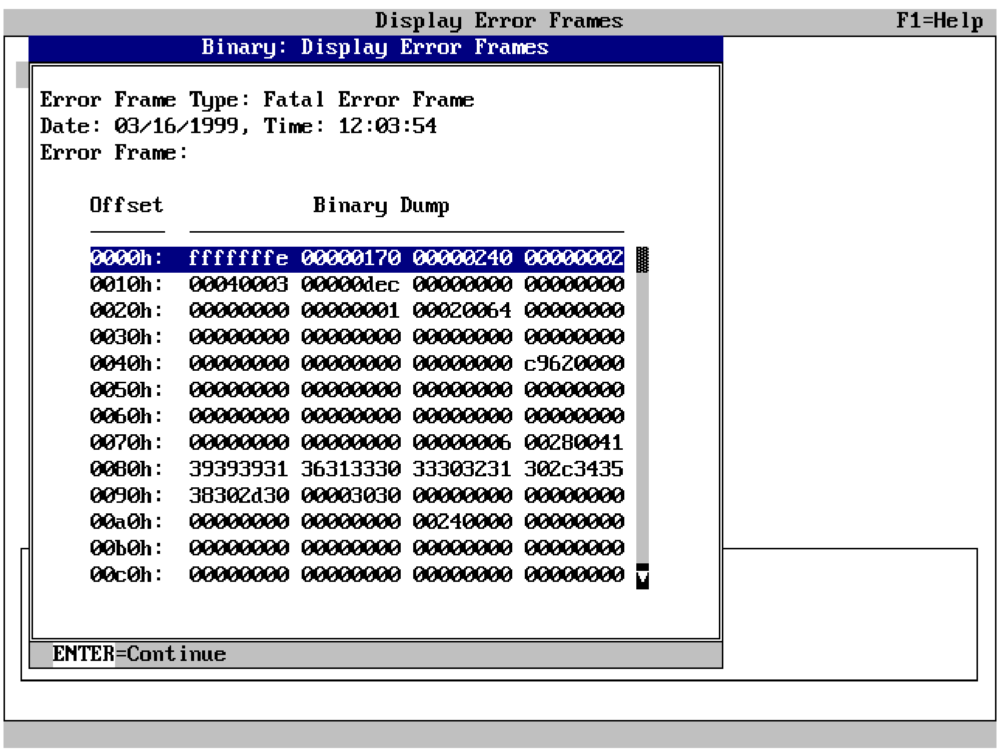
Figure 5–9 Browsing Error Logs



5.5.2 Viewing a Binary Dump of the Error Frame

Press the F6 key to get a binary dump of the entire error frame.

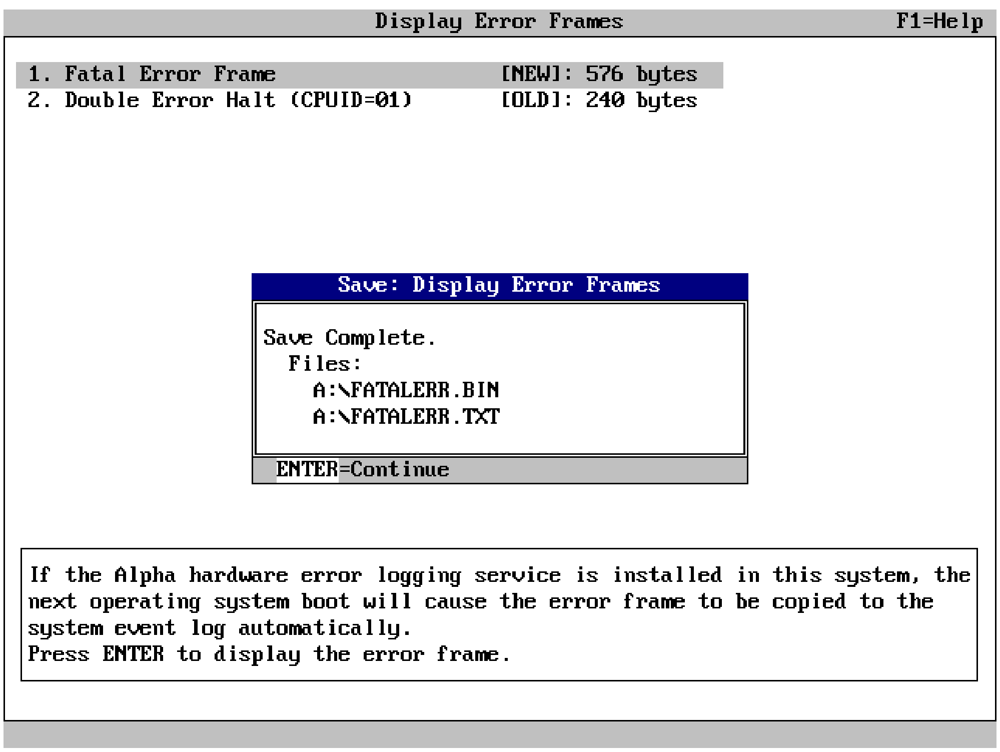
Figure 5–10 Binary Dump of Error Frame



5.5.3 Saving the Error Frame to the Floppy

Press F10 to save the error frame to the floppy. For the formatted text style, an ASCII (text) file is generated. For the binary dump, a raw file is generated. If the same file name already exists on the floppy, a warning message is displayed. Press Enter to continue the save.

Figure 5–11 Save to the Floppy



Continued on next page

The OS fatal and double error halt files are named as follows. The <*cpuNumber*> is two digits.

Type of Error Frame	File Name
Fatal error frame (Binary)	FATALERR.BIN
Fatal error frame (ASCII)	FATALERR.TXT
Double error frame (Binary)	DBLERR<*cpuNumber*>.BIN
Double error frame (ASCII)	DBLERR<*cpuNumber*>.TXT

Figure 5–12 shows an example of a formatted text file.

Figure 5–12 Formatted Text File

```
Error Frame Type: Fatal Error Frame.   Date: 12/04/1998, Time: 03:15:46
    D-Stream Error, Uncorrectable ECC.
    Reference Register(s): EV6 C_ADDR[42:6]

Register Name               Offset  Value
--------------------------  ------  ----------------------------
Event Leader Packet         0000h   fffffffeh
Extended Header Length      0004h   00000170h
Event Length                0008h   00000240h
Header Major Revision       000ch   0002h
Header Minor Revision       000eh   0000h
Operating System Type       0010h   0003h
Hardware Architecture       0012h   0004h
Vendor ID                   0014h   00000dech
Hardware System Type        0018h   0000000000000000h
Logging CPU/Module Number   0020h   00000000h
Number Of Active CPUs       0024h   00000001h
Category Of Event           0028h   0064h
Sub Category Of Event       002ah   0002h
DSR Number                  002ch   00000000h
Device                      0030h   0000h
Priority                    0032h   00h
DidFmt                      0033h   00h
SubID ErrCode               0034h   00000000h
SubID Num                   0038h   00000000h
Chip Type                   003ch   00000000h
Device ID 0                 0040h   00000000h
Device ID 1                 0044h   00000000h
Device ID 2                 0048h   00000000h
Universally Unique ID       004ch   76ed0000h
Reserved [0]                0050h   0000000000000000h
Reserved [1]                0058h   0000000000000000h
Reserved [2]                0060h   0000000000000000h
Reserved [3]                0068h   0000000000000000h
Reserved [4]                0070h   0000000000000000h
```

```
Number of TLVs in header     0078h   00000006h
Wall-Clock Time (Tag)        007ch   0041h
Wall-Clock Time (Length)     007eh   0028h
Wall-Clock Time (String)     0080h   "19981204031546,00-0800"
DSR (Tag)                    00a8h   0000h
DSR (Length)                 00aah   0024h
DSR (String)                 00ach   ""
OS Version (Tag)             00d0h   0081h
OS Version (Length)          00d2h   0024h
OS Version (String)          00d4h   "Windows NT 4.00"
OS Build Number (Tag)        00f8h   00a1h
OS Build Number (Length)     00fah   0024h
OS Build Number (String)     00fch   "Build Number 1381"
System Serial Num.(Tag)      0120h   0000h
System Serial Num.(Length)   0122h   0024h
System Serial Num.(String)   0124h   ""
System Name (Tag)            0148h   0124h
System Name (Length)         014ah   0024h
System Name (String)         014ch   "NTMASATO4"
EV6 MCHK Frame Size          0170h   000000c8h
EV6 MCHK Flags               0174h   00000000h
EV6 Processor Offset         0178h   00000018h
EV6 System Offset            017ch   000000a0h
EV6 Machine Check Code       0180h   00000098h
EV6 Frame Revision           0184h   00000001h
EV6 I_STAT 21264             0188h   0000000000000000h
EV6 DC_STAT 21264            0190h   0000000000000000h
EV6 C_ADDR                   0198h   0000000006c92080h
               : [42:6]      42:06   0000000006c92080h :Shift_L 6
               : [19:6]      19:06   00092080h :Shift_L 6
EV6 DC1_SYNDROME             01a0h   0000000000000005h
EV6 DC0_SYNDROME             01a8h   0000000000000000h
EV6 C_STAT                   01b0h   0000000000000010h
EV6 C_STS                    01b8h   0000000000000006h
EV6 MM_STAT                  01c0h   0000000000000280h
EV6 EXC_ADDR                 01c8h   0000000000403620h
EV6 IER_CM                   01d0h   0000000e8000c008h
EV6 I_SUM                    01d8h   0000000000000000h
EV6 PAL_BASE                 01e8h   0000000000680000h
EV6 I_CTL                    01f0h   ffffffff837d438fh
EV6 PCTX                     01f8h   0000000000000000h
TSUNAMI Summary Flags        0210h   0000000000000000h
TSUNAMI C-CHIP DIR           0218h   0000000000000000h
TSUNAMI C-CHIP Misc          0220h   0000000100000020h
TSUNAMI P-CHIP0 PError       0228h   0038000000800000h
               : SYN         63:56   00h
               : CMD         55:52   3h
               : DAC         16:16   0h
               : ADDR        47:18   00000080h :Shift_L 2
TSUNAMI P-CHIP1 PError       0230h   0008000000000000h
               : SYN         63:56   00h
               : CMD         55:52   0h
               : DAC         16:16   0h
               : ADDR        47:18   00000000h :Shift_L 2
```

5.5.4 Deleting an Error Frame

Use the DEL key to delete the error frame from the flash ROM. If you delete a new error frame, a warning message is displayed, as shown in Figure 5–13. If you delete an old error frame, a message similar to that in Figure 5–14 is displayed. Press F10 to continue a deletion. When the deletion is complete, a "Delete Complete" message is displayed.

Figure 5–13 Deleting a New Error Frame

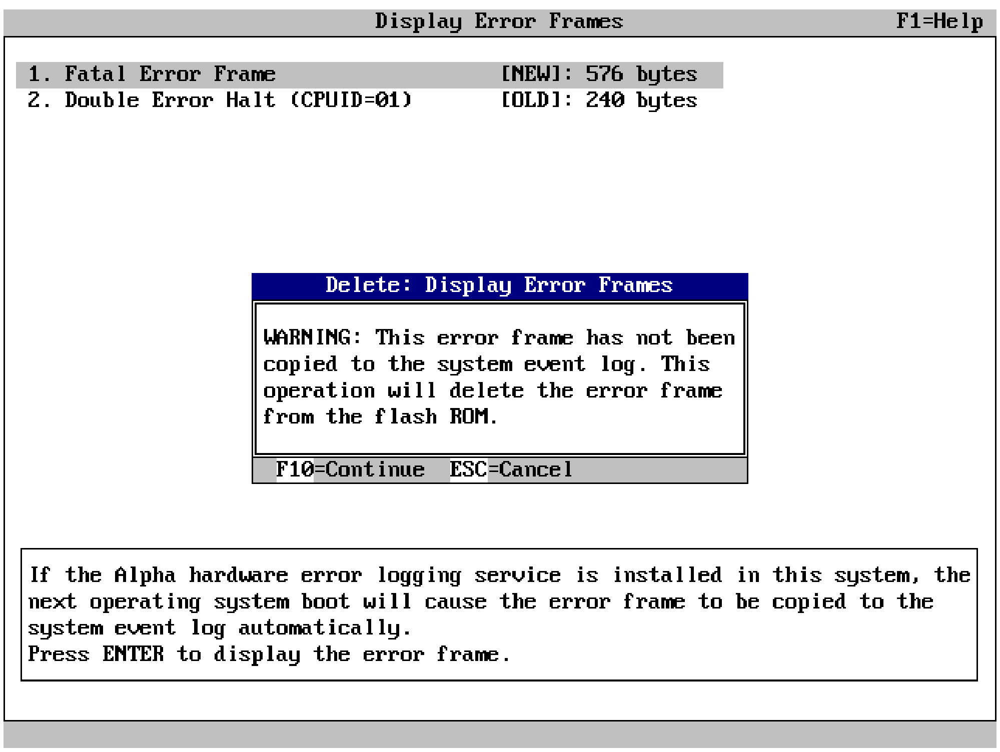
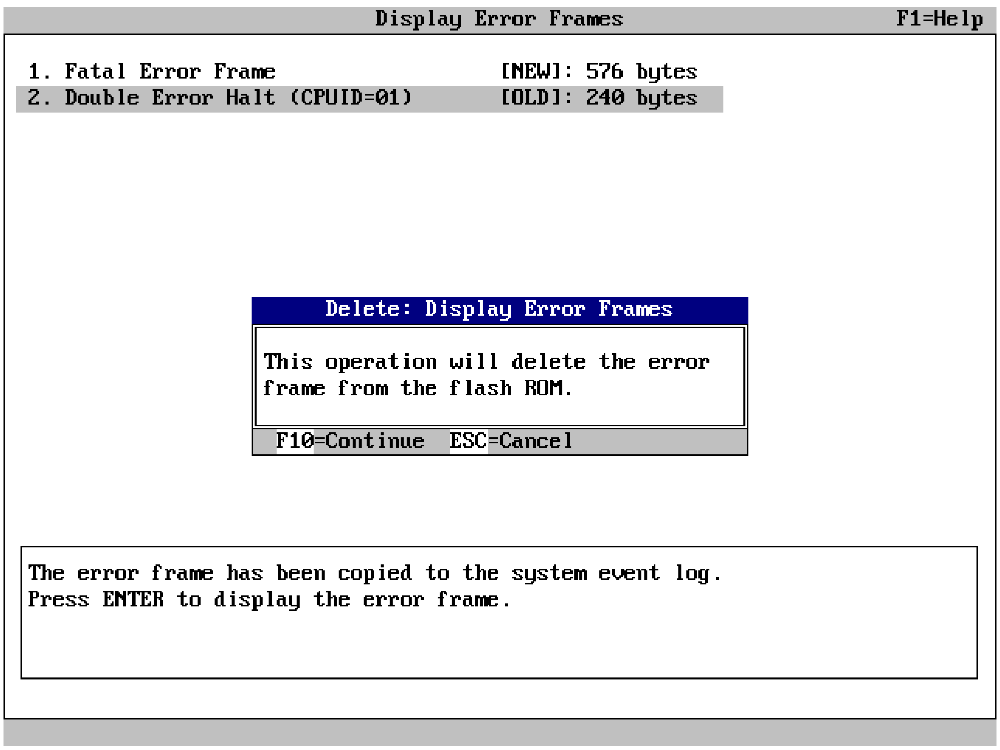


Figure 5–14 Deleting an Old Error Frame



Chapter 6
System Configuration and Setup

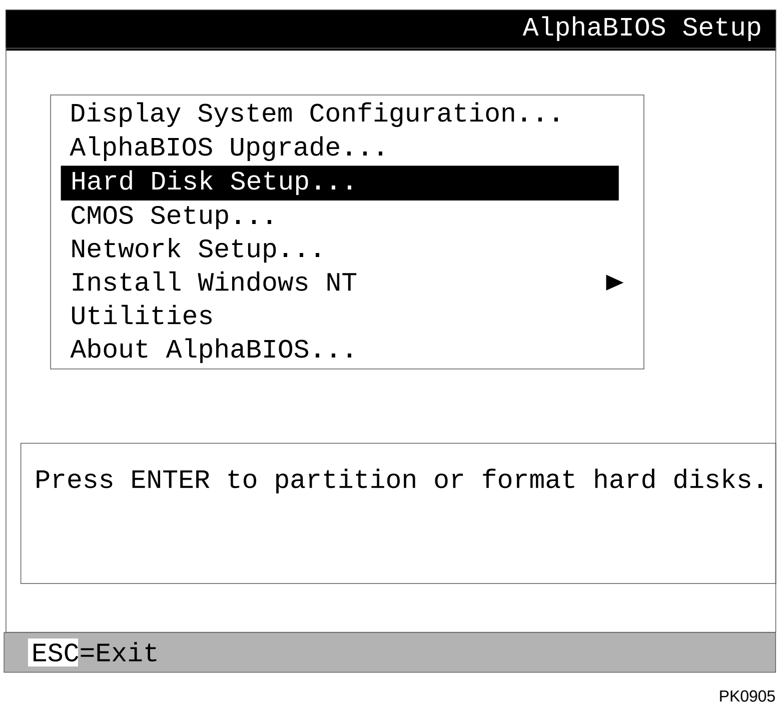
This chapter describes how to configure and set up *Compaq AlphaServer* ES40 systems. The following topics are covered:

- System Consoles
- Displaying the Hardware Configuration
- Setting Environment Variables for Tru64 UNIX or OpenVMS
- Setting Up a System for Windows NT
- Setting Automatic Booting
- Changing the Default Boot Device
- Running AlphaBIOS-Based Utilities
- Setting SRM Security
- Setting Windows NT Security
- Configuring Devices
- Switching Between Operating Systems

6.1 System Consoles

System console programs are located in a flash ROM on the system motherboard. From the console interface, you can set up and boot the operating system, display the system configuration, and run diagnostics. For complete information on the SRM and AlphaBIOS consoles, see the *Compaq AlphaServer ES40 User Interface Guide*.

Figure 6–1 AlphaBIOS Setup Screen



SRM Console

Systems running the Tru64 UNIX or OpenVMS operating systems are configured from the SRM console, a command-line interface (CLI). From the CLI you can enter commands to configure the system, view the system configuration, boot the system, and run ROM-based diagnostics.

AlphaBIOS Console

Systems running the Windows NT operating system are configured from the AlphaBIOS console, a menu interface. From the AlphaBIOS boot screen, you can boot the operating system or press **F2** to enter a setup screen to set up the system. The Setup screen is shown in Figure 6–1. From the Utilities menu on the Setup screen, you can select options to run maintenance programs and display error frames for hardware errors logged to the flash ROM.

RMC CLI

The remote management console (RMC) provides a command-line interface (CLI) for controlling the system. You can use the CLI either locally or remotely (modem connection) to power the system on and off, halt or reset the system, and monitor the system environment. You can also use the **dump**, **env**, and **status** commands to help diagnose errors. See Chapter 7 for details.

6.1.1 Switching Between Consoles

Under some circumstances, you may need to switch between the system consoles. For example, error frames for Windows NT systems are viewed from the AlphaBIOS console.

Figure 6–2 Invoking SRM from AlphaBIOS

```
                         Advanced CMOS Setup       F1=Help

 PCI Parity Checking:  Disabled

 Power-up Memory Test:  Partial

 AlphaBIOS Password Option:  Disabled

 Console Selection:   Digital UNIX Console (SRM)

                     Windows NT Console (AlphaBIOS)
                     OpenVMS Console (SRM)
                     Tru64 UNIX Console (SRM)


 Press ↑ or ↓ to select the firmware console that will be
 presented the next time the system is power-cycled.


 ESC=Discard Changes F10=Save Changes
```

PK0924

- To enter the SRM console from Windows NT, shut down the operating system and wait for the message indicating is it safe to power off the system. Next, press the Reset button, and then press the Halt button. You can also enter SRM by changing the Console Selection option on the AlphaBIOS Advanced CMOS Setup screen. See Figure 6–2.
- To enter the AlphaBIOS console from SRM, issue the **alphabios** command:

```
P00>>> alphabios
```

6.1.2 Selecting the Console and Display Device

The SRM os_type environment variable determines which user interface (SRM or AlphaBIOS) is the final console loaded on a power-up or reset. The SRM console environment variable determines to which display device (VT-type terminal or VGA monitor) the console display is sent.

Selecting the Console

The **os_type** variable selects the console. **Os_type** is factory configured as follows:

- For Windows NT, **os_type** is set to **nt**.
- For UNIX or OpenVMS, **os_type** is set to **unix** or **vms**, respectively.

If **os_type** is set to **unix** or **vms**, the SRM console is loaded on a power-up or reset. If **os_type** is set to **nt**, the SRM console is loaded and then SRM starts the AlphaBIOS console from system flash ROM.

Selecting the Display Device

The console terminal that displays the SRM user interface can be either a serial terminal (VT320 or higher, or equivalent) or a VGA monitor. A VGA monitor is required to run Windows NT.

The SRM **console** environment variable determines the display device.

- If **console** is set to **serial**, and a VT-type device is connected, the SRM console powers on in serial mode and sends power-up information to the VT device. The VT device can be connected to the MMJ port or to COM2.
- If **console** is set to **graphics**, the SRM console expects to find a VGA card connected to PCI 0 and, if so, displays power-up information on the VGA monitor after VGA initialization has been completed.

Continued on next page

You can verify the display device with the SRM **show console** command and change the display device with the SRM **set console** command. If you change the display device setting, you must reset the system (with the Reset button or the **init** command) to put the new setting into effect.

In the following example, the user displays the current console device (a graphics device) and then resets it to a serial device. After the system initializes, output will be displayed on the serial terminal.

```
P00>>> show console
console                 graphics
P00>>> set console serial
P00>>> init
.
.
.
```

6.1.3 Setting the Control Panel Message

If you are running Tru64 UNIX or OpenVMS, you can create a customized message to be displayed on the operator control panel after startup self-tests and diagnostics have been completed.

When the operating system is running, the control panel displays the console revision. It is useful to create a customized message if you have a number of systems and you want to identify each system by a node name.

You can use the SRM **set ocp_text** command to change this message (see Example 6–1). The message can be up to 16 characters and must be entered in quotation marks.

Example 6–1 set ocp_text

```
P00>>> set ocp_text “Node Alpha1”
```

6.2 Displaying the Hardware Configuration

View the system hardware configuration for UNIX and OpenVMS systems from the SRM console. View a Windows NT hardware configuration from the AlphaBIOS console. It is useful to view the hardware configuration to ensure that the system recognizes all devices, memory configuration, and network connections.

Displaying a Tru64 UNIX or OpenVMS Configuration

Use the following SRM console commands to view the system configuration for UNIX or OpenVMS systems. See the *Compaq AlphaServer ES40 User Interface Guide* for details.

show boot*	Displays the boot environment variables.
show config	Displays the logical configuration of interconnects and buses on the system and the devices found on them.
show device	Displays the bootable devices and controllers in the system.
show fru	Displays the physical configuration of FRUs (field-replaceable units).
show memory	Displays configuration of main memory.

Displaying a Windows NT Hardware Configuration

View a Windows NT configuration as follows:

1. From the AlphaBIOS Setup screen, select **Display System Configuration** and press Enter.
2. In the Display System Configuration screen, use the arrow keys to select the configuration category you want to see.

Figure 6–3 Display System Configuration Screen

```
                         Display System Configuration

  Systemboard Configuration
  Hard Disk Configuration
  PCI Configuration
  SCSI Configuration
  Memory Configuration
  Integrated Peripherals

      System Type:  AlphaServer ES40

   Processor:   Alpha 21264, Revision 4.0 (4 Processors)
       Speed:   500 MHz
       Cache:   4 MB
      Memory:   2048 MB

   Floppy Drive A:  3.5" 1.44 MB
   Floppy Drive B:  None
         Keyboard:  U.S. 101-key keyboard

   AlphaBIOS Version:  5.68

  ESC=Exit
```

PK0902

6.3 Setting Environment Variables for Tru64 UNIX or OpenVMS

Environment variables pass configuration information between the console and the operating system. Their settings determine how the system powers up, boots the operating system, and operates.

- To check the setting for a specific environment variable, enter the **show *envar*** command, where the name of the environment variable is substituted for *envar*.
- To reset an environment variable, use the **set *envar*** command, where the name of the environment variable is substituted for *envar*.

set *envar*

The **set** command sets or modifies the value of an environment variable. It can also be used to create a new environment variable if the name used is unique. Environment variables pass configuration information between the console and the operating system. Their settings determine how the system powers up, boots the operating system, and operates. The syntax is:

set *envar value*

envar	The name of the environment variable to be modified.
value	The new value of the environment variable.

New values for the following environment variables take effect only after you reset the system by pressing the Reset button or issuing the **init** command.

auto_action
console
cpu_enabled
os_type
pk*0_fast
pk*0_host_id
pk*0_soft_term

show *envar*

The **show *envar*** command displays the current value (or setting) of an environment variable. The syntax is:

show *envar*

envar	The name of the environment variable to be displayed. The wildcard * displays all environment variables.

Table 6–1 summarizes the SRM environment variables used most often on the ES40 system.

Table 6–1 SRM Environment Variables Used on ES40 Systems

Variable	Attributes	Description
auto_action	NV,W[1]	Action the console should take following an error halt or power failure. Defined values are: **boot**—Attempt bootstrap. **halt**—Halt, enter console I/O mode. **restart**—Attempt restart. If restart fails, try boot.
bootdef_dev	NV,W	Device or device list from which booting is to be attempted when no path is specified. Set at factory to disk with factory-installed software; otherwise **NULL**.
boot_file	NV,W	Default file name used for the primary bootstrap when no file name is specified by the **boot** command. The default value is **NULL**.
boot_osflags	NV,W	Default parameters to be passed to system software during booting if none are specified by the **boot** command.
		OpenVMS: Additional parameters are the *root_number* and *boot flags*. The default value is **NULL**.
		root_number: Directory number of the system disk on which OpenVMS files are located.
		0 (default)—[SYS0.SYSEXE] 1—[SYS1.SYSEXE] 2—[SYS2.SYSEXE] 3—[SYS3.SYSEXE]

[1] NV—Nonvolatile. The last value saved by system software or set by console commands is preserved across cold bootstraps (when the system goes through a full initialization), and long power outages.

W—Warm nonvolatile. The last value set by system software is preserved across warm bootstraps (UNIX **shutdown -r** command, OpenVMS **REBOOT** command, or a crash and reboot; not all of the SRM initialization is run) and restarts.

Table 6–1 SRM Environment Variables Used on ES40 Systems (Continued)

Variable	Attributes	Description
boot_osflags (continued)	NV,W	*boot_flags:* The hexadecimal value of the bit number or numbers to set. To specify multiple boot flags, add the flag values (logical OR). 1—Bootstrap conversationally (enables you to modify SYSGEN parameters in SYSBOOT). 2—Map XDELTA to running system. 4—Stop at initial system breakpoint. 8—Perform a diagnostic bootstrap. 10—Stop at the bootstrap breakpoints. 20—Omit header from secondary bootstrap file. 80—Prompt for the name of the secondary bootstrap file. 100—Halt before secondary bootstrap. 10000—Display debug messages during booting. 20000—Display user messages during booting. **Tru64 UNIX:** The following parameters are used with this operating system: **a**—Autoboot. Boots /vmunix from bootdef_dev, goes to multi-user mode. Use this for a system that should come up automatically after a power failure. **s**—Stop in single-user mode. Boots /vmunix to single-user mode and stops at the # (root) prompt. **i**—Interactive boot. Requests the name of the image to boot from the specified boot device. Other flags, such as -kdebug (to enable the kernel debugger), may be entered using this option.

Table 6–1 SRM Environment Variables Used on ES40 Systems (Continued)

Variable	Attributes	Description
boot_osflags (continued)		**D**—Full dump; implies **s** as well. By default, if Tru64 UNIX crashes, it completes a partial memory dump. Specifying **D** forces a full dump at system crash. Common settings are **a**, autoboot, and **Da**, autoboot and create full dumps if the system crashes.
com1_baud	NV,W	Sets the baud rate of the COM1 (MMJ) port. The default baud rate is 9600. Baud rate values are 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 38400, 57600.
com2_baud	NV,W	Sets the baud rate of the COM2 port. The default baud rate is 9600. Baud rate values are 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 38400, 57600.
com1_flow **com2_flow**	NV,W	The **com1_flow** and **com2_flow** environment variables indicate the flow control on the serial ports. Defined values are: **none**—No data flows in or out of the serial ports. Use this setting for devices that do not recognize XON/XOFF or that would be confused by these signals. **software**—Use XON/XOFF(default). This is the setting for a standard serial terminal. **hardware**—Use modem signals CTS/RTS. Use this setting if you are connecting a modem to a serial port.
com1_mode	NV	Specifies the COM1 data flow paths so that data either flows through the RMC or bypasses it.

Table 6–1 SRM Environment Variables Used on ES40 Systems (Continued)

Variable	Attributes	Description
com1_modem **com2_modem**	NV,W	Used to tell the operating system whether a modem is present on the COM1 or COM2 ports, respectively **On**—Modem is present. **Off**—Modem is not present (default value).
console	NV	Sets the device on which power-up output is displayed. **Graphics**—Sets the power-up output to be displayed at a VGA monitor or device connected to the VGA module. **Serial**—Sets the power-up output to be displayed on the device that is connected to the COM1 (MMJ) port.
cpu_enabled	NV	Enables or disables a specific secondary CPU. All CPUs are enabled by default. The primary CPU cannot be disabled. The primary CPU is the lowest numbered working CPU.
ei*0_inet_init or **ew*0_inet_init**	NV	Determines whether the interface's internal Internet database is initialized from nvram or from a network server (via the bootp protocol).
ei*0_mode or **ew*0_mode**	NV	Sets the Ethernet controller to the default Ethernet device type. **aui**—Sets the default device to AUI. **bnc**—Sets the default device to ThinWire. **fast**—Sets the default device to fast 100BaseT. **fastfd**—Sets the default device to fast full duplex 100BaseT. **full**—Set the default device to full duplex twisted pair.

Table 6–1 SRM Environment Variables Used on ES40 Systems (Continued)

Variable	Attributes	Description
ei*0_mode or **ew*0_mode** (continued)		**twisted-pair**— Sets the default device to 10BaseT (twisted-pair).
ei*0_protocols or **ew*0_protocols**	NV	Determines which network protocols are enabled for booting and other functions. **mop**—Sets the network protocol to MOP for systems using the OpenVMS operating system. **bootp**—Sets the network protocol to bootp for systems using the Tru64 UNIX operating system. **bootp,mop**—When the settings are used in a list, the mop protocol is attempted first, followed by bootp.
heap_expand	NV	Increases the amount of memory available for the SRM console's heap. Valid selections are: NONE (default) 64KB 128KB 256KB 512KB 1MB 2MB 3MB 4MB
kbd_hardware type	NV	Sets the keyboard hardware type as either PCXAL or LK411 and enables the system to interpret the terminal keyboard layout correctly.
kzpsa_host_id	W	Specifies the default value for the KZPSA host SCSI bus node ID.

Table 6–1 SRM Environment Variables Used on ES40 Systems (Continued)

Variable	Attributes	Description
language	NV	Specifies the console keyboard layout. The default is English (American).
memory_test	NV	Specifies the extent to which memory will be tested on Tru64 UNIX. The options are: **Full**—Full memory test will be run. Required for OpenVMS. **Partial**—First 256 MB of memory will be tested. **None**—Only first 32 MB will be tested.
ocp_text	NV	Overrides the default control panel display text with specified text.
os_type	NV	Sets the default operating system. **vms or unix**—Sets system to boot the SRM firmware. **nt**—Sets system to boot the AlphaBIOS firmware.
password	NV	Sets a console password. Required for placing the SRM into secure mode.
pci_parity	NV	Disable or enable parity checking on the PCI bus. **On**—PCI parity enabled (default value) **Off**—PCI parity disabled Some PCI devices do not implement PCI parity checking, and some have a parity-generating scheme in which the parity is sometimes incorrect or is not fully compliant with the PCI specification. In such cases, the device functions properly so long as parity is not checked.

Table 6–1 SRM Environment Variables Used on ES40 Systems (Continued)

Variable	Attributes	Description
pk*0_fast	NV	Enables fast SCSI devices on a SCSI controller to perform in standard or fast mode. **0**—Sets the default speed for devices on the controller to standard SCSI. If a controller is set to standard SCSI mode, both standard and fast SCSI devices will perform in standard mode. **1**—Sets the default speed for devices on the controller to fast SCSI mode. Devices on a controller that connects to both standard and Fast SCSI devices will automatically perform at the appropriate rate for the device, either fast or standard mode.
pk*0_host_id	NV	Sets the controller host bus node ID to a value between 0 and 7. 0 to 7—Assigns bus node ID for specified host adapter.
pk*0_soft_term	NV	Enables or disables SCSI terminators for optional SCSI controllers. This environment variable applies to systems using the Qlogic SCSI controller, though it does not affect the onboard controller. The Qlogic SCSI controller implements the 16-bit wide SCSI bus. The Qlogic module has two terminators, one for the 8 low bits and one for the high 8 bits. There are five possible values: **off**—Turns off both low 8 bits and high 8 bits. **Low**—Turns on low 8 bits and turns off high 8 bits. **High**—Turns on high 8 bits and turns off low 8 bits. **On**—Turns on both low 8 bits and high 8 bits. **Diff**—Places the bus in differential mode.

Table 6–1 SRM Environment Variables Used on ES40 Systems (Continued)

Variable	Attribute	Description
sys_serial_num	NV	Sets the system serial number, which is then propagated to all FRUs that have EEPROMs. The serial number can be read by the operating system.
tt_allow_login	NV	Enables or disables login to the SRM console firmware on alternative console ports. **0**—Disables login on alternative console ports. **1**—Enables login on alternative console ports (default setting). If the console output device is set to **serial**, **set tt_allow_login 1** allows you to log in on the primary COM1(MMJ) port, or alternate COM2 port, or the VGA monitor. If the console output device is set to **graphics**, **set tt_allow_login 1** allows you to log in through either the COM1(MMJ) or COM2 console port.

6.4 Setting Up a System for Windows NT

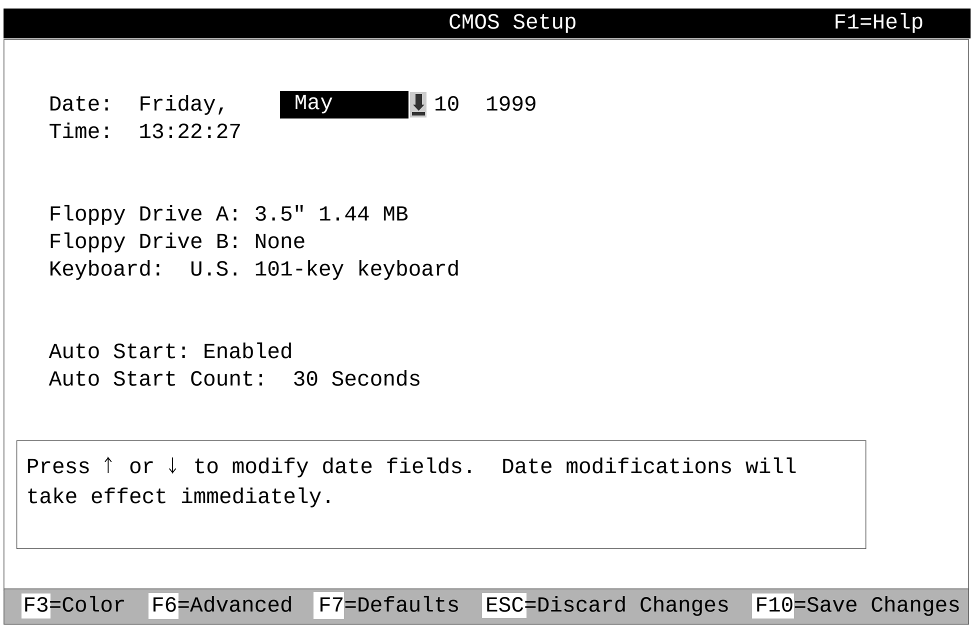
Before you install and boot Windows NT for the first time, set the system date and time and set up the hard disks. Optionally, you can set the level of memory testing and set system password protection.

If you are installing Windows NT from CD-ROM, use the AlphaBIOS CMOS Setup screen and the Hard Disk Setup screen to set up your system. Use the Advanced CMOS Setup screen to set the level of memory testing and to set password protection, if desired.

6.4.1 Setting the Date and Time

Set the date and time from the CMOS Setup screen.

Figure 6–4 CMOS Setup Screen



PK0901

1. Start AlphaBIOS.
2. From the AlphaBIOS Boot screen, press **F2** to enter AlphaBIOS Setup.
3. From AlphaBIOS Setup select **CMOS Setup**, and press Enter.
4. From CMOS Setup set the system date and time. Accept the defaults for all other items.

6.4.2 Setting Up the Hard Disk

Set up the hard disk from the Hard Disk Setup screen.

Figure 6–5 Hard Disk Setup Screen

```
                          Hard Disk Setup


Disk  0   NCRC8xx #0, SCSI ID 0            4091 MB
          Partition 1                      4085 MB      FAT
          Partition 2                         6 MB      FAT

Disk  1   NCRC8XX #0, SCSI ID 1            4091 MB
          Partition 1                      4091 MB      NTFS

Disk  2   NCRC8XX #0, SCSI ID 2            4091 MB
          Partition 1                      4091 MB      NTFS


 INSERT =New  DEL=Delete  F6 =Format  F7 =Express   ESC=Exit
```

PK0940a

Set the date and time as described in Section 6.4.1 before setting up the hard disk.

1. From CMOS Setup press **F10** to return to the AlphaBIOS Setup screen.
2. Select **Hard Disk Setup** and press Enter.
3. Use the arrow keys to select the drive that you want to prepare for Windows NT installation.
4. Press **F7** to perform an express setup on the hard disk that is highlighted.
5. Press **F10** to commit and verify the hard disk setup operation.

CAUTION: *Pressing F10 destroys the contents of the disk drive. Be sure you have selected the drive that you want to prepare before pressing F10.*

For detailed information on hard disk setup, see the *Compaq AlphaServer ES40 User Interface Guide.*

6.4.3 Setting the Level of Memory Testing

Set the level of memory testing that occurs when the system is power cycled from the advanced CMOS Setup screen.

Figure 6–6 Advanced CMOS Setup Screen

```
                    Advanced CMOS Setup              F1=Help

  PCI Parity Checking:           Disabled
  Power-up Memory Test:          Partial
  AlphaBIOS Password Option:     Disabled
  SCSI BIOS Emulation:           Enabled For All

  Console Selection:             Windows NT Console (AlphaBIOS)

 Press ↑ or ↓ to enable or disable power-up memory testing.
 When enabling memory test, PARTIAL will test the first 256 MB,
 FULL will test all of the memory.

 ESC=Discard Changes  F10=Save Changes
```

PK0903a

1. From Advanced CMOS Setup, select **Power-up Memory Test**.
2. Select the level of memory testing you want to occur when the system is power cycled. The three memory test settings are:

Disabled	No memory test performed by AlphaBIOS
Partial	Tests first 256 MB of memory
Full	Tests all of the memory

6.5 Setting Automatic Booting

Windows NT systems are factory set to auto start; UNIX and OpenVMS systems are factory set to halt in the SRM console. You can change these defaults, if desired.

Systems can boot automatically (if set to autoboot) from the default boot device under the following conditions:

- When you first turn on system power
- When you power cycle or reset the system
- When system power comes on after a power failure
- After a bugcheck (OpenVMS and Windows NT) or panic (UNIX)

6.5.1 Windows NT and Auto Start

On Windows NT systems the Auto Start option is enabled by default, which causes the primary operating system to start automatically whenever the machine is power cycled or reset.

If more than one version of Windows NT is installed (for example, Version 4.0 and Version 5.0), the version selected as the primary operating system starts automatically if Auto Start is enabled.

If you want a different version of the operating system to become the primary, you can reorder the boot selections. On the Operating System Selection Setup screen, the current default is the first selection in the list. Use the arrow keys to highlight the boot selection you want to make the primary and press F8. Your selection will move to the top of the list and become the default. The new selection will start automatically if Auto Start is enabled.

If you do not want the Windows NT system to boot an operating system automatically, change the Auto Start setting on the CMOS Setup screen to Disabled.

6.5.2 Setting Tru64 UNIX or OpenVMS Systems to Auto Start

The SRM auto_action environment variable determines the default action the system takes when the system is power cycled, reset, or experiences a failure.

On systems that are factory configured for UNIX or OpenVMS, the factory setting for **auto_action** is **halt**. The **halt** setting causes the system to stop in the SRM console. You must then boot the operating system manually.

For maximum system availability, **auto_action** can be set to **boot** or **restart**.

- With the **boot** setting, the operating system boots automatically after the SRM **init** command is issued or the Reset button is pressed.
- With the **restart** setting, the operating system boots automatically after the SRM **init** command is issued or the Reset button is pressed, and it also reboots after an operating system crash.

To set the default action to **boot**, enter the following SRM commands:

```
P00>>> set auto_action boot
P00>>> init
```

For more information on **auto_action**, see the *Compaq AlphaServer ES40 User Interface Guide.*

6.6 Changing the Default Boot Device

It is not necessary to modify the boot file setting for Windows NT. You can change the default boot device for UNIX or OpenVMS with the set bootdef_dev command.

Windows NT

AlphaBIOS boots Windows NT from the operating system loader program, OSLOADER.EXE. A boot file setting is created along with the operating system selection during Windows NT setup, and this setting is usually not modified by the user. You can, however, modify this setting, if necessary. See the *Compaq AlphaServer ES40 User Interface Guide* for instructions.

UNIX or OpenVMS

With the UNIX or OpenVMS operating systems, you can designate a default boot device. You change the default boot device by using the **set bootdef_dev** SRM console command. For example, to set the boot device to the IDE CD-ROM, enter commands similar to the following:

```
P00>>> show bootdef_dev
bootdef_dev   dka400.4.0.1.1
P00>>> set bootdef_dev dqa500.5.0.1.1
P00>>> show bootdef_dev
bootdef_dev dqa500.5.0.1.1
```

See the *Compaq AlphaServer ES40 User Interface Guide* for more information.

6.7 Running AlphaBIOS-Based Utilities

Depending upon the type of hardware you have, you may have to run hardware configuration utilities. Hardware configuration diskettes are shipped with your system or with options that you order.

Typical configuration utilities include:

- RAID standalone configuration utility for setting up RAID devices
- KZPSA configuration utility for configuring SCSI adapters

These utilities are run from the AlphaBIOS console

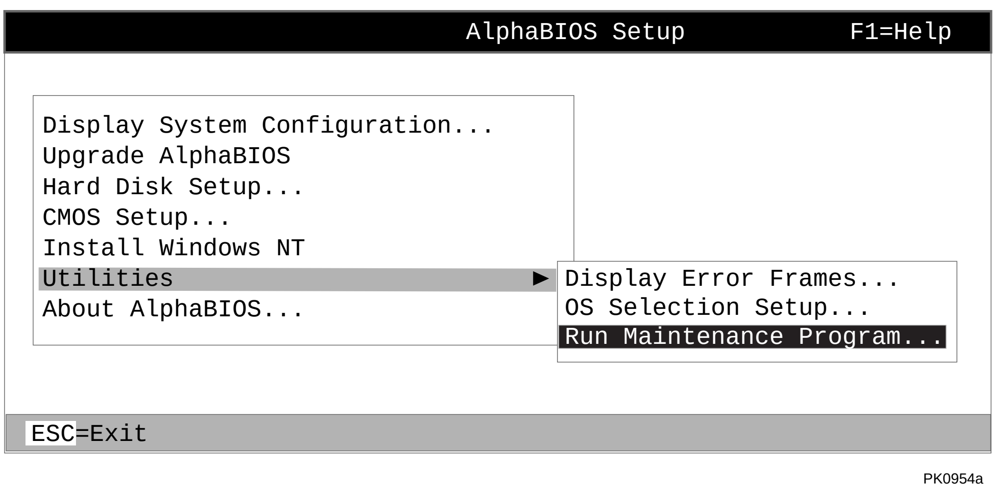
Utilities can be run either in graphics or serial mode. The SRM **console** environment variable controls which mode AlphaBIOS runs in at the time it is loaded by the SRM console.

If you are running Windows NT, your monitor is already in graphics mode. If you are running UNIX or OpenVMS and you have a VGA monitor attached, set the **console** environment variable to **graphics** and enter the **init** command to reset the system before invoking AlphaBIOS.

6.7.1 Running Utilities from a VGA Monitor

If you are running Windows NT, no terminal setup is required for running utilities.

Figure 6–7 AlphaBIOS Utilities Menu

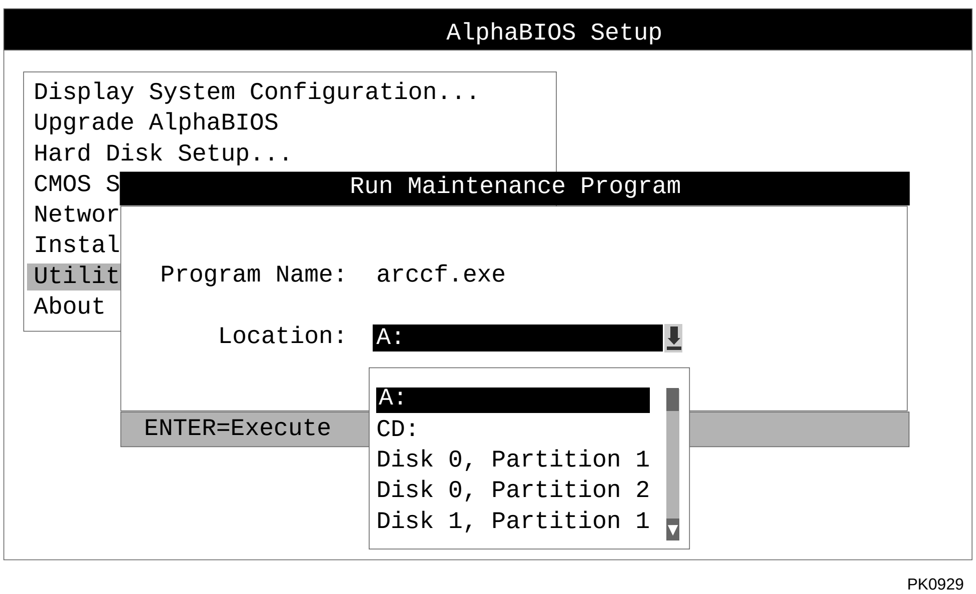


Running a Utility from a VGA Monitor

1. Start the AlphaBIOS console.
2. Press **F2** from the Windows NT Boot screen to display the AlphaBIOS Setup screen.
3. From AlphaBIOS Setup, select **Utilities**, then select **Run Maintenance Program** from the sub-menu that is displayed, and press Enter.

4. In the Run Maintenance Program dialog box, type the name of the program to be run in the Program Name field. Then Tab to the Location list box, and select the hard disk partition, floppy disk, or CD-ROM drive from which to run the program.
5. Press Enter to execute the program.

Figure 6–8 Run Maintenance Program Dialog Box



6.7.2 Setting Up Serial Mode

Serial mode requires a VT320 or higher (or equivalent) terminal. To run AlphaBIOS and maintenance programs in serial mode, set the console environment variable to serial and enter the init command to reset the system.

Set up the serial terminal as follows:

1. From the General menu, set the terminal mode to VT*xxx* mode, 8-bit controls.
2. From the Comm menu, set the character format to 8 bit, no parity, and set receive XOFF to 128 or greater.

6.7.3 Running Utilities from a Serial Terminal

Utilities are run from a serial terminal the same way as from a VGA monitor. The menus are the same, but some key mappings are different.

Table 6–2 AlphaBIOS Option Key Mapping

AlphaBIOS Key	VTxxx Key
F1	Ctrl/A
F2	Ctrl/B
F3	Ctrl/C
F4	Ctrl/D
F5	Ctrl/E
F6	Ctrl/F
F7	Ctrl/P
F8	Ctrl/R
F9	Ctrl/T
F10	Ctrl/U
Insert	Ctrl/V
Delete	Ctrl/W
Backspace	Ctrl/H
Escape	Ctrl/[

1. Issue the **alphabios** command at the P00>>> prompt to start the AlphaBIOS console.
2. From the AlphaBIOS Boot screen, press **F2**.
3. From AlphaBIOS Setup, select **Utilities**, and select **Run Maintenance Program** from the sub-menu that is displayed. Press Enter.
4. In the Run Maintenance Program dialog box, type the name of the program to be run in the Program Name field. Then tab to the Location list box, and select the hard disk partition, floppy disk, or CD-ROM drive from which to run the program.
5. Press Enter to execute the program.

6.7.4 Running the RAID Standalone Configuration Utility

The RAID Standalone Configuration Utility is used to set up RAID disk drives and logical units. The Standalone Utility is run from the AlphaBIOS Utilities menu.

The system supports KZPAC-*xx* Ultra SCSI RAID controllers. The KZPAC-*xx* kit includes the controller, RAID Array 230/Plus Subsystem software, and documentation.

1. Start AlphaBIOS Setup. If the system is in the SRM console, issue the **alphabios** command. (If the system has a VGA monitor, you can set the SRM **console** environment variable to **graphics**.)
2. At the Utilities screen, select Run Maintenance Program. Press Enter.
3. In the Run Maintenance Program dialog box, type **arccf** in the Program Name: field.
4. Press Enter to execute the program. The Main menu displays the following options:

```
[01.View/Update Configuration]
 02.Automatic Configuration
 03.New Configuration
 04.Initialize Logical Drive
 05.Parity Check
 06.Rebuild
 07.Tools
 08.Select Controller
 09.Controller Setup
 10.Diagnostics
```

Refer to the RAID Array Subsystems 230/Plus documentation for information on using the Standalone Configuration Utility to set up RAID drives.

6.8 Setting SRM Security

The set password and set secure commands set SRM security. The login command turns off security for the current session. The clear password command returns the system to user mode.

The SRM console has two modes, user mode and secure mode.

- User mode allows you to use all SRM console commands. User mode is the default mode.
- Secure mode allows you to use only the **boot** and **continue** commands. The **boot** command cannot take command-line parameters when the console is in secure mode. The console boots the operating system using the environment variables stored in NVRAM (**boot_file**, **bootdef_dev**, **boot_flags**).

Example 6–2 set password

```
P00>>> set password                                     ❶
Please enter the password:
Please enter the password again:
P00>>>

P00>>> set password                                     ❷
Please enter the password:
Please enter the password again:
Now enter the old password:
P00>>>

P00>>> set password
Please enter the password:
Password length must be between 15 and 30 characters    ❸
P00>>>
```

Continued on next page

❶ Setting a password. If a password has not been set and the **set password** command is issued, the console prompts for a password and verification. The password and verification are not echoed.

❷ Changing a password. If a password has been set and the **set password** command is issued, the console prompts for the new password and verification, then prompts for the old password. The password is not changed if the validation password entered does not match the existing password stored in NVRAM.

❸ The password length must be between 15 and 30 alphanumeric characters. Any characters entered after the 30th character are not stored.

Example 6–3 set secure

```
P00>>> set secure                                              ❶
Console is secure. Please login.
P00>>> login                                                   ❷
Please enter the password:
P00>>> b dkb0
```

❶ The **set secure** command console puts the console into secure mode. A password must be set before you can issue **set secure**. Once the console is secure, only the **boot** and **continue** commands can be used. The **boot** command cannot take command-line parameters.

❷ Entering the **login** command turns off security features for the current console session. This allows the operator to enter any SRM command—in this case, a **boot** command with command-line parameters.

Example 6–4 clear password

```
P00>>> clear password
Please enter the password:
Password successfully cleared.
P00>>>
```

Clearing the password returns the system to user mode.

If You Forget the Password

If you forget the current password, use the **login** command in conjunction with the control panel Halt button to clear the password, as follows:

1. Enter the **login** command:

   ```
   P00>>> login
   ```

2. When prompted for the password, press the Halt button to the latched position and then press the Return (or Enter) key.
3. Press the Halt button to release the halt. The password is now cleared and the console cannot be put into secure mode unless you set a new password.

6.9 Setting Windows NT Security

Password protection provides two levels of security for a Windows NT system: setup protection and startup protection. When system setup protection is enabled, a password is required to start AlphaBIOS Setup. When startup password protection is enabled, a password is required before the system initializes.

Example 6–5 Advanced CMOS Setup Screen

```
                         Advanced CMOS Setup             F1=Help

      PCI Parity Checking:        Disabled
      Power-up Memory Test:
      AlphaBIOS Password Option:  Enabled
      SCSI BIOS Emulation:        Enabled For All

      Console Selection:          Windows NT Console (AlphaBIOS)

 Press ↑ or ↓ to choose your security preference, then press ENTER to set
 (or change) the password. A setup password protects AlphaBIOS Setup. A
 Start-up password protects all system access.

     ESC=Discard Changes  F10=Save Changes
```

PK0903b

Startup password protection provides more comprehensive protection than setup password protection because with startup protection the system cannot be used at all until the correct password is entered.

To enable password protection:

1. Start AlphaBIOS Setup, select **CMOS Setup**, and press Enter.
2. In the CMOS Setup screen, press **F6** to enter Advanced CMOS Setup.
3. In the Advanced CMOS Setup screen (Example 6–5), select **AlphaBIOS Password Option** and use the arrow keys to select the type of protection you want. An explanatory dialog box appears. Read the dialog box and press Enter to continue.
4. Enter your password in the Enter New Password dialog box, then press Enter.
5. Enter your password in the Confirm New Password dialog box, then press Enter.
6. Press **F10** to save your changes.

NOTE: *To change your password, set up your password again.*

6.10 Configuring Devices

Become familiar with the configuration requirements for CPUs and memory before removing or replacing those components. See Chapter 8 for removal and replacement procedures.

6.10.1 CPU Configuration

Figure 6–9 CPU Slot Locations (Pedestal/Rack)

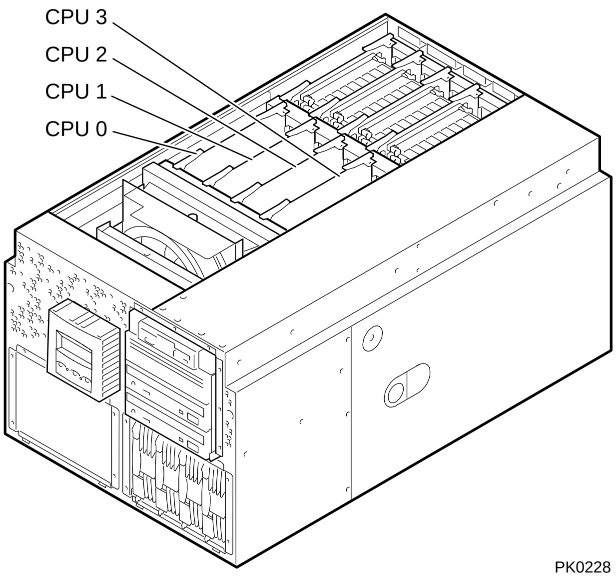
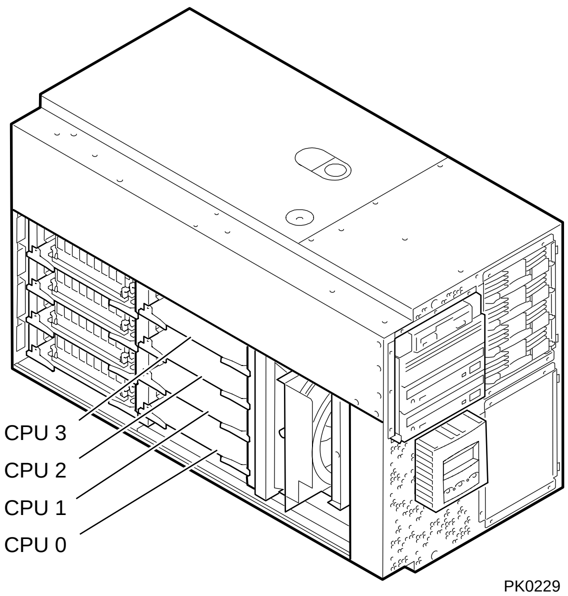


Figure 6–10 CPU Slot Locations (Tower)



CPU Configuration Rules

6. A CPU must be installed in slot 0. The system will not power up without a CPU in slot 0.

7. CPU cards must be installed in numerical order, starting at CPU slot 0. The slots are populated from left to right on a pedestal or rackmount system and from bottom to top on a tower. See Figure 6–9 and Figure 6–10.

8. CPUs must be identical in speed and cache size.

6.10.2 Memory Configuration

Become familiar with the rules for memory configuration before adding DIMMs to the system. For the Model 2 system, do not mix stacked and unstacked DIMMs within an array.

Refer to Figure 6–12 or Figure 6–13 and observe the following rules for installing DIMMs.

- You can install up to 16 DIMMs or up to 32 DIMMs, depending on the system model.
- A set consists of 4 DIMMs. You must install all 4 DIMMs.
- Fill sets in numerical order. Populate all 4 slots in Set 0, then populate Set 1, and so on.
- An "array" is one set for systems that support 16 DIMMs and two sets for systems that support 32 DIMMs.
- DIMMs in an array must be the same capacity and type. For example, suppose you have populated Sets 0, 1, 2, and 3. When you populate Set 4, the DIMMs must be the same capacity and type as those installed in Set 0. Similarly, Set 5 must be populated with DIMMs of the same capacity and type as are in Set 1, and so on, as indicated in the following table.

Array	Model 2 System (Supports 32 DIMMs)	Model 1 System (Supports 16 DIMMs)
0	Set 0 and Set 4	Set 0
1	Set 1 and Set 5	Set 1
2	Set 2 and Set 6	Set 2
3	Set 3 and Set 7	Set 3

DIMM Information for Model 2 Systems

DIMMs are manufactured with two types of SRAMs, stacked and unstacked (see Figure 6–11). Stacked DIMMs provide twice the capacity of unstacked DIMMs, and, at the time of shipment, are the highest capacity DIMMs offered by Compaq. The system may have either stacked or unstacked DIMMs.

You can mix stacked and unstacked DIMMs within the system, but not within an array. The DIMMs within an array must be of the same capacity and type (stacked or unstacked) because of different memory addressing.

When installing sets 0, 1, 2, and 3, an incorrect mix will not occur. When installing sets 4, 5, 6, or 7, however, you must ensure that the four DIMMs being installed match the capacity and type of DIMMs in the existing array. If necessary, rearrange DIMMs for proper configuration.

Figure 6–11 Stacked and Unstacked DIMMs

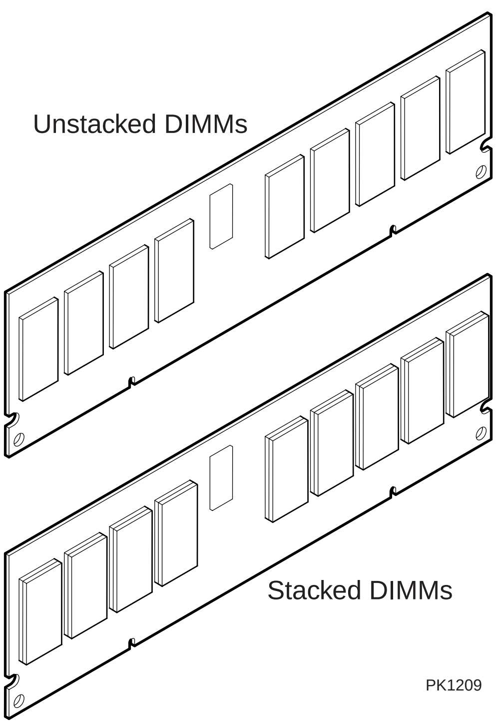


Figure 6–12 Memory Configuration (Pedestal/Rack)

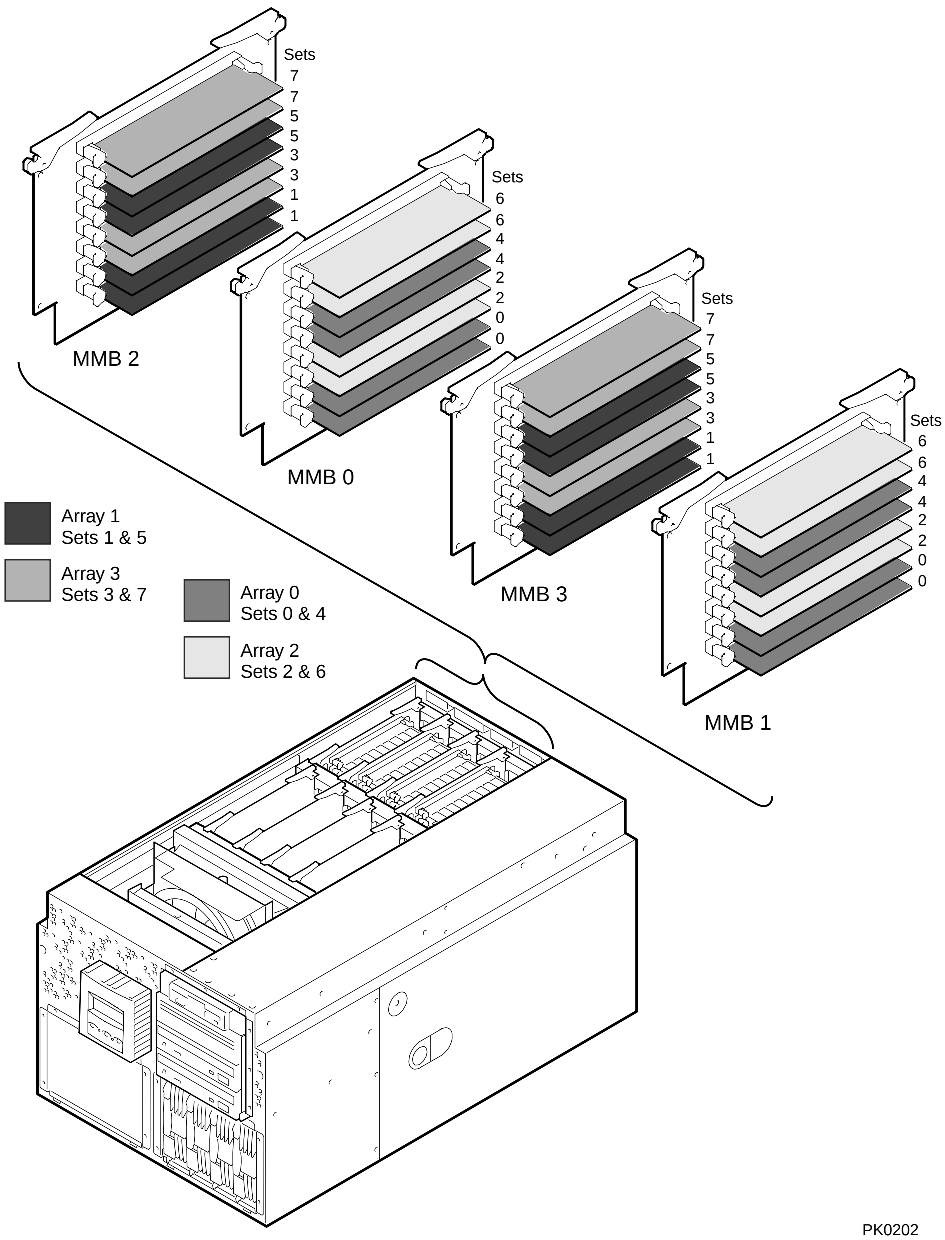
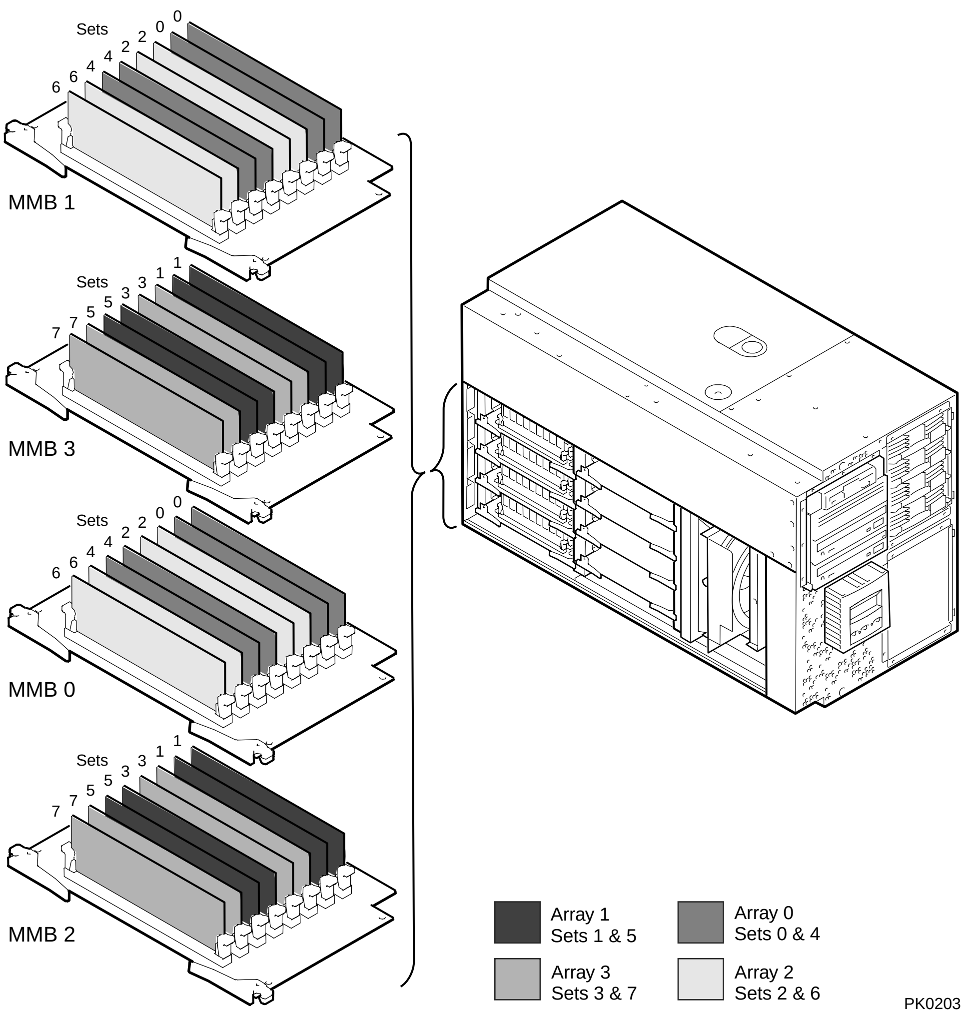


Figure 6–13 Memory Configuration (Tower)



6.10.3 PCI Configuration

Figure 6–14 PCI Slot Locations (Pedestal/Rack)

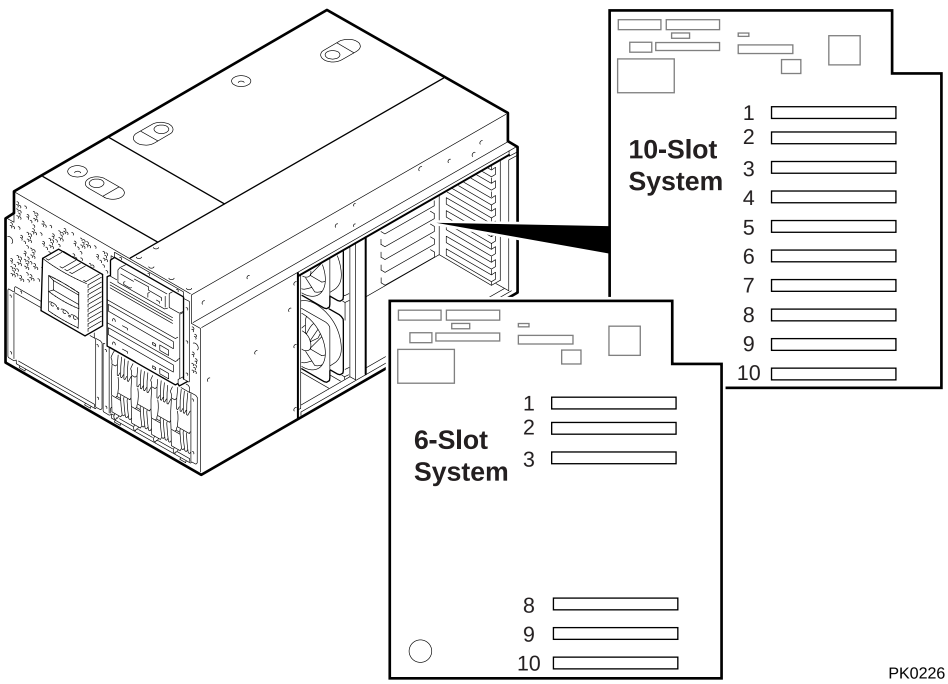
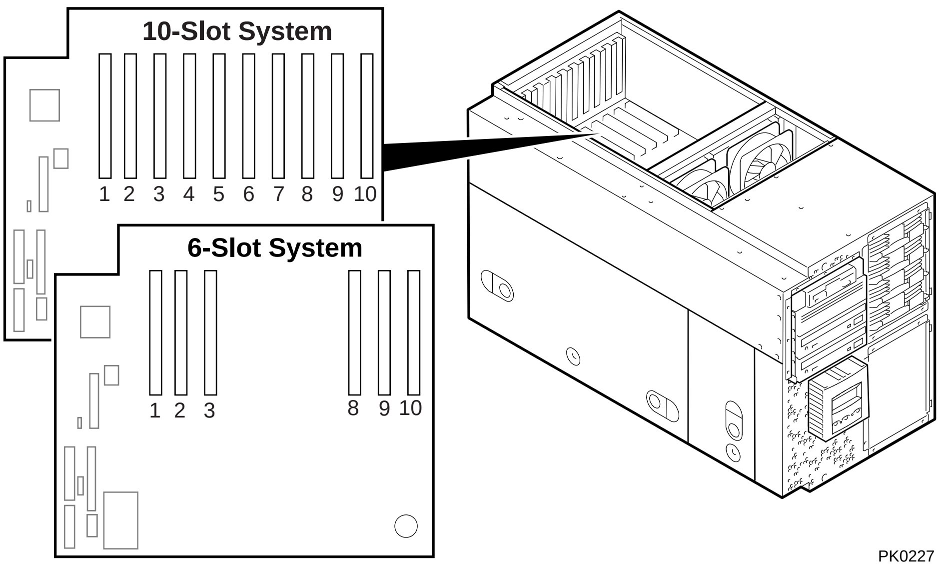


Figure 6–15 PCI Slot Locations (Tower)



The PCI slots are split across two independent 64-bit, 33 MHz PCI buses: PCI0 and PCI1. These buses correspond to Hose 0 and Hose 1 in the system logical configuration. The slots on each bus are listed below.

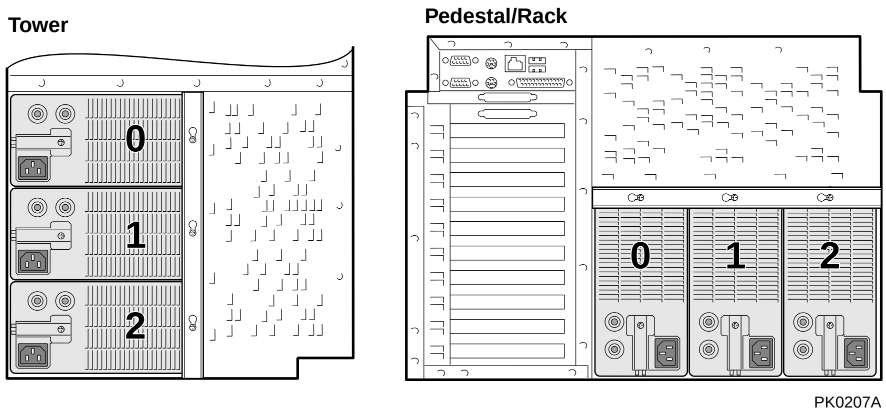
System Variant	Slots on PCI 0	Slots on PCI 1
Six-slot system	1–3	8–10
Ten-slot system	1–4	5–10

Some PCI options require drivers to be installed and configured. These options come with a floppy or a CD-ROM. Refer to the installation document that came with the option and follow the manufacturer's instructions.

NOTE: *If you have a VGA controller, it must be installed on PCI 0.*

6.10.4 Power Supply Configurations

Figure 6–16 Power Supply Locations



The system can have the following power configurations:

Single Power Supply. A single power supply is provided with entry-level systems, such as a system configured with:

- One or two CPUs
- One storage cage

Two Power Supplies. Two power supplies are required if the system has more than two CPUs or if the system has a second storage cage.

Redundant Power Supply. If one power supply fails, the redundant supply provides power and the system continues to operate normally. A second power supply adds redundancy for an entry-level system such as the system described under "Single Power Supply." A third power supply adds redundancy for a system that requires two power supplies.

Recommended Installation Order. Generally, power supply 0 is installed first, power supply 1 second, and power supply 2 third, but the supplies can be installed in any order. See Figure 6–16. The power supply numbering corresponds to the numbering displayed by the SRM **show power** command.

6.11 Switching Between Operating Systems

The system supports three operating systems. You can install Tru64 UNIX, OpenVMS, or Windows NT. You can also switch from one operating system to another by removing the disk for the operating system that is currently installed and installing the disk for the operating system you want to run.

CAUTION: *The file structures of the three operating systems are incompatible. When you switch between operating systems, you cannot read the data off disks associated with the operating system that was running previously.*

When you switch between operating systems, be sure to pull out the system and data disks for the operating system you will not be using. Otherwise, you risk corrupting data on the system disk.

To run Windows NT on an AlphaServer ES40 system, you must use only options that are supported on Windows NT. See the Supported Options List.

6.11.1 Switching from UNIX or OpenVMS to Windows NT

Follow this procedure if you have already installed UNIX or OpenVMS and want to switch to Windows NT.

CAUTION: *Before switching operating systems, make a note of the boot path and location of the system disk (controller, SCSI ID number, and so on) of the operating system you are removing so that you can restore that operating system at a later date.*

1. Shut down the operating system and power off the system. Unplug the power cord from each power supply.
2. Remove the enclosure panels and system covers as described in Chapter 8.
3. Remove any options that are not supported on Windows NT and replace them with supported options.
4. Remove the UNIX or OpenVMS operating system disk and insert the Windows NT system disk.
5. Plug in the power supplies and power up the system.
6. Enter the following commands at the SRM console prompt:

```
P00>>> set console graphics
P00>>> set os_type nt
P00>>> init
```

7. At the AlphaBIOS boot screen, start AlphaBIOS Setup (**F2**), select **CMOS Setup**, and press Enter. Set the system date and time.
8. In CMOS Setup, check that the setup for the floppy and other basic parameters is accurate. Set system-specific parameters, such as the memory test and password, in Advanced CMOS Setup as needed. Press **F10** to save the changes.
9. From the AlphaBIOS Setup screen select **Utilities**. In the selection box that is displayed, choose **OS Selection Setup**. Make sure the selections (boot name, boot file, and so on) are what you want. Press **F10** to save any changes.

NOTE: *Adding or removing SCSI option cards as noted in step 3 may cause the logical drive numbers to be reordered and the boot selections to be invalid. Upon entering the OS Selection Setup screen, you will see warning dialogs, and AlphaBIOS will attempt to set the boot selections to the new locations.*

---

10. Return to the boot screen and boot Windows NT.

6.11.2 Switching from Windows NT to UNIX or OpenVMS

Follow this procedure if you have already installed Windows NT and want to switch to UNIX or OpenVMS.

CAUTION: *Before switching operating systems, make a note of the boot path and location of the system disk (controller, SCSI ID number, and so on) of the operating system you are removing so that you can restore that operating system at a later date.*

1. Shut down the operating system and power off the system. Unplug the power cord from each power supply.
2. Remove the enclosure panels and system covers as described in Chapter 8.
3. Remove any options that are not supported on Tru64 UNIX or OpenVMS and replace them with supported options.
4. Remove the Windows NT system disk and insert the UNIX or OpenVMS system disk.
5. Plug in the power supplies and power up the system.
6. In AlphaBIOS, access the Advanced CMOS Setup screen and change the Console Selection to UNIX console (SRM) or OpenVMS Console (SRM), as appropriate. Press **F10** to save the change. This menu selection changes the setting of the **os_type** environment variable so that the SRM console is loaded the next time you reset your system.
7. Press the Reset button to reset the system.
8. In the SRM console, restore the boot parameters you saved previously for UNIX or OpenVMS.
9. Boot the UNIX or OpenVMS operating system.
10. Set the system date and time.

Chapter 7
Using the Remote Management Console

You can manage the system through the remote management console (RMC). The RMC is implemented through an independent microprocessor that resides on the system motherboard. The RMC also provides access to the repository for all error information in the system.

This chapter explains the operation and use of the RMC. Sections are:

- RMC Overview
- Operating Modes
- Terminal Setup
- Connecting to the RMC CLI
- SRM Environment Variables for COM1
- RMC Command-Line Interface
- Resetting the RMC to Factory Defaults
- Troubleshooting Tips

7.1 RMC Overview

The remote management console provides a mechanism for monitoring the system (voltages, temperatures, and fans) and manipulating it on a low level (reset, power on/off, halt). It also provides functionality to read and write configuration and error log information to FRU error log devices.

The RMC performs monitoring and control functions to ensure the successful operation of the system.

- Monitors thermal sensors on the CPUs, the PCI backplane, and the power supplies
- Monitors voltages, power supplies, and fans
- Handles hot swap of power supplies and fans
- Controls the operator control panel (OCP) display and writes status messages on the display
- Detects alert conditions such as excessive temperature, fan failure, and power supply failure. On detection, RMC displays messages on the OCP, pages an operator, and sends an interrupt to SRM or AlphaBIOS, which then passes the interrupt to the operating system or an application.
- Shuts down the system if any fatal conditions exist. For example:
  - ➢ The temperature reaches the failure limit.
  - ➢ The cover to the system card cage is removed.
  - ➢ The main fan (Fan 6) and the redundant fan (Fan 5) fail.
- Retrieves and passes information about a system shutdown to SRM or AlphaBIOS at the next power-up. SRM or AlphaBIOS displays a message regarding the last shutdown.
- Provides a command-line interface (CLI) for the user to control the system. From the CLI you can power the system on and off, halt or reset the system, and monitor the system environment.
- Passes error log information to the DPR so that this information can be accessed by the system.
- Retrieves information from the DPR and stores it in FRU EEROMs.

The RMC logic is implemented using an 8-bit microprocessor, PIC17C44, as the primary control device. The firmware code is resident within the microprocessor and in flash memory. If the RMC firmware should ever become corrupted or obsolete, you can update it manually using the Loadable Firmware Update Utility. See Chapter 3 for details. The microprocessor can also communicate with the system power control logic to turn on or turn off power to the rest of the system.

The RMC is powered by an auxiliary 5V supply. You can gain access to the RMC as long as AC power is available to the system (through an AC outlet). Thus, if the system fails, you can still access the RMC and gather error/fault information about the failure.

DPR Error Repository

The RMC manages an extensive network of FRU I^2C EEPROMs. Information from these EEPROMs is stored in dual-port RAM (DPR)—a shared RAM that facilitates interaction between the RMC and the system—and can be accessed to diagnose hardware failures.

At system power-up, the RMC reads 256 bytes of data from each FRU EEPROM and stores it in the DPR. The EEPROM data contains information on configuration and errors. The data is accessible through the TIG chip on the system motherboard.

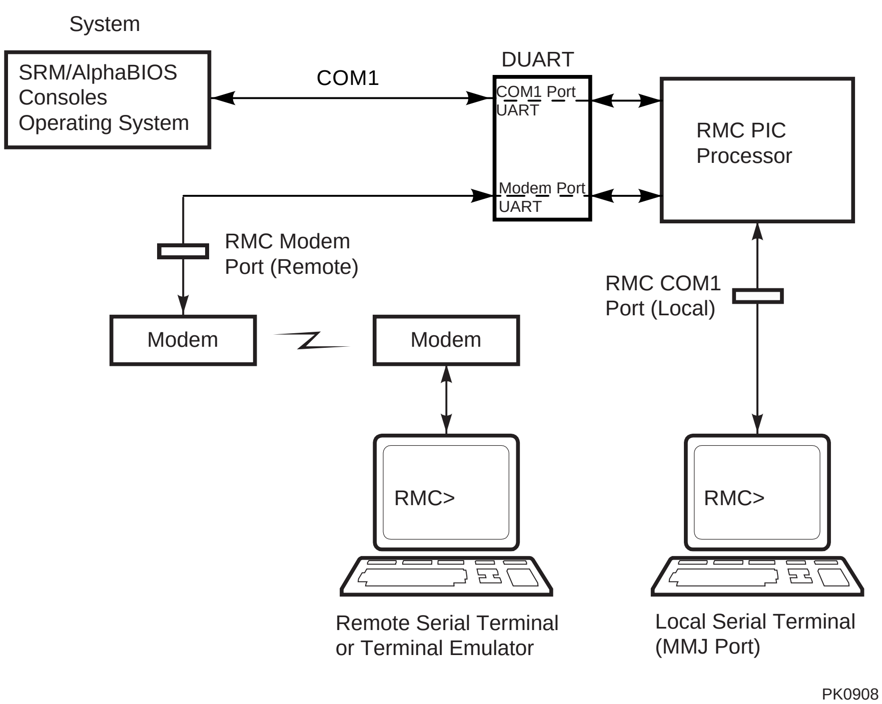
As one of its functions, the TIG provides interfaces for the firmware and the operating system to communicate with the server management logic. The data accessed from DPR provides configuration information to the firmware during start-up. Remote or local applications can read the DPR system error and configuration repository. The error log information is written to the DPR by an error handling agent and then written back to the EEPROMs by the RMC. This arrangement ensures that the error log is available on a FRU after power has been lost.

The RMC console provides several commands for accessing error information in the DPR. See Section 7.6. Compaq Analyze, described in Chapter 5, can access the FRU EEPROM error logs to provide diagnostic information for system FRUs.

7.2 Operating Modes

The RMC can be configured to manage different data flow paths defined by the com1_mode environment variable. In Through mode (the default), all data and control signals flow from the system COM1 port through the RMC to the active external port. You can also set bypass modes so that the signals partially or completely bypass the RMC. The com1_mode environment variable can be set from either SRM or the RMC. See Section 7.6.1.

Figure 7–1 Data Flow in Through Mode



Through Mode

Through mode is the default operating mode. The RMC routes every character of data between the internal system COM1 port and the active external port, either the local COM1 serial port (MMJ) or the 9-pin modem port. If a modem is connected, the data goes to the modem. The RMC filters the data for a specific escape sequence. If it detects the escape sequence, it connects to the RMC CLI.

Figure 7–1 illustrates the data flow in Through mode. The internal system COM1 port is connected to one port of the DUART chip, and the other port is connected to a 9-pin external modem port, providing full modem controls. The DUART is controlled by the RMC microprocessor, which moves characters between the two UART ports. The local MMJ port is always connected to the internal UART of the microprocessor. The escape sequence signals the RMC to connect to the CLI. Data issued from the CLI is transmitted between the RMC microprocessor and the active port that connects to the RMC CLI.

NOTE: *The internal system COM1 port should not be confused with the external COM1 serial port on the back of the system. The internal COM1 port is used by the system software to send data either to the COM1 port on the system or to the RMC modem port if a modem is connected.*

Local Mode

You can set a Local mode in which only the local channel can communicate with the system COM1 port. In Local mode the modem is prevented from sending characters to the system COM1 port, but you can still connect to the RMC CLI from the modem.

7.2.1 Bypass Modes

For modem connection, you can set the operating mode so that data and control signals partially or completely bypass the RMC. The bypass modes are Snoop, Soft Bypass, and Firm Bypass.

Figure 7–2 Data Flow in Bypass Mode

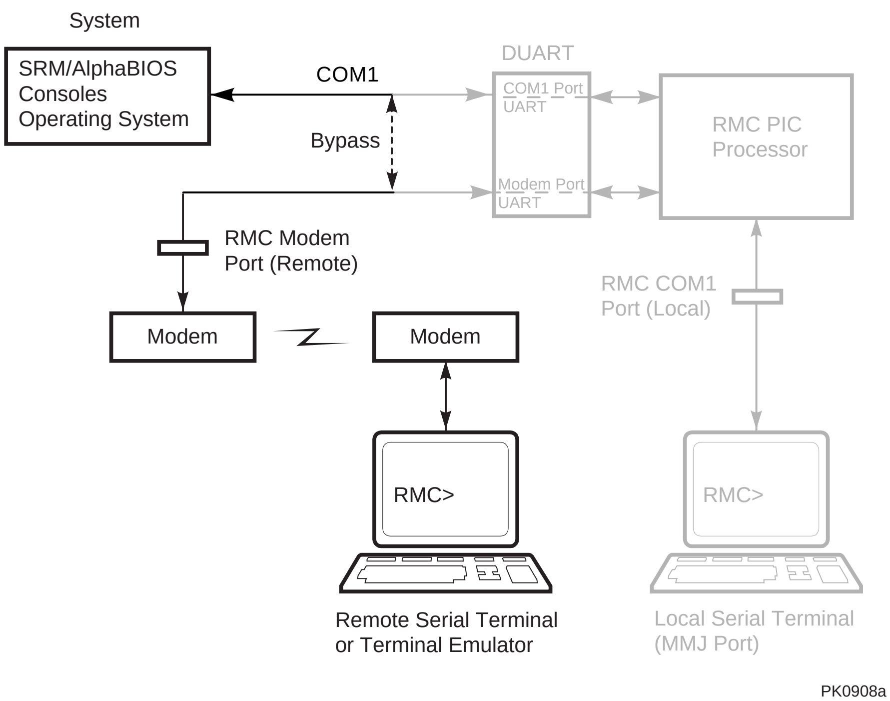


Figure 7–2 shows the data flow in the bypass modes. Note that the internal system COM1 port is connected directly to the modem port.

NOTE: *You can connect a serial terminal to the modem port in any of the bypass modes.*

The local terminal is still connected to the RMC and can still connect to the RMC CLI to switch the COM1 mode if necessary.

Snoop Mode

In Snoop mode data partially bypasses the RMC. The data and control signals are routed directly between the system COM1 port and the external modem port, but the RMC taps into the data lines and listens passively for the RMC escape sequence. If it detects the escape sequence, it connects to the RMC CLI.

The escape sequence is also passed to the system on the bypassed data lines. If you decide to change the default escape sequence, be sure to choose a unique sequence so that the system software does not interpret characters intended for the RMC.

In Snoop mode the RMC is responsible for configuring the modem for dial-in as well as dial-out alerts and for monitoring the modem connectivity.

Because data passes directly between the two UART ports, Snoop mode is useful when you want to monitor the system but also ensure optimum COM1 performance.

Soft Bypass Mode

In Soft Bypass mode all data and control signals are routed directly between the system COM1 port and the external modem port, and the RMC does not listen to the traffic on the COM1 data lines. The RMC is responsible for configuring the modem and monitoring the modem connectivity. If the RMC detects loss of carrier or the system loses power, it switches automatically into Snoop mode. If you have set up the dial-out alert feature, the RMC pages the operator if an alert is detected and the modem line is not in use.

Soft Bypass mode is useful if management applications need the COM1 channel to perform a binary download, because it ensures that RMC does not accidentally interpret some binary data as the escape sequence.

Continued on next page

After downloading binary files, you can set the **com1_mode** environment variable from the SRM console to switch back to Snoop mode or other modes for accessing the RMC, or you can hang up the current modem session and reconnect it.

Firm Bypass Mode

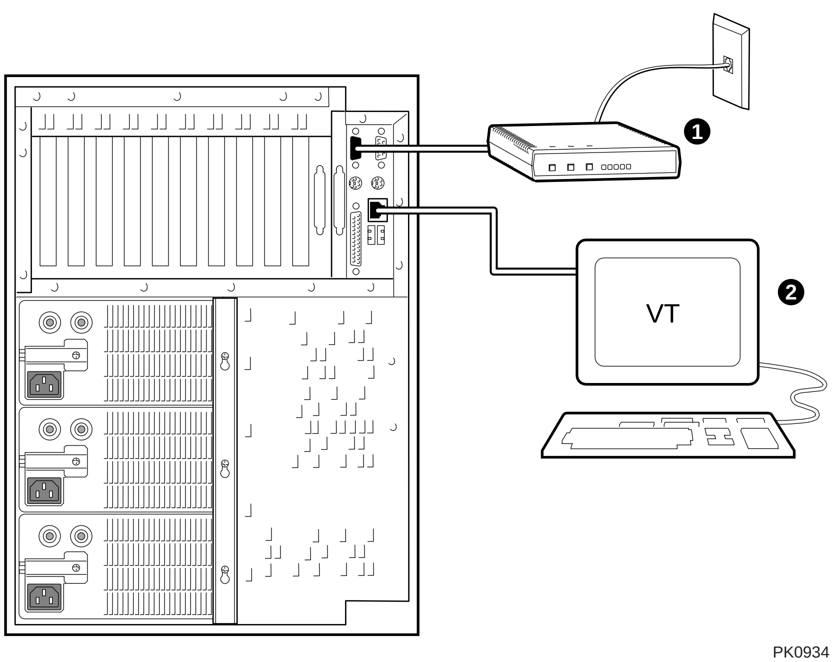
In Firm Bypass mode all data and control signals are routed directly between the system COM1 port and the external modem port. The RMC does not configure or monitor the modem. Firm Bypass mode is useful if you want the system, not the RMC, to fully control the modem port and you want to disable RMC remote management features such as remote dial-in and dial-out alert.

You can switch to other modes by resetting the **com1_mode** environment variable from the SRM console, but you must then set up the RMC again from the local terminal.

7.3 Terminal Setup

You can use the RMC from a modem hookup or the serial terminal connected to the system. As shown in Figure 7–3, a modem is connected to the dedicated 9-pin modem port ❶ and a terminal is connected to the COM1 serial port/terminal port (MMJ) ❷.

Figure 7–3 Terminal Setup for RMC (Tower View)



7.4 Connecting to the RMC CLI

You type an escape sequence to connect to the RMC CLI. You can connect to the CLI from any of the following: a modem, the local serial console terminal, the local VGA monitor, or the system. The "system" includes the operating system, SRM, AlphaBIOS, or an application.

- You can connect to the RMC CLI from the local terminal regardless of the current operating mode.
- You can connect to the RMC CLI from the modem if the RMC is in Through mode, Snoop mode, or Local mode. In Snoop mode the escape sequence is passed to the system and displayed.

NOTE: *Only one RMC CLI session can be active at a time.*

Connecting from a Serial Terminal

Invoke the RMC CLI from a serial terminal by typing the following default escape sequence:

```
^[^[ rmc
```

This sequence is equivalent to typing Ctrl/left bracket, Ctrl/left bracket, rmc. On some keyboards, the Esc key functions like the Ctrl/left bracket combination.

To exit, enter the **quit** command. This action returns you to whatever you were doing before you invoked the RMC CLI. In the following example, the **quit** command returns you to the system COM1 port.

```
RMC> quit
Returning to COM port
```

Connecting from the Local VGA Monitor

To connect to the RMC CLI from the local VGA monitor, the **console** environment variable must be set to **graphics** and the SRM console must be running.

Invoke the SRM console and enter the **rmc** command.

```
P00>>> rmc
You are about to connect to the Remote Management Console.
Use the RMC reset command or press the front panel reset
button to disconnect and to reload the SRM console.
Do you really want to continue? [y/(n)] y
Please enter the escape sequence to connect to the Remote
Management Console.
```

After you enter the escape sequence, the system connects to the CLI and the RMC> prompt is displayed.

When the RMC CLI session is completed, reset the system with the Reset button on the operator control panel or issue the RMC **reset** command.

```
RMC> reset
Returning to COM port
```

7.5 SRM Environment Variables for COM1

Several SRM environment variables allow you to set up the COM1 serial port (MMJ) for use with the RMC.

You may need to set the following environment variables from the SRM console, depending on how you decide to set up the RMC.

com1_baud Sets the baud rate of the COM1 serial port and the modem port. The default is 9600.

com1_flow Specifies the flow control on the serial port. The default is **software**.

com1_mode Specifies the COM1 data flow paths so that data either flows through the RMC or bypasses it. This environment variable can be set from either the SRM or the RMC.

com1_modem Specifies to the operating system whether or not a modem is present.

See the *Compaq AlphaServer ES40 User Interface Guide* for information on setting SRM environment variables.

7.6 RMC Command-Line Interface

The remote management console supports setup commands and commands for managing the system.

The RMC commands are listed below.

clear {alert, port}
dep
disable {alert, remote}
dump
enable {alert, remote}
env
halt {in, out}
hangup
help or **?**
power {on, off}
quit
reset
send alert
set {alert, com1_mode, dial, escape, init, logout, password, user}
status

The commands for setting up and using the RMC are described in the following sections. The **dep** command is reserved. For an RMC commands reference, see the *Compaq AlphaServer ES40 User Interface Guide*.

Continued on next page

Command Conventions

Observe the following conventions for entering RMC commands:

- Enter enough characters to distinguish the command.

NOTE: *The* **reset** *and* **quit** *commands are exceptions. You must enter the entire string for these commands to work.*

- For commands consisting of two words, enter the entire first word and at least one letter of the second word. For example, you can enter **disable a** for **disable alert**.
- For commands that have parameters, you are prompted for the parameter.
- Use the Backspace key to erase input.
- If you enter a nonexistent command or a command that does not follow conventions, the following message is displayed:

```
*** ERROR - unknown command ***
```

- If you enter a string that exceeds 14 characters, the following message is displayed:

```
*** ERROR - overflow ***
```

- Use the Backspace key to erase input.

7.6.1 Defining the COM1 Data Flow

Use the set com1_mode command from SRM or RMC to define the COM1 data flow paths.

You can set **com1_mode** to one of the following values:

through All data passes through RMC and is filtered for the escape sequence. This is the default.

snoop Data partially bypasses RMC, but RMC taps into the data lines and listens passively for the escape sequence.

soft_bypass Data bypasses RMC, but RMC switches automatically into Snoop mode if loss of carrier occurs.

firm_bypass Data bypasses RMC. RMC remote management features are disabled.

local Changes the focus of the COM1 traffic to the local MMJ port if RMC is currently in one of the bypass modes or is in Through mode with an active remote session.

Example 7–1 set com1_mode

```
RMC> set com1_mode
Com1_mode (THROUGH, SNOOP, SOFT_BYPASS, FIRM_BYPASS, LOCAL): local
```

NOTE: *For more details, see the Compaq AlphaServer ES40 User Interface Guide.*

7.6.2 Displaying the System Status

The RMC status command displays the current RMC settings. Table 7–1 explains the status fields.

Example 7–2 status

```
RMC> status
   PLATFORM STATUS
On-Chip Firmware Revision: V1.0
Flash Firmware Revision: V1.2
Server Power: ON
System Halt: Deasserted
RMC Power Control: ON
Escape Sequence: ^[^[RMC
Remote Access: Enabled
RMC Password: set
Alert Enable: Disabled
Alert Pending: YES
Init String: AT&F0E0V0X0S0=2
Dial String: ATXDT9,15085553333
Alert String: ,,,,,,5085553332#;
Com1_mode: THROUGH
Last Alert: CPU door opened
Logout Timer: 20 minutes
User String:
```

Table 7–1 Status Command Fields

Field	Meaning
On-Chip Firmware Revision:	Revision of RMC firmware on the microcontroller.
Flash Firmware Revision:	Revision of RMC firmware in flash ROM.
Server Power:	ON = System is on. OFF = System is off.
System Halt:	Asserted = System has been halted. Deasserted = Halt has been released.
RMC Power Control:	ON= System has powered on from RMC. OFF = System has powered off from RMC.
Escape Sequence:	Current escape sequence for access to RMC console.
Remote Access:	Enabled = Modem for remote access is enabled. Disabled = Modem for remote access is disabled.
RMC Password:	Set = Password set for modem access. Not set = No password set for modem access.
Alert Enable:	Enabled = Dial-out enabled for sending alerts. Disabled = Dial-out disabled for sending alerts.
Alert Pending:	YES = Alert has been triggered. NO = No alert has been triggered.
Init String:	Initialization string that was set for modem.
Dial String:	Pager string to be dialed when an alert occurs.
Alert String:	Identifies the system that triggered the alert to the paging service. Usually the phone number of the monitored system.
Com1_mode:	Identifies the current COM1 mode.
Last Alert:	Type of alert (for example, power supply 1 failed).
Logout Timer:	The amount of time before the RMC terminates an inactive modem connection. The default is 20 minutes.
User String:	Notes supplied by user.

7.6.3 Displaying the System Environment

The RMC env command provides a snapshot of the system environment.

Example 7–3 env

```
RMC> env

       System Hardware Monitor

Temperature (warnings at 45.0°C, power-off at 50.0°C)                ❶
    CPU0: 26.0°C    CPU1: 26.0°C    CPU2: 27.0°C    CPU3: 26.0°C     ❷
    Zone0: 29.0°C    Zone1: 30.0°C    Zone2: 31.0°C
Fan RPM
    Fan1: 2295    Fan2: 2295    Fan3: 2205                           ❸
    Fan4: 2235    Fan5: OFF     Fan6: 2518
Power Supply(OK, FAIL, OFF, '----' means not present)                ❹
    PS0 : OK       PS1 : OK       PS2 : ----
    CPU0: OK       CPU1: OK       CPU2: OK       CPU3: OK
CPU CORE voltage
    CPU0: +2.192V    CPU1: +2.192V    CPU2: +2.192V    CPU3: +2.192V ❺
CPU IO voltage
    CPU0: +1.488V    CPU1: +1.488V    CPU2: +1.488V    CPU3: +1.488V
Bulk voltage
    +3.3V Bulk: +3.328V    +5V Bulk: +5.076V   +12V Bulk: +12.096V   ❻
         Vterm: +1.824V       Cterm: +2.000V   -12V Bulk: -12.480V
```

❶ CPU temperature. In this example four CPUs are present.

❷ Temperature of PCI backplane: Zone 0 includes PCI slots 1–3, Zone 1 includes PCI slots 7–10, and Zone 2 includes PCI slots 4–6.

❸ Fan RPM. With the exception of Fan 5, all fans are powered as long as the system is powered on. Fan 5 is OFF unless Fan 6 fails.

❹ The normal power supply status is either OK (system is powered on) or OFF (system is powered off or the power supply cord is not plugged in). FAIL indicates a problem with a supply.

❺ CPU CORE voltage and CPU I/O voltage. In a healthy system, the core voltage for all CPUs should be the same, and the I/O voltage for all CPUs should be the same.

❻ Bulk power supply voltage.

7.6.4 Dumping DPR Data

The dump command dumps unformatted data from DPR locations 0–3FFF hex. The information might be useful for system trouble-shooting. Use the DPR address table in Appendix C to analyze the data.

Example 7–4 dump

```
RMC> dump
Address: 10         ❶
Count: ee           ❷
  ❸
0010:03 31 07 28 01 09 00 00 00 00 00 00 00 00 00 00
0020:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040:01 80 01 01 01 01 01 01 00 00 00 00 00 00 00 00
0050:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090:00 00 00 00 00 00 00 00 00 00 1D 00 19 18 19 00
00A0:00 00 00 00 00 00 00 00 00 00 00 FF FF FA FA 3B
00B0:00 00 00 00 00 00 00 00 00 00 BA 00 00 00 00 00
00C0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00D0:00 00 00 00 00 00 00 00 00 00 22 00 00 00 00 00
00E0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F0:00 00 00 00 00 00 00 00 00 10 00 00 00 0A 03 0A
RMC>
```

❶ DPR address

❷ Number of bytes dumped (in hex). In the example the **dump** command dumps EF bytes from address 10.

❸ Bytes 10:15 are the time stamp. See Appendix C for the meaning of other locations.

The **dump** command allows you to dump data from the DPR. You can use this command locally or remotely if you are not able to access the SRM console because of a system crash.

The **dump** command accepts two arguments:

Address: Prompts for the starting address

Count: Prompts for the number of following consecutive bytes. If no count is specified, the count defaults to 0.

7.6.5 Power On and Off, Reset, and Halt

The RMC power {on, off}, halt {in, out}, and reset commands perform the same functions as the buttons on the operator control panel.

Power On and Power Off

The RMC **power on** command powers the system on, and the **power off** command powers the system off. The Power button on the OCP, however, has precedence.

- If the system has been powered off with the Power button, the RMC cannot power the system on. If you enter the **power on** command, the message "Power button is OFF" is displayed, indicating that the command will have no effect.
- If the system has been powered on with the Power button, and the **power off** command is used to turn the system off, you can toggle the Power button to power the system back on.

When you issue the **power on** command, the terminal exits RMC and reconnects to the server's COM1 port.

Example 7–5 power on/off

```
RMC> power on
Returning to COM port
RMC> power off
```

Halt In and Halt Out

The **halt in** command halts the system. The **halt out** command releases the halt. When you issue either the **halt in** or **halt out** command, the terminal exits RMC and reconnects to the server's COM1 port.

Example 7–6 halt in/out

```
RMC> halt in
Returning to COM port
RMC> halt out
Returning to COM port
```

The **halt out** command cannot release the halt if the Halt button is latched in. If you enter the **halt out** command, the message "Halt button is IN" is displayed, indicating that the command will have no effect. Toggling the Power button on the operator control panel overrides the **halt in** condition.

Reset

The RMC **reset** command restarts the system. The terminal exits RMC and reconnects to the server's COM1 port.

Example 7–7 reset

```
RMC> reset
Returning to COM port
```

7.6.6 Configuring Remote Dial-In

Before you can dial in through the RMC modem port or enable the system to call out in response to system alerts, you must configure RMC for remote dial-in.

Connect your modem to the 9-pin modem port and turn it on. Connect to the RMC CLI from either the local serial terminal or the local VGA monitor to set up the parameters.

Example 7–8 Dial-In Configuration

```
RMC> set password                                            ❶
RMC Password: ****
Verification: ****
RMC> set init                                                ❷
Init String: AT&F0E0V0X0S0=2
RMC> enable remote                                           ❸
RMC> status                                                  ❹
.
.
Remote Access: Enabled
.
.
.
```

NOTE: *The following modems require the initialization strings shown here. For other modems, see your modem documentation.*

Modem	Initialization String
Motorola 3400 Lifestyle 28.8	AT&F0E0V0X0S0=2
AT &T Dataport 14.4/FAX	AT&F0E0V0X0S0=2
Hayes Smartmodem Optima 288 V-34/V.FC + FAX	AT&FE0V0X0S0=2

❶ Sets the password that is prompted for at the beginning of a modem session. The string cannot exceed 14 characters and is not case sensitive. For security, the password is not echoed on the screen. When prompted for verification, type the password again.

❷ Sets the initialization string. The string is limited to 31 characters and can be modified depending on the type of modem used. Because the modem commands disallow mixed cases, the RMC automatically converts all alphabetic characters entered in the init string to uppercase.

The RMC automatically configures the modem's flow control according to the setting of the SRM **com1_flow** environment variable. The RMC also enables the modem carrier detect feature to monitor the modem connectivity.

❸ Enables remote access to the RMC modem port by configuring the modem with the setting stored in the initialization string.

❹ Verifies the settings. Check that the Remote Access field is set to Enabled.

Dialing In

The following example shows the screen output when a modem connection is established.

```
ATDT915085553333

RINGING

RINGING

CONNECT 9600/ARQ/V32/LAPM

RMC Password: *********

Welcome to RMC V1.2
P00>>> ^[^[rmc
RMC>
```

1. At the RMC> prompt, enter commands to monitor and control the remote system.
2. When you have finished a modem session, enter the **hangup** command to cleanly terminate the session and disconnect from the server.

7.6.7 Configuring Dial-Out Alert

When you are not monitoring the system from a modem connection, you can use the RMC dial-out alert feature to remain informed of system status. If dial-out alert is enabled, and the RMC detects alarm conditions within the managed system, it can call a preset pager number.

You must configure remote dial-in for the dial-out feature to be enabled. See Section 7.6.6.

To set up the dial-out alert feature, connect to the RMC CLI from the local serial terminal or local VGA monitor.

Example 7–9 Dial-Out Alert Configuration

```
RMC> set dial                                          ❶
Dial String: ATXDT9,15085553333
RMC> set alert                                         ❷
Alert String: ,,,,,,5085553332#;
RMC> enable alert                                      ❸
RMC> clear alert                                       ❹
RMC> send alert                                        ❺
Alert detected!
RMC> clear alert                                       ❻
RMC> status                                            ❼
.
.
Alert Enable: Enabled
.
.
```

A typical alert situation might be as follows:

- The RMC detects an alarm condition, such as over temperature warning.
- The RMC dials your pager and sends a message identifying the system.
- You dial the system from a remote serial terminal.
- You connect to the RMC CLI, check system status with the **env** command, and, if the situation requires, power down the managed system.
- When the problem is resolved, you power up and reboot the system.

The elements of the dial string and alert string are shown in Table 7–2. Paging services vary, so you need to become familiar with the options provided by the paging service you will be using. The RMC supports only numeric messages.

❶ Sets the string to be used by the RMC to dial out when an alert condition occurs. The dial string must include the appropriate modem commands to dial the number.

❷ Sets the alert string, typically the phone number of the modem connected to the remote system. The alert string is appended after the dial string, and the combined string is sent to the modem when an alert condition is detected.

❸ Enables the RMC to page a remote system operator.

❹ Clears any alert that may be pending. This ensures that the **send alert** command will generate an alert condition.

❺ Forces an alert condition. This command is used to test the setup of the dial-out alert function. It should be issued from the local serial terminal or local VGA monitor. As long as no one connects to the modem and there is no alert pending, the alert will be sent to the pager immediately. If the pager does not receive the alert, re-check your setup.

❻ Clears the current alert so that the RMC can capture a new alert. The last alert is stored until a new event overwrites it. The Alert Pending field of the **status** command becomes NO after the alert is cleared.

❼ Verifies the settings. Check that the Alert Enable field is set to Enabled.

NOTE: *If you do not want dial-out paging enabled at this time, enter the* **disable alert** *command after you have tested the dial-out alert function. Alerts continue to be logged, but no paging occurs.*

Continued on next page

Table 7–2 Elements of Dial String and Alert String

Dial String	
	The dial string is case sensitive. The RMC automatically converts all alphabetic characters to uppercase.
ATXDT	AT = Attention.
	X = Forces the modem to dial “blindly” (not seek the dial tone). Enter this character if the dial-out line modifies its dial tone when used for services such as voice mail.
	D = Dial
	T = Tone (for touch-tone)
9,	The number for an outside line (in this example, 9). Enter the number for an outside line if your system requires it.
	, = Pause for 2 seconds.
15085553333	Phone number of the paging service.

Alert String	
,,,,,,	Each comma (,) provides a 2-second delay. In this example, a delay of 12 seconds is set to allow the paging service to answer.
5085553332#	A call-back number for the paging service. The alert string must be terminated by the pound (#) character.
;	A semicolon (;) must be used to terminate the entire string.

7.6.8 Resetting the Escape Sequence

The RMC set escape command sets a new escape sequence.

The new escape sequence can be any character string, not to exceed 14 characters. A typical sequence consists of two or more control characters. It is recommended that control characters be used in preference to ASCII characters. Use the **status** command to verify the new escape sequence before exiting the RMC.

The following example consists of two instances of the Esc key and the letters "FUN." The "F" is not displayed when you set the sequence because it is preceded by the escape character. Enter the **status** command to see the new escape sequence.

Example 7–10 set escape

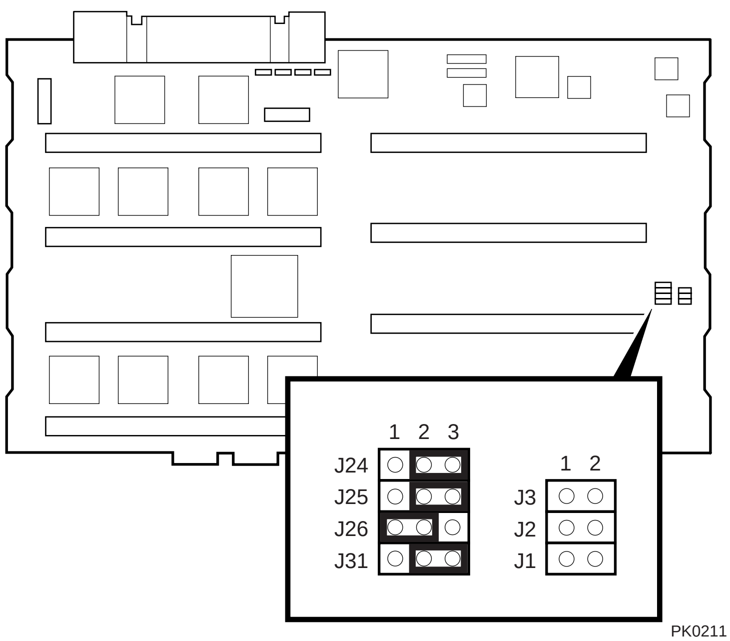
```
RMC> set escape
Escape Sequence: un
RMC> status
.
.
.
Escape Sequence: ^[^[FUN
```

CAUTION: *Be sure to record the new escape sequence. Restoring the default sequence requires moving a jumper on the system motherboard.*

7.7 Resetting the RMC to Factory Defaults

If the non-default RMC escape sequence has been lost or forgotten, RMC must be reset to factory settings to restore the default escape sequence.

Figure 7–4 RMC Jumpers (Default Positions)



NOTE: *J1, J2, and J3 are reserved.*

The following procedure restores the default settings:

1. Shut down the operating system and press the Power button on the operator control panel to the OFF position.
2. Unplug the power cord from each power supply. Wait until the +5V Aux LEDs on the power supplies go off before proceeding.
3. Remove enclosure panels as described in Chapter 8.
4. Remove the system card cage cover and fan cover from the system chassis, as described in Chapter 8.
5. Remove CPU 1 as described in Chapter 8.
6. On the system motherboard, install jumper J25 over pins 1 and 2. See Figure 7–4. (The default jumper positions are shown.)
7. Plug a power cord into one power supply and wait for the control panel to display the message "System is down."
8. Unplug the power cord. Wait until the +5V Aux LED on the power supply goes off before proceeding.
9. Install jumper J25 over pins 2 and 3.
10. Reinstall CPU 1, the card cage cover and fan cover, and the enclosure panels.
11. Plug the power cord into each of the power supplies.

NOTE: *After the RMC has been reset to defaults, perform the setup procedures to enable remote dial-in and call-out alerts. See Section 7.6.6.*

7.8 Troubleshooting Tips

Table 7–3 lists possible causes and suggested solutions for symptoms you might see.

Table 7–3 RMC Troubleshooting

Symptom	Possible Cause	Suggested Solution
You cannot connect to the RMC CLI from the modem.	The RMC may be in Soft Bypass or Firm Bypass mode.	Issue the **show com1_mode** command from SRM and change the setting if necessary. If in Soft Bypass mode, you can disconnect the modem session and reconnect it.
The terminal cannot communicate with the RMC correctly.	System and terminal baud rates do not match.	Set the baud rate for the terminal to be the same as for the system. For first-time setup, suspect the console terminal, since the RMC and system default baud is 9600.
RMC will not answer when the modem is called.	Modem cables may be incorrectly installed.	Check modem phone lines and connections.
	RMC remote access is disabled or the modem was power cycled since last being initialized.	From the local serial terminal or VGA monitor, enter the **set password** and **set init** commands, and then enter the **enable remote** command.
	The modem is not configured correctly.	Modify the modem initialization string according to your modem documentation.

Table 7–3 RMC Troubleshooting (Continued)

Symptom	Possible Cause	Suggested Solution
RMC will not answer when modem is called. (continued from previous page)	On AC power-up, RMC defers initializing the modem for 30 seconds to allow the modem to complete its internal diagnostics and initializations.	Wait 30 seconds after powering up the system and RMC before attempting to dial in.
After the system is powered up, the COM1 port seems to hang or you seem to be unable to execute RMC commands.	There is a normal delay while the RMC completes the system power-on sequence.	Wait about 40 seconds.
New escape sequence is forgotten.		RMC console must be reset to factory defaults.
During a remote connection, you see a "+++" string on the screen.	The modem is confirming whether the modem has really lost carrier. This is normal behavior.	
The message "unknown command" is displayed when you enter a carriage return by itself.	The terminal or terminal emulator is including a line feed character with the carriage return.	Change the terminal or terminal emulator setting so that "new line" is not selected.

Chapter 8
FRU Removal and Replacement

This chapter describes the procedures for removing and replacing FRUs on *Compaq AlphaServer ES40* systems.

Unless otherwise specified, install a FRU by reversing the steps shown in the removal procedures.

NOTE: *If you are installing or replacing CPU cards, memory DIMMs, or PCI cards, become familiar with the location of the card slots and configuration rules. See Chapter 6.*

CAUTION: *Static electricity can damage integrated circuits. Always use a grounded wrist strap (29-26246) and grounded work surface when working with internal parts of a computer system.*

Remove jewelry before working on internal parts of the system.

IMPORTANT! **After you have replaced FRUs and have determined that the system has been restored to its normal operating condition, you must clear the system error information repository (error information logged to the DPR). Use the clear_error all command to clear all errors logged in the FRU EEPROMs and to initialize the central error repository. See Chapter 4 for details on clear_error.**

8.1 FRUs

Table 8–1 lists the FRUs by part number and description. Figure 8–1 shows the location of FRUs in the pedestal/rack systems, and Figure 8–2 shows the location of FRUs in the tower system.

Table 8–1 FRU List

Part #	Description	
Cables		
17-04787-01	Power and signal harness assembly	
17-04785-01	Fan harness assembly	
17-04786-01	Sensor cable harness assembly	
17-03971-07	OCP cable assembly	
17-04678-02	IDE cable assembly	
17-03970-04	Floppy cable assembly	
17-04400-06	Junk I/O connector cable	
17-04867-01	68-conductor SCSI cable	
17-03971-08	10-pin storage subsystem management cable	
17-04914-01	4-conductor storage subsystem management cable	
Fans		
70-40074-01	Fan assembly, 172 MM	Fan 6
70-40073-01	Fan assembly, 120 MM	Fans 1 and 2
70-40073-02	Fan assembly, 120 MM	Fan 5
70-40072-01	Fan assembly, 120 MM	Fan 3
70-40071-01	Fan assembly, 120 MM	Fan 4

Table 8–1 FRU List (Continued)

Part #	Description
CPU Modules	
54-30158-03	500 MHz EV6 4 MB cached CPU
54-30158-05	Acceptable substitute for 54-24801-03
54-30158-06	500 MHz EV6 4 MB cached CPU (EV6 V2.4)
54-30158-07	500 MHz EV6 4 MB cached CPU (EV6 V2.4)
Memory DIMMs	
54-25053-BA	64 MB, 200-pin DIMM
54-24941-EA	128 MB, 200-pin DIMM
54-24941-FA	256 MB, 200-pin DIMM
54-24941-JA	512 MB, 200-pin DIMM
Other Modules and Components	
70-33894-01	OCP
54-25582-01	8-slot MMB for 200-pin DIMMs
54-25582-02	4-slot MMB for 200-pin DIMMs
70-31349-01	Speaker assembly
30-50802-02	Hard drive cage assembly, 4 slot, 1.6-in.
54-25385-01	System motherboard
54-25575-01	I/O connector module
54-25573-01	PCI backplane, 10-slot
54-25573-02	PCI backplane, 6-slot

Continued on next page

Table 8–1 FRU List (Continued)

Part #	Description
30-49448-01	Power supply, 720 Watts
SN-LKQ46-Ax	Keyboard, OpenVMS
SN-LKQ47-Ax	Keyboard, Tru64 UNIX
SN-LKQ97-Ax	Keyboard, Windows NT
SN-PBQWS-WA	Mouse, 3-button
12-37977-02	Key for doors
3X-RRD32-AC 3R-A0284-AA	CD-ROM drive, half-height
RX23L-AC	Floppy drive

8.1.1 Power Cords

Tower enclosures ordered in North America include a 120 V power cord. Non-North American orders require one country-specific power cord. Pedestal systems ordered in North American include two 120 V power cords. Non-North American orders require two country-specific power cords.

Table 8–2 lists the country-specific power cords for tower and pedestal systems.

Table 8–2 Country-Specific Power Cords

Power Cord	Country	Length
BN26J-1K	North American 120 V	75 in.
3X-BN46F-02	Japan	2.5 m
BN19H-2E	Australia, New Zealand	2.5 m
BN19C-2E	Central Europe	2.5 m
BN19A-2E	UK, Ireland	2.5 m
BN19E-2E	Switzerland	2.5 m
BN19K-2E	Denmark	2.5 m
BN19M-2E	Italy	2.5 m
BN19S-2E	Egypt, India, South Africa	2.5 m

8.1.2 FRU Locations

Figure 8–1 and Figure 8–2 show the location of FRUs in the pedestal and rackmount configurations.

Figure 8–1 FRUs — Front/Top (Pedestal/Rack View)

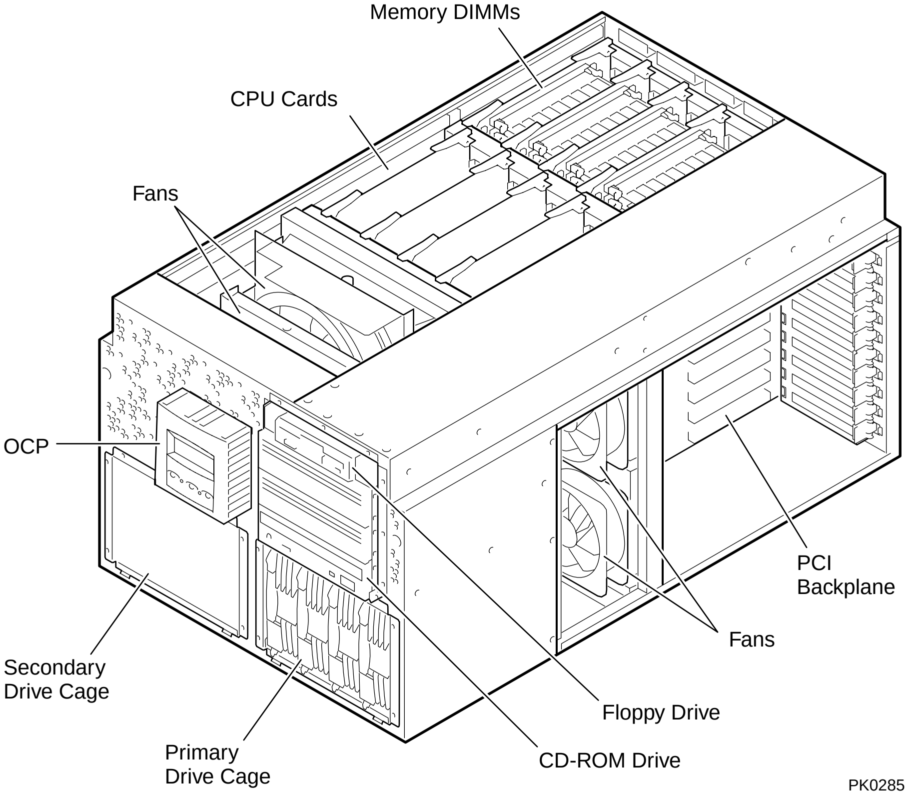
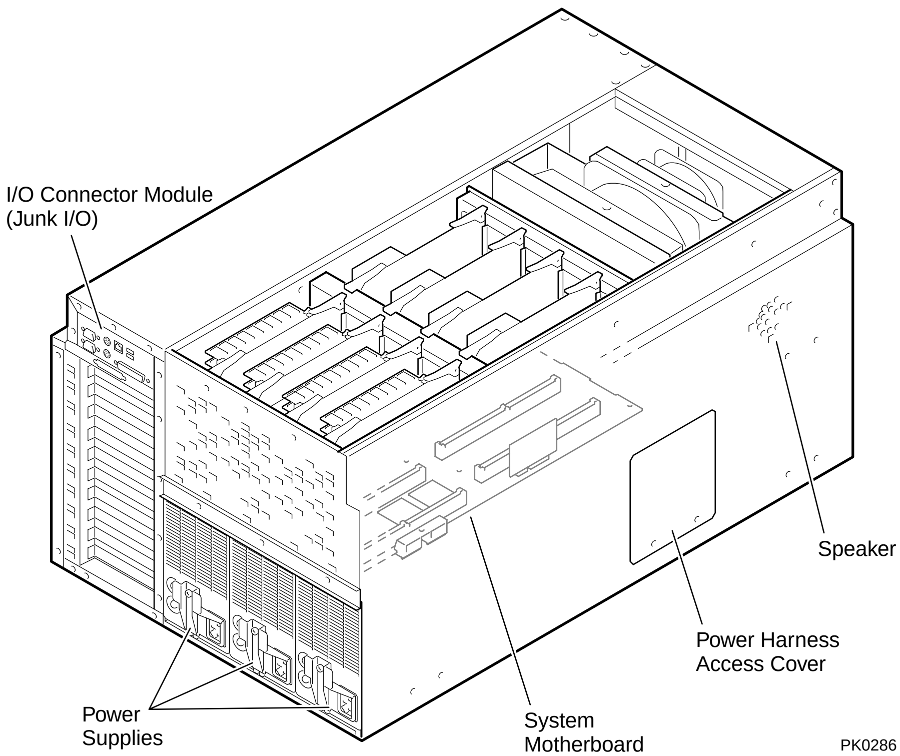


Figure 8–2 FRUs — Rear (Pedestal/Rack View)



8.1.3 Important Information Before Replacing FRUs

The system must be shut down before you replace most FRUs. The exceptions are power supplies, individual fans, and hard drives. After replacing FRUs you must clear the system error information repository with the SRM clear_error all command.

Tools

You need the following tools to remove or replace FRUs.

- Phillips #2 screwdriver (a magnetic screwdriver is recommended)
- Allen wrench (3 mm)
- Anti-static wrist strap

Hot-Plug FRUs

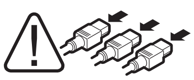
The following are hot-plug FRUs. You can replace them while the system is operating.

- Power supplies
- Individual fans
- Hard drives (hot-swappable if supported by the operating system)

Before Replacing Non Hot-Plug FRUs

Follow the procedure below before replacing any non hot-plug FRU.

1. Shut down the operating system.
2. Shut down power to external options, where appropriate.
3. Turn off power to the system.
4. Unplug the power cord from each power supply.



WARNING: To prevent injury, unplug the power cord from each power supply before installing components.

After Replacing FRUs

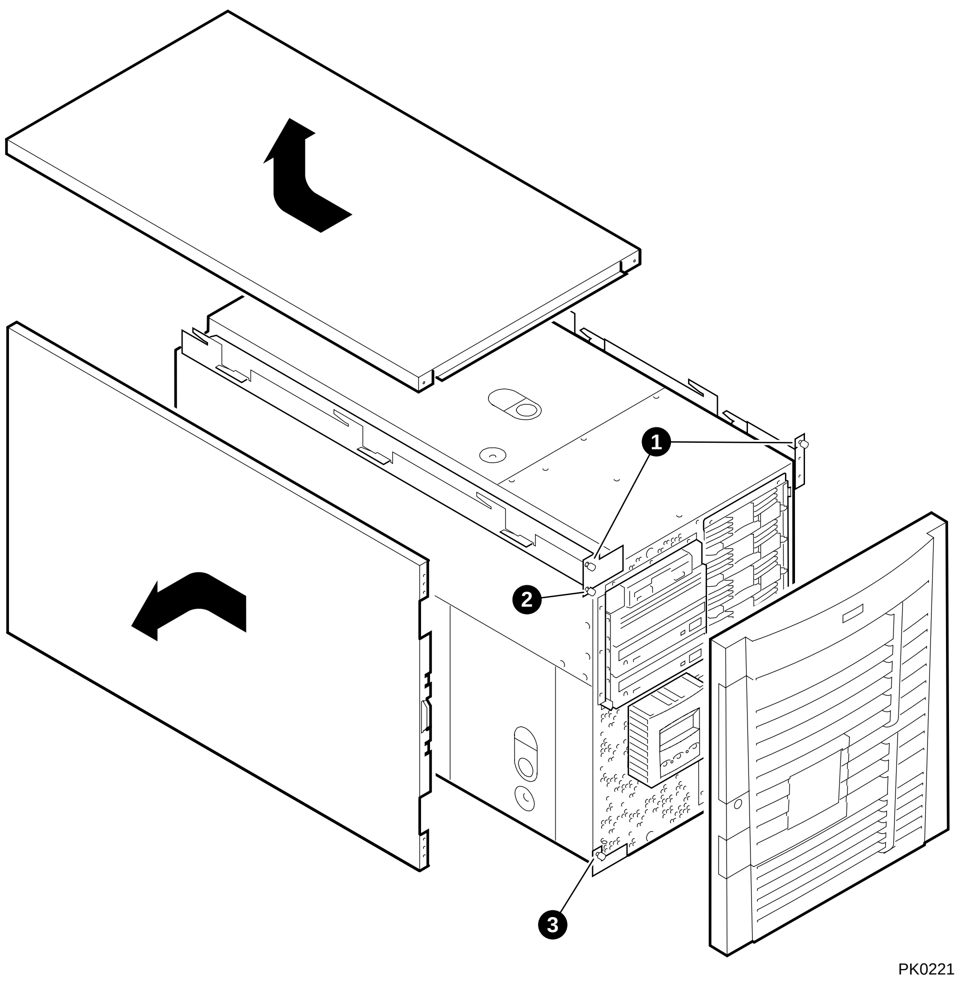
After you have replaced FRUs and have determined that the system has been restored to its normal operating condition, you must clear the system error information repository (error information logged to the DPR).

Use the **clear_error all** command to clear all errors and initialize the central error repository. See Chapter 4 for details.

8.2 Removing Enclosure Panels on a Tower or Pedestal

Open and remove the front door. Loosen the captive screws that allow you to remove the top and side panels.

Figure 8–3 Enclosure Panel Removal (Tower)

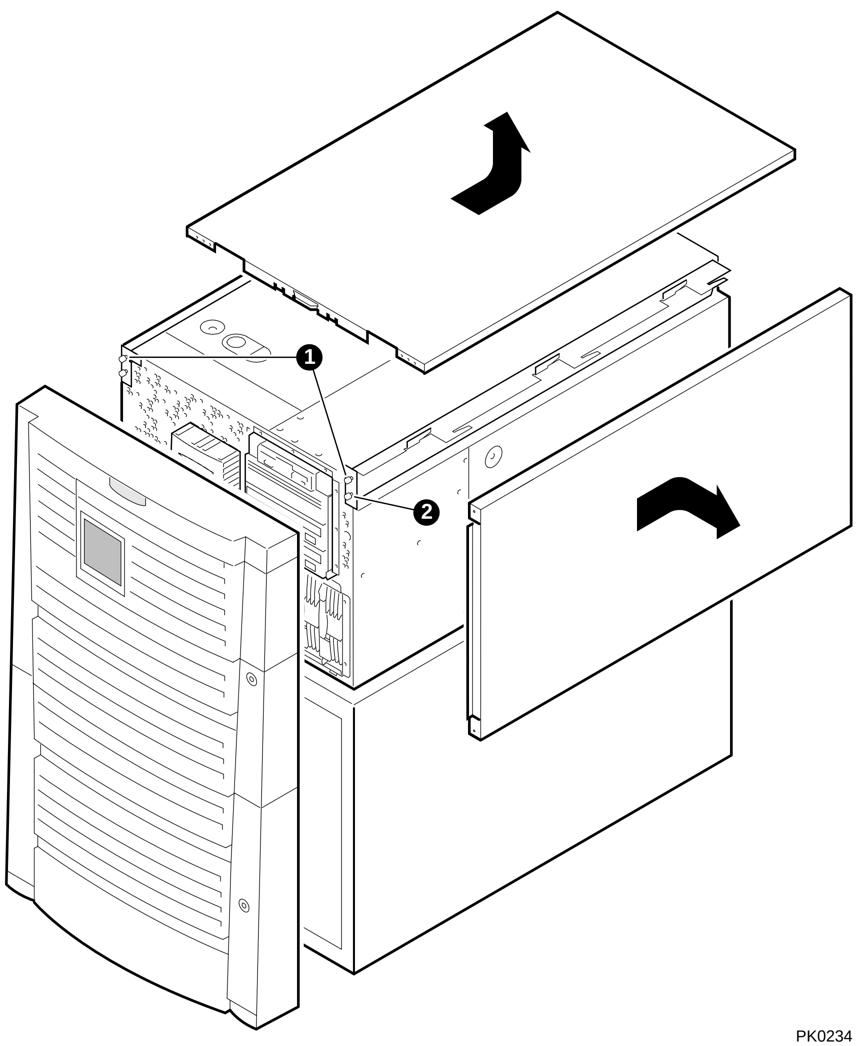


To Remove Enclosure Panels from a Tower

The enclosure panels are secured by captive screws.

1. Remove the front door.
2. To remove the top panel, loosen the top left and top right captive screws ❶. Slide the top panel back and lift it off the system.
3. To remove the left panel, loosen the captive screw ❷ at the top and the captive screw ❸ at the bottom. Slide the panel back and then tip it outward. Lift it off the system.

Figure 8–4 Enclosure Panel Removal (Pedestal)



To Remove Enclosure Panels from a Pedestal

The enclosure panels are secured by captive screws.

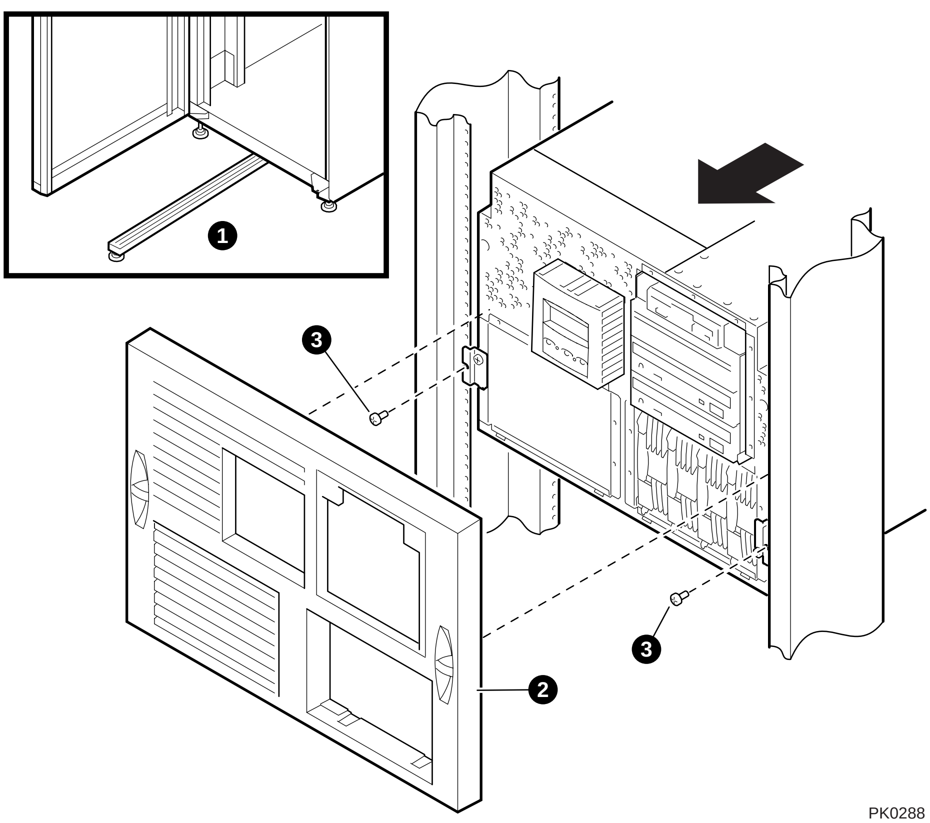
1. Open and remove the front doors.
2. To remove the top enclosure panel, loosen top left and top right captive screws ❶. Slide the top panel back and lift it off the system.
3. To remove the right enclosure panel, loosen the captive screw shown in ❷. Slide the panel back and then tip it outward. Lift the panel from the three tabs.

8.3 Accessing the System Chassis in a Cabinet

In a rackmount system, the system chassis is mounted to slides.

WARNING: Pull out the stabilizer bar and extend the leveler foot to the floor before you pull out the system. This precaution prevents the cabinet from tipping over.

Figure 8–5 Accessing the Chassis in a Cab



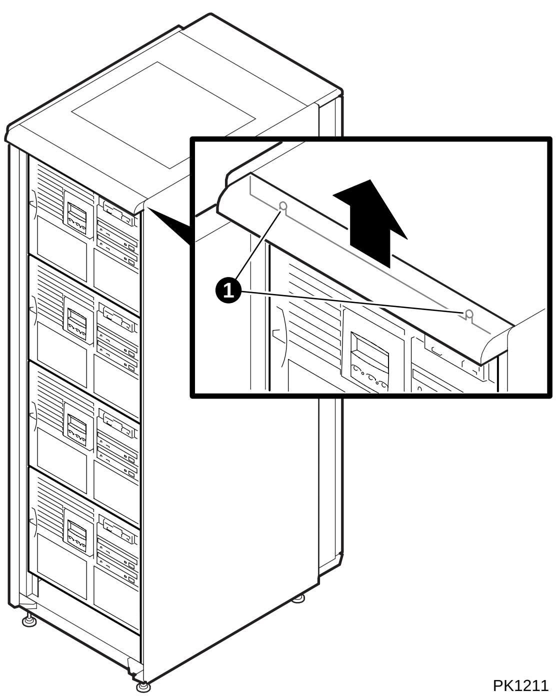
PK0288

To Gain Access to the System Chassis

1. Open the front door of the cabinet.
2. Pull out the stabilizer bar ❶ at the bottom of the cabinet until it stops.
3. Extend the leveler foot at the end of the stabilizer bar to the floor.
4. Snap out the front bezel ❷.
5. Remove and set aside the two screws ❸ (one per side), if present, that secure the system to the cabinet.
6. Pull the system out until it locks.

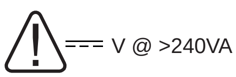
NOTE: *In a 4-system H9A10 cabinet, remove the top overhang bezel by loosening the two screws ❶.*

Figure 8–6 H9A10 Overhang Bezel

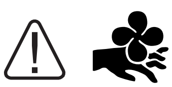


8.4 Removing Covers from the System Chassis

The system chassis has three covers: the fan cover, the system card cage cover, and the PCI card cage cover. Remove a cover by loosening the quarter-turn captive screw, pulling up on the ring, and sliding the cover from the system chassis.



WARNING: High current area. Currents exceeding 240 VA can cause burns or eye injury. Avoid contact with parts or remove power prior to access.



WARNING: Contact with moving fan can cause severe injury to fingers. Avoid contact or remove power prior to access.

Figure 8–7 and Figure 8–8 show the location and removal of covers on the tower and pedestal/rackmount systems, respectively. The numbers in the illustrations correspond to the following:

❶ 3mm Allen captive quarter-turn screw that secures each cover.

❷ Spring-loaded ring that releases cover. Each cover has a ring.

❸ Fan area cover. This area contains the 6.75-in main system fan and a redundant fan.

❹ System card cage cover. This area contains CPUs, memory DIMMs, MMBs, and system motherboard. To remove the system card cage cover, you must first remove the fan area cover ❸. An interlock switch shuts the system down when you remove the system card cage cover.

❺ PCI card cage cover. This area contains PCI cards, the PCI backplane, and four fans.

Figure 8–7 Covers on the System Chassis (Tower)

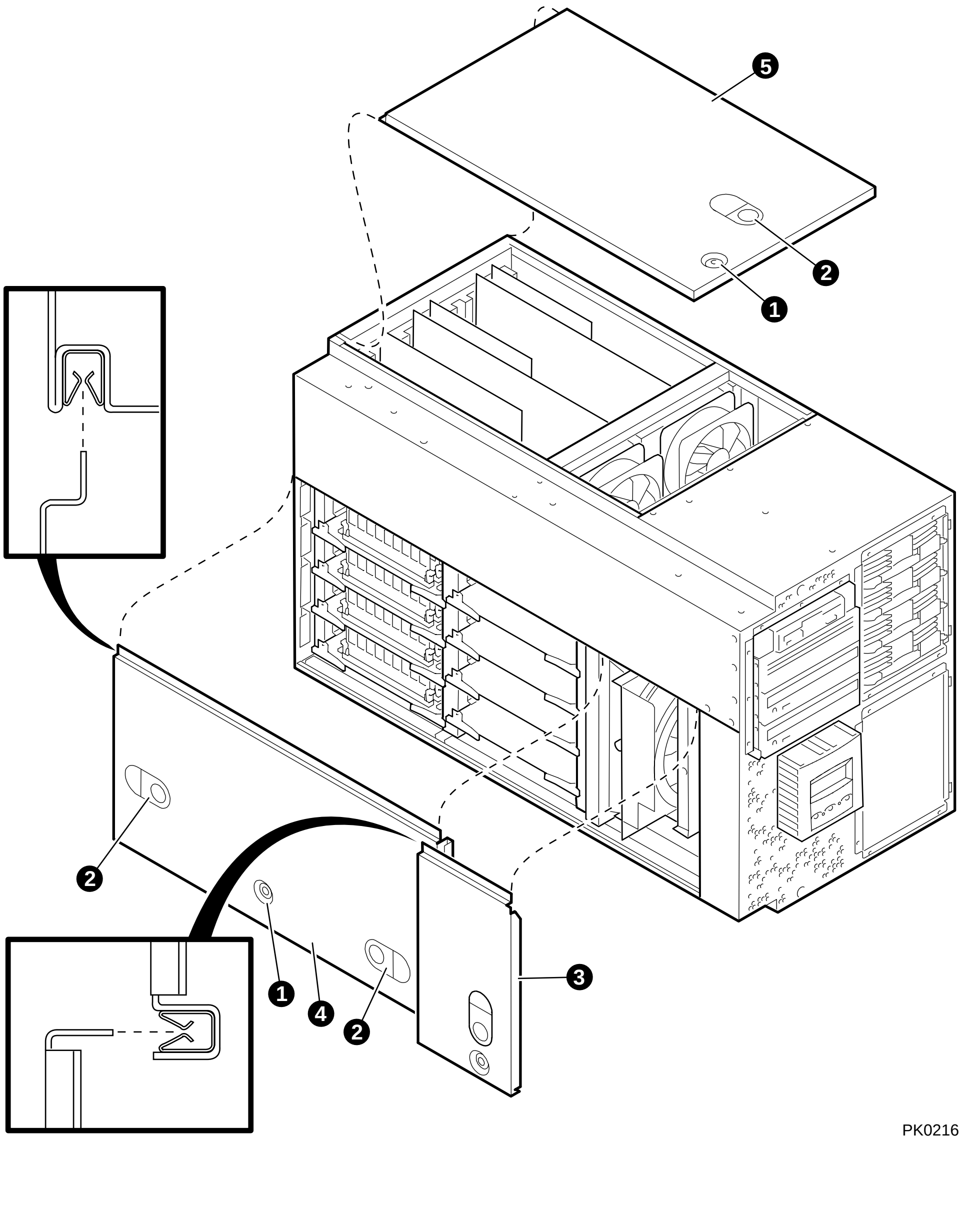
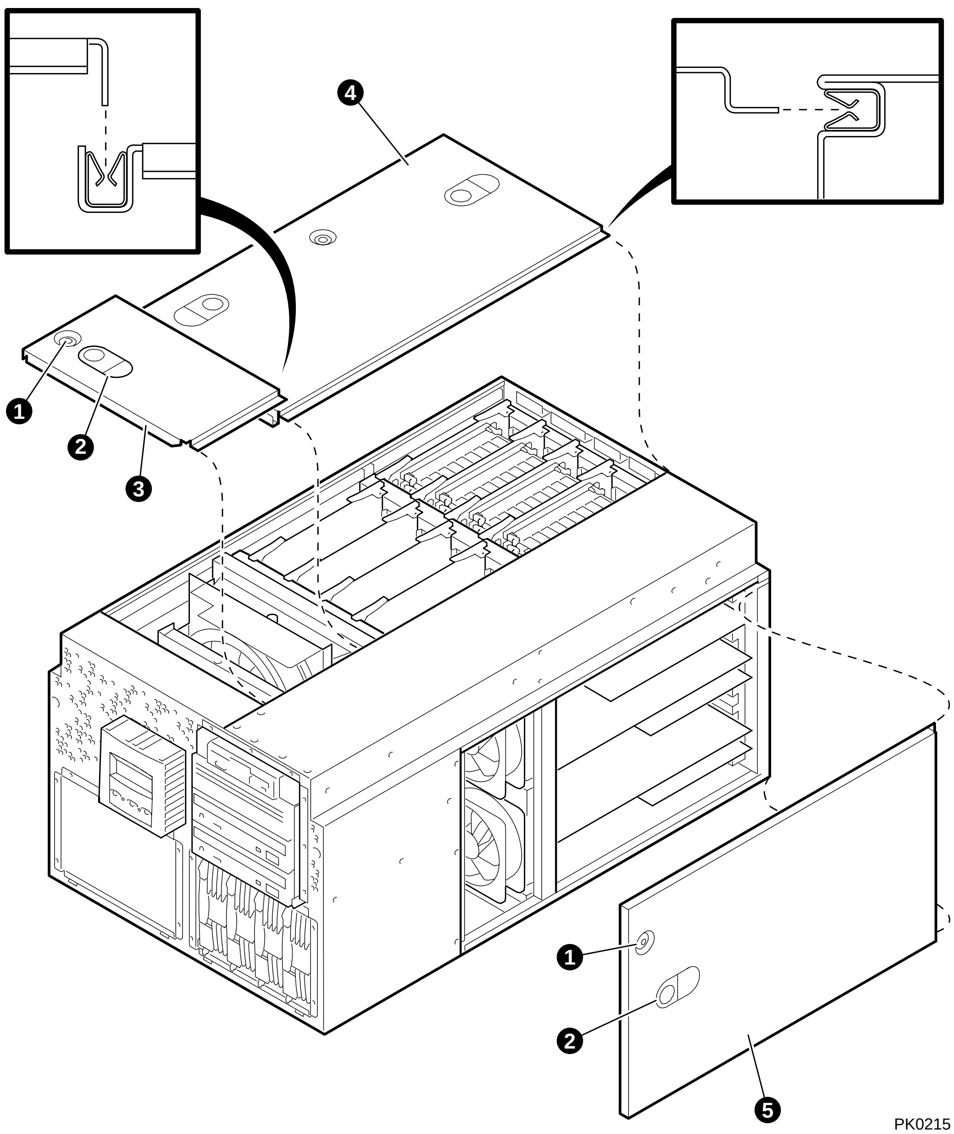
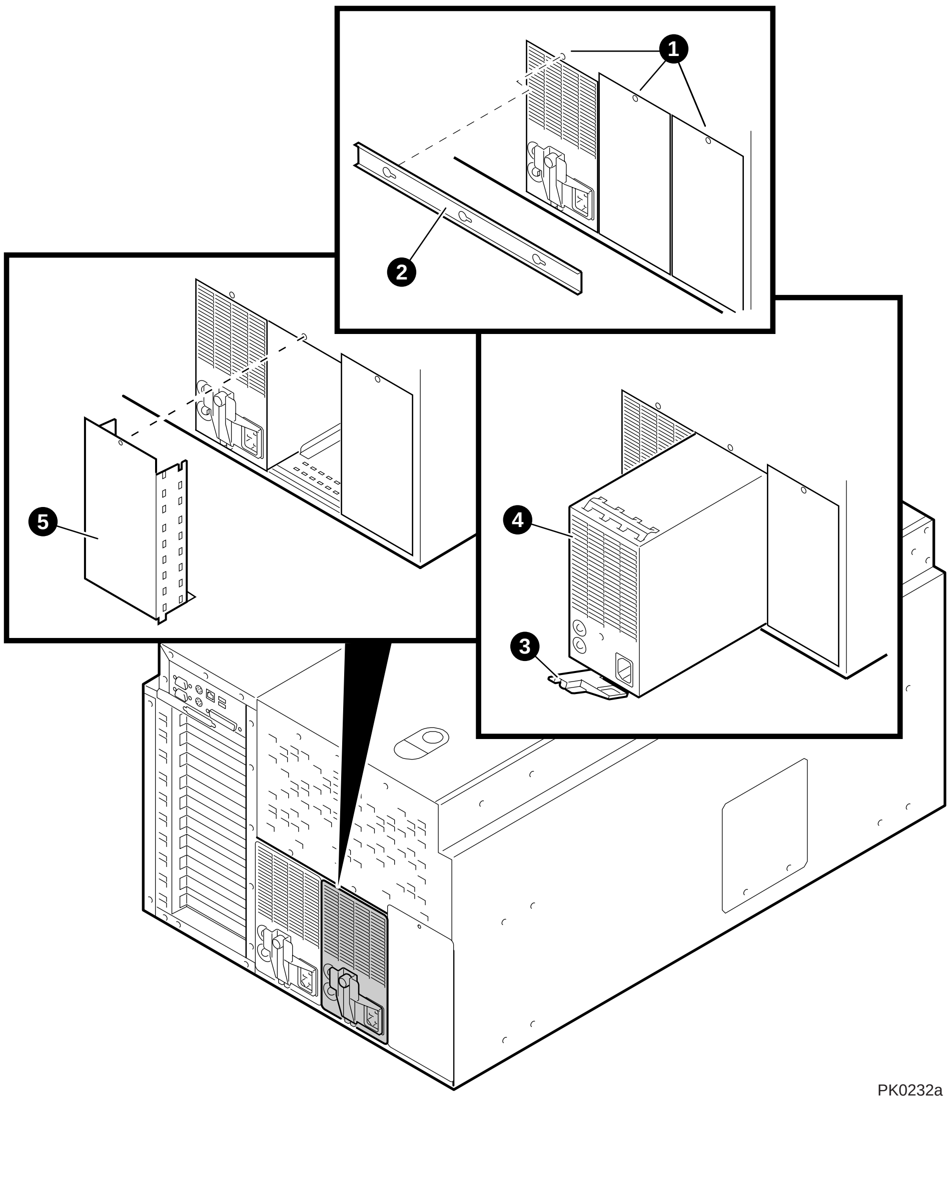


Figure 8–8 Covers on the System Chassis (Pedestal/Rack)



8.5 Power Supply

Figure 8–9 Removing a Power Supply



WARNING: Hazardous voltages are contained within the power supply. Do not attempt to service. Return to factory for service.

The power supply is a hot-plug component. As long as the system has a redundant supply, you can replace a supply while the system is running.

Removing a Power Supply

1. Unplug the AC power cord.
2. Loosen the three Phillips screws ❶ that secure the power supply bracket. (Do not remove the screws.) Remove the bracket ❷.
3. Loosen the captive screw on the latch ❸ and swing the latch to unlock the power supply.
4. Pull the power supply ❹ out of the system.

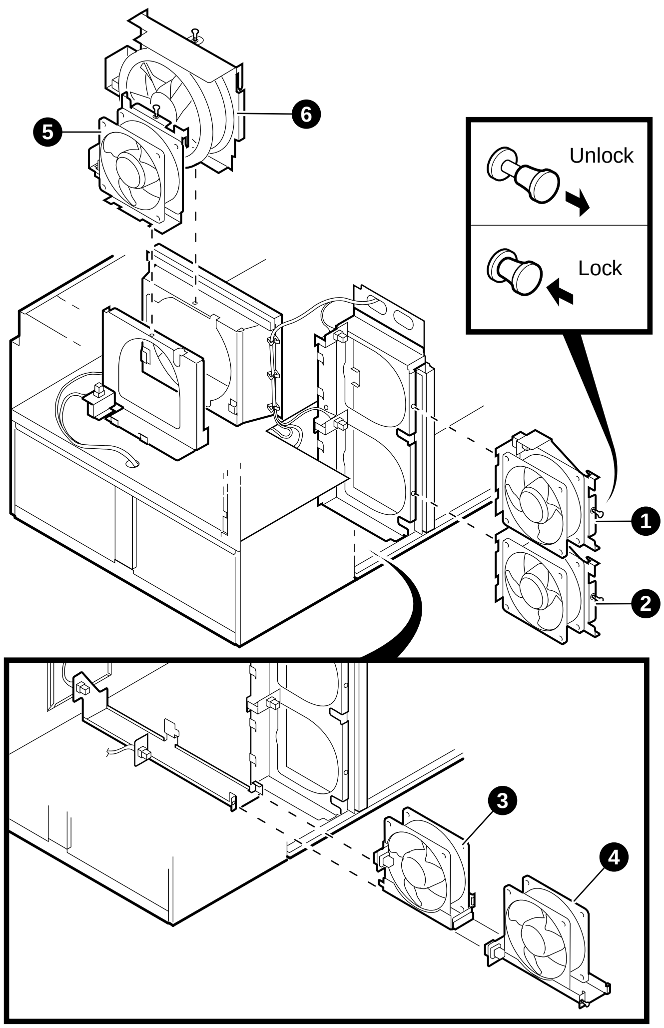
NOTE: *When installing an additional supply, remove the screw and blank cover ❺ on the slot into which you are installing the supply.*

Verification

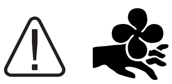
1. Plug the AC power cord into the supply. Wait a few seconds for the POK LED to light.
2. Check that both power supply LEDs are lit.

8.6 Fans

Figure 8–10 Replacing Fans



The fans are hot-plug components. You can replace individual fans while the system is running.



WARNING: Contact with moving fan can cause severe injury to fingers. Avoid contact or remove power prior to access.

Replacing Fans

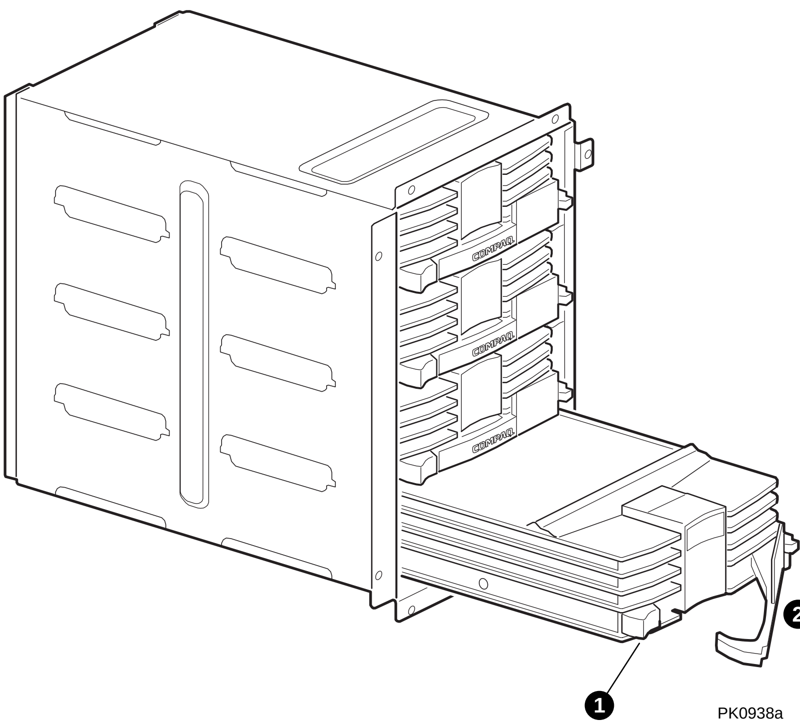
1. Remove the cover from the fan area (fans ❺ and ❻) or the PCI card cage (fans ❶,❷,❸, and ❹).
2. Pull the pop-up latch to unlock it, and lift the fan out of the system. Fan ❸ has no pop-up latch. It is held in place by fan ❹.
3. Install the new fan, taking care to align it as it slides in. Press the pop-up latch to lock the fan in place.
4. Replace the cover to the fan area or the PCI card cage.

Verification — RMC

1. Invoke the remote management console.
2. Enter the **env** command to verify the fan status.

8.7 Hard Disk Drives

Figure 8–11 Removing a Hard Drive



Hard drives are hot-plug components.

CAUTION: *Before replacing a hard disk drive, ensure that the SCSI controller and/or the operating system support hot-swapping of drives. Otherwise, shut down the operating system and return to the SRM console level before starting the replacement procedure.*

Removing a Hard Disk Drive

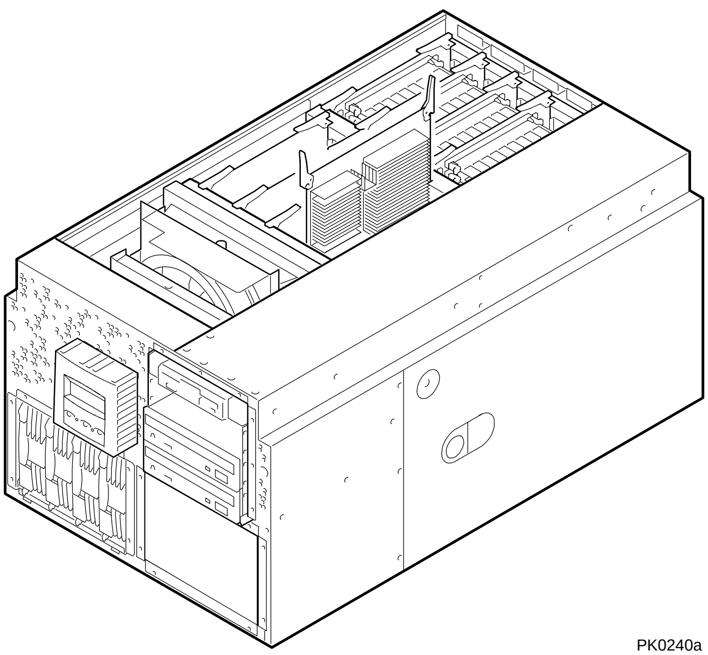
1. Access the storage drive area.
2. Push the button ❶ to release the plastic handle ❷ on the front of the drive carrier. Pull out the plastic handle toward you and slide the drive out.

NOTE: *Remove the blank cover from the next available slot before installing an additional hard disk drive.*

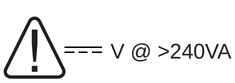
8.8 CPUs

You must shut the system down before adding or replacing a CPU.

Figure 8–12 Removing CPU Cards



WARNING: CPU cards have parts that operate at high temperatures. Wait 2 minutes after power is removed before touching any module.



WARNING: High current area. Currents exceeding 240 VA can cause burns or eye injury. Avoid contact with parts or remove power prior to access.

Replacing a CPU Card

1. Remove the covers from the fan area and the system card cage.
2. Pull up on the clips at each end of the card and remove the card.
3. Install the new CPU card in the connector and push down firmly on both clips simultaneously.

NOTE: *When installing an additional CPU, remove the blank CPU air deflector from the next available slot.*

Verification — SRM Console

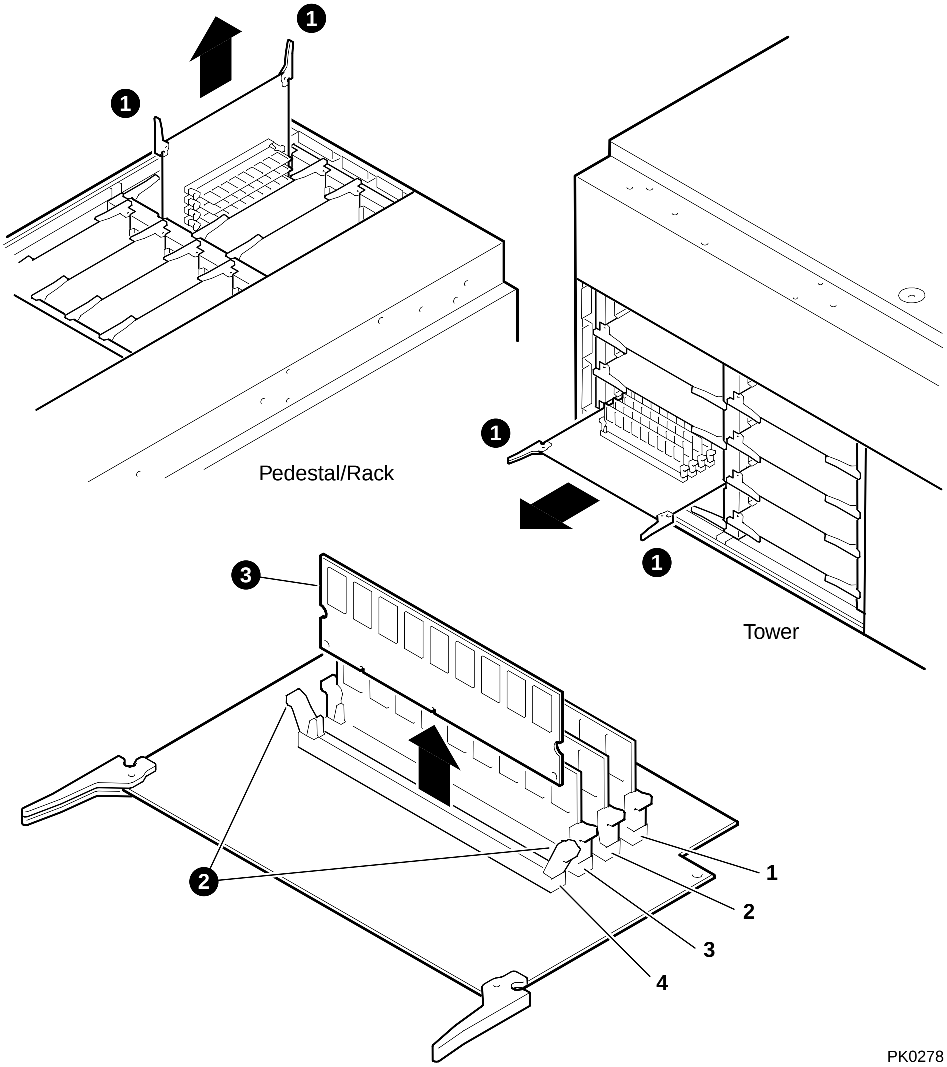
1. Turn on power to the system.
2. During power-up, observe the screen display. The newly installed CPU should appear in the display.
3. Issue the **show config** command. The new CPU should be listed as one of the processors.

Verification — AlphaBIOS

1. Start AlphaBIOS Setup, select **Display System Configuration**, and press Enter.
2. Using the arrow keys, select **Systemboard Configuration** and check the Processor field to determine how many processors the system sees.

8.9 Memory DIMMs

Figure 8–13 Removing MMBs and DIMMs



WARNING: Memory DIMMs have parts that operate at high temperatures. Wait 2 minutes after power is removed before touching any module.

=== V @ >240VA

WARNING: High current area. Currents exceeding 240 VA can cause burns or eye injury. Avoid contact with parts or remove power prior to access.

CAUTION: *DIMMs come in two types, stacked or unstacked. See Chapter 6 before replacing DIMMs.*

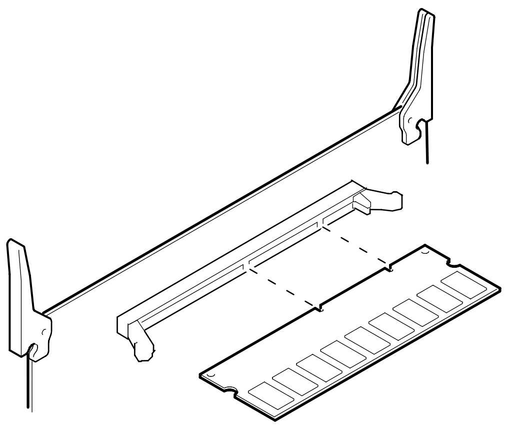
Replacing DIMMs

You must shut the system down before adding or replacing DIMMs.

1. Remove the fan cover and the system card cage cover.
2. Release the clips ❶ that secure the MMB to the system backplane and slide out the MMB.
3. Release the clips ❷ on the MMB slot containing the bad DIMM and remove the DIMM ❸.

Continued on next page

Figure 8–14 Aligning DIMM in MMB



4. Install the new DIMM. Align the notches on the gold fingers with the connector keys (Figure 8–14) and secure the DIMM with the clips on the MMB slot.
5. Reinstall the MMB and secure it to the system backplane with the clips.

Verification — SRM Console

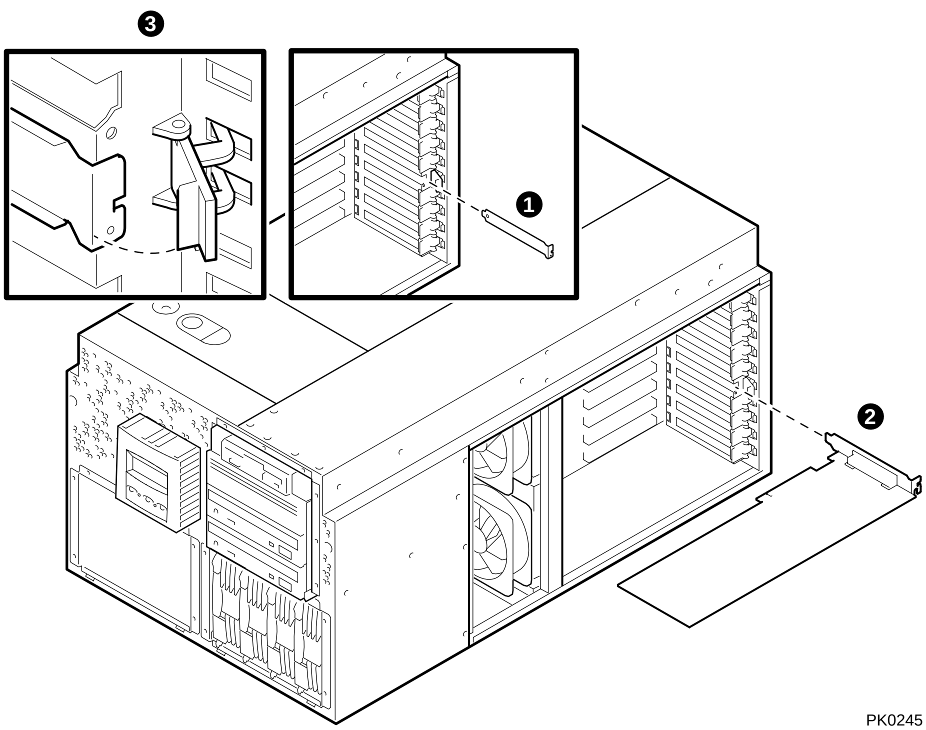
1. Turn on power to the system.
2. During power-up, observe the screen display for memory.
3. Issue the **show memory** command to display the total amount of memory in the system.

Verification — AlphaBIOS Console

1. Start AlphaBIOS Setup, select **Display System Configuration**, and press Enter.
2. Using the arrow keys, select **Memory Configuration** to display the new memory.

8.10 PCI Cards

Figure 8–15 Installing or Replacing a PCI Card



WARNING: To prevent fire, use only modules with current limited outputs. See National Electrical Code NFPA 70 or Safety of Information Technology Equipment, Including Electrical Business Equipment EN 60 950.

V @ >240VA

WARNING: High current area. Currents exceeding 240 VA can cause burns or eye injury. Avoid contact with parts or remove power prior to access.

Installing or Replacing a PCI Card

You must shut the system down before adding or replacing a PCI card.

1. Remove the cover to the PCI card cage.
2. If installing a new card, remove and discard the bulkhead filler plate ❶ from the PCI slot.
3. If replacing a card, disconnect and remove the failed card.
4. Insert the new PCI card ❷ into the connector.

NOTE: *Some full-length PCI cards may have extender brackets for installing into ISA/EISA-style card cages. Remove the extender brackets before installing such a card.*

---

5. Secure the card to the card cage with the latch ❸.

Verification — SRM Console

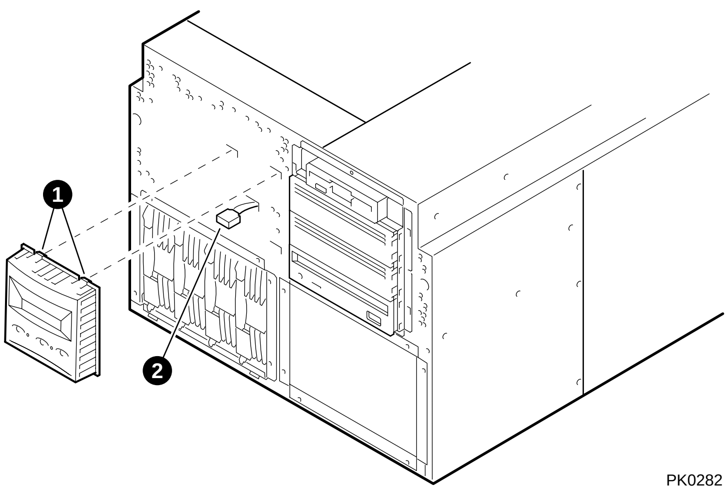
1. Turn on power to the system.
2. During power-up, observe the screen display for PCI information. The new option should be listed in the display.
3. Issue the SRM **show config** command. Examine the PCI bus information in the display to make sure that the new option is listed.
4. Enter the SRM **show device** command to display the device name of the new option.

Verification — AlphaBIOS Console

1. Start AlphaBIOS Setup, select **Display System Configuration**, and press Enter.
2. Using the arrow keys, select **PCI Configuration** to determine that the new option is listed.

8.11 OCP Assembly

Figure 8–16 Removing the OCP Assembly



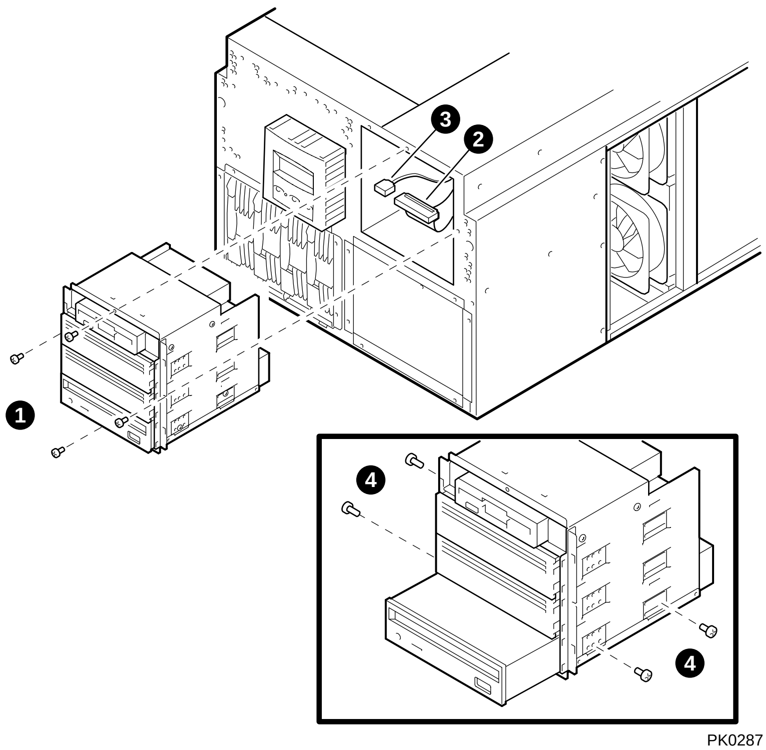
Removing the OCP Assembly

You must shut the system down before removing the OCP assembly.

1. Press the two tabs ❶ on the top of the OCP assembly to release it.
2. Rotate the assembly toward you and lift it out of the two bottom tabs.
3. Disconnect the control panel cable ❷.

8.12 Removable Media

Figure 8–17 Removing a 5.25-Inch Device



Removing a 5.25-Inch Removable Media Device

You must shut the system down before adding or replacing a removable media device.

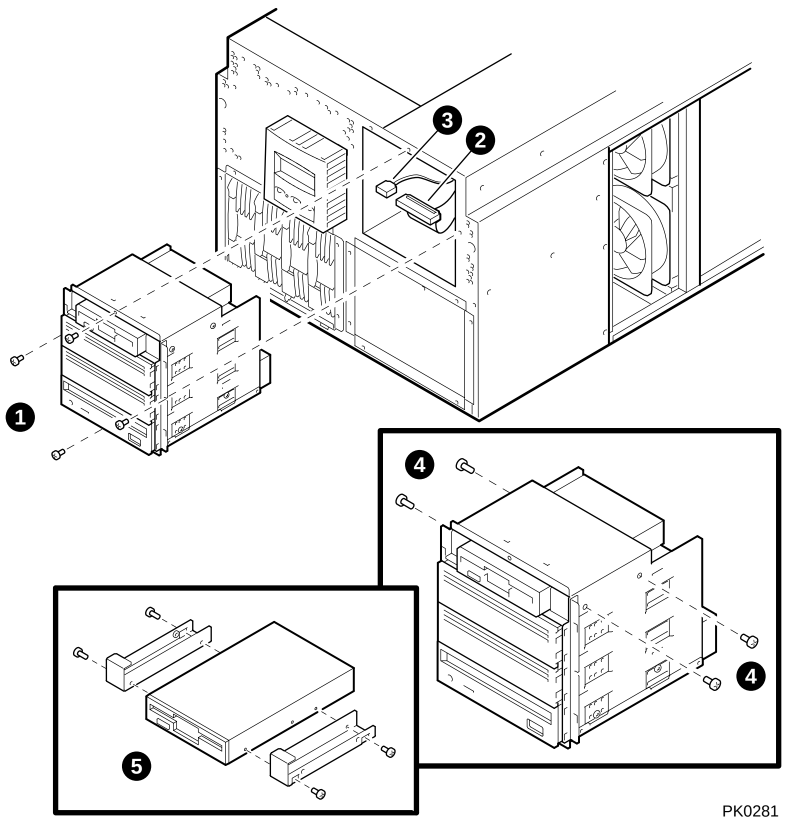
4. Remove the cover to the PCI card cage.
5. Remove and set aside the four screws ❶ that secure the removable media cage.
6. Unplug the signal cable ❷ and power cable ❸ from all devices except the floppy.
7. Remove the cage.
8. Unplug the signal cable and power cable from the floppy.
9. Remove the four screws ❹ that secure the device and set aside the screws. Slide the device out of the storage slot.

NOTE: *When installing a removable media device, remove the blank bezel from the next available slot. For installation instructions, see the Compaq AlphaServer ES40 Owner's Guide.*

For information on installing disk cages, see the *Compaq AlphaServer ES40 Release Notes*.

8.13 Floppy Drive

Figure 8–18 Removing the Floppy Drive



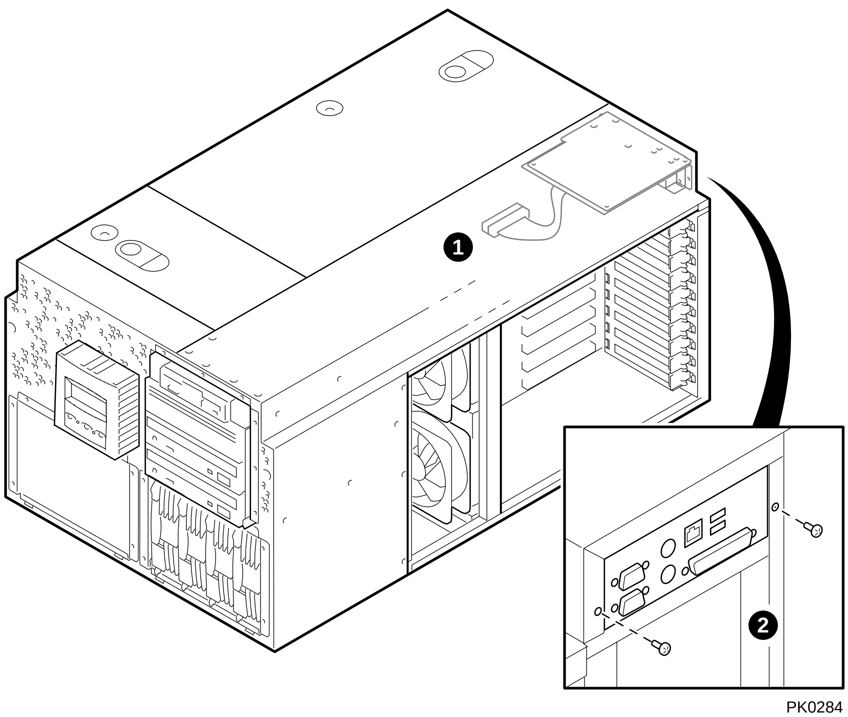
Removing the Floppy Drive

You must shut the system down before removing the floppy drive.

1. Remove the cover to the PCI card cage.
2. Remove and set aside the four screws ❶ that secure the removable media cage.
3. Unplug the signal cable ❷ and power cable ❸ from all devices except the floppy.
4. Remove the cage.
5. Unplug the signal cable and power cable from the floppy.
6. Remove the four screws ❹ that secure the floppy drive, and slide the drive out.
7. Remove the mounting brackets ❺ (two screws in each bracket) from the drive.

8.14 I/O Connector Assembly

Figure 8–19 Removing the I/O Connector Assembly



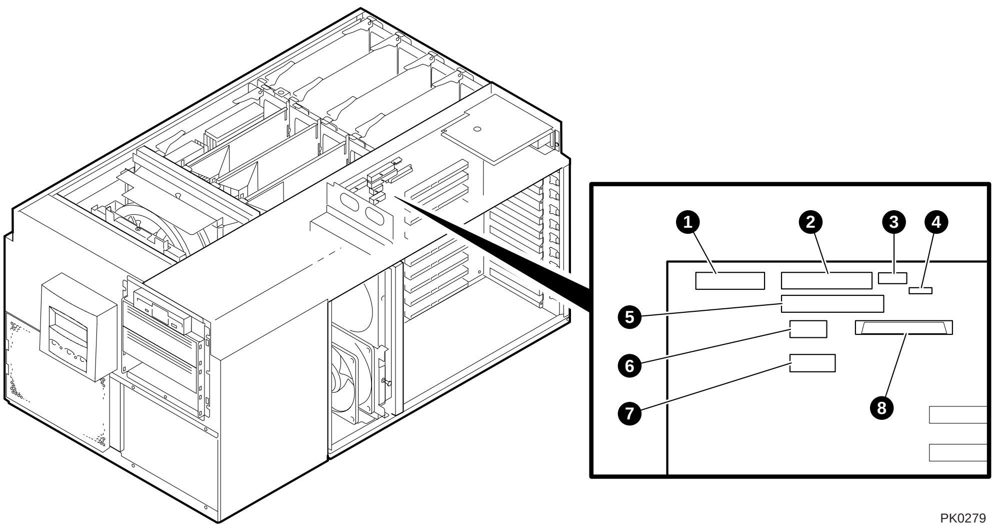
Removing the I/O Connector Assembly

You must shut the system down before removing the I/O connector assembly.

1. Unplug all I/O connectors from the rear of the unit.
2. Remove the cover from the PCI card cage.
3. Unplug the 68-pin signal cable ➊.
4. Remove the two screws ➋ that secure the assembly to the back of the unit.
5. Pull the assembly out through the PCI area.

8.15 PCI Backplane

Figure 8–20 Cables Connected to PCI Backplane



	Connecting Cable	Connects To:
❶	17-04785-01	Fans
❷	17-03970-04	Floppy
❸	17-04786-01	Cover sensors
❹	70-31349-01	Speaker
❺	17-04678-02	CD-ROM
❻	17-03971-07	OCP
❼	17-04914-01 (if present)	Storage disk cage
❽	17-04400-06	I/O controller module

WARNING: High current area. Currents exceeding 240 VA can cause burns or eye injury. Avoid contact with parts or remove power prior to access.

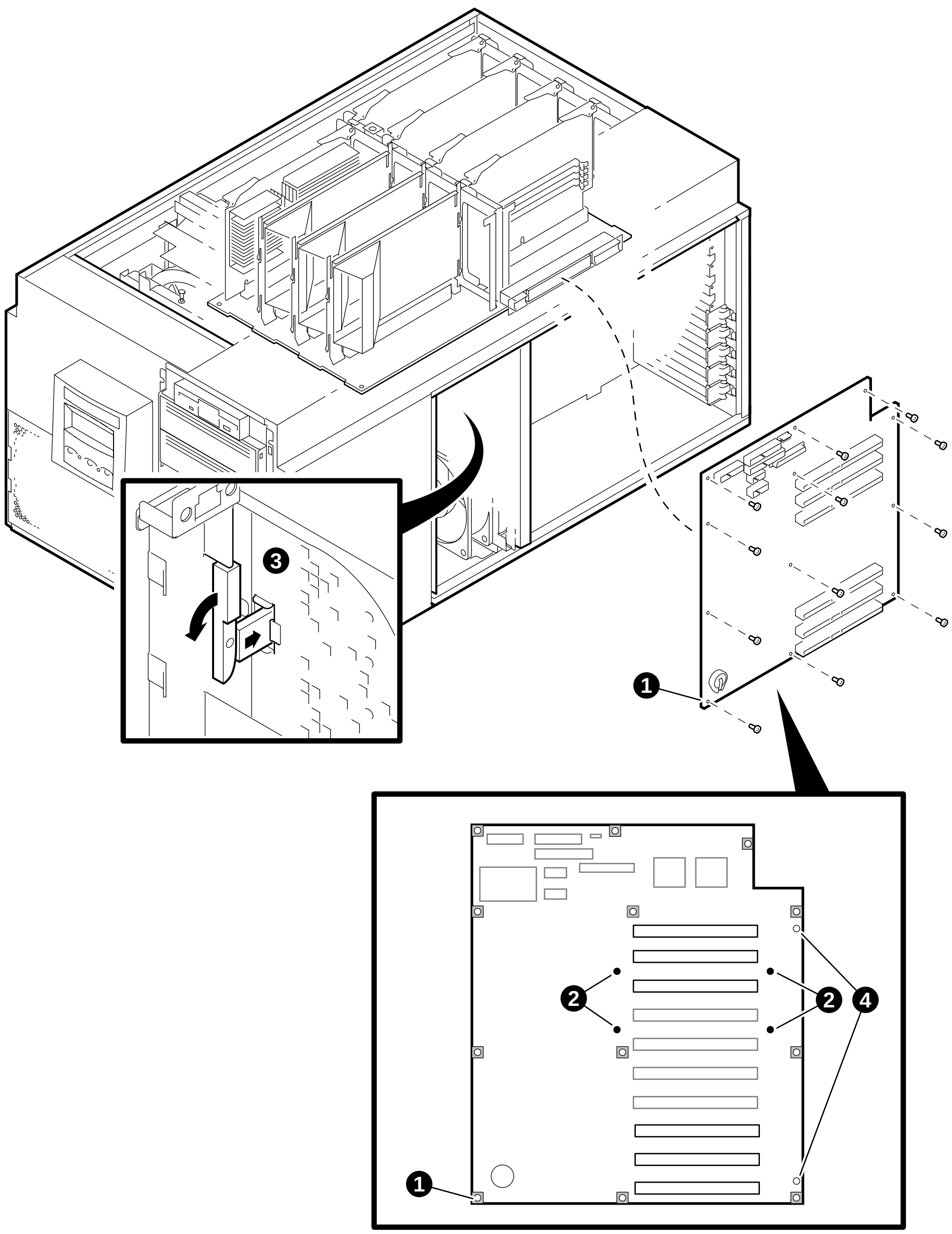
Disconnecting the Cables

You must shut the system down before accessing the PCI area.

1. Remove the cover to the PCI card cage.
2. Record the location of installed PCI cards.
3. Remove all external cables from the PCI bulkheads in the rear of the unit. Remove internal cables from PCI cards.
4. Unlatch and remove the cards from the card cage.
5. Disconnect cables connected to the PCI backplane. See Figure 8–20.
6. Remove the top fan (pedestal/rack orientation) or left fan (tower orientation). This permits access to an ejector lever needed for removing the PCI backplane.

Continued on next page

Figure 8–21 Removing the PCI Backplane



Removing the PCI Backplane

CAUTION: *When removing the PCI backplane, be careful not to flex the board. Flexing the board may damage the BGA component connections.*

1. Remove the 12 screws ❶ that secure the PCI backplane to the chassis.

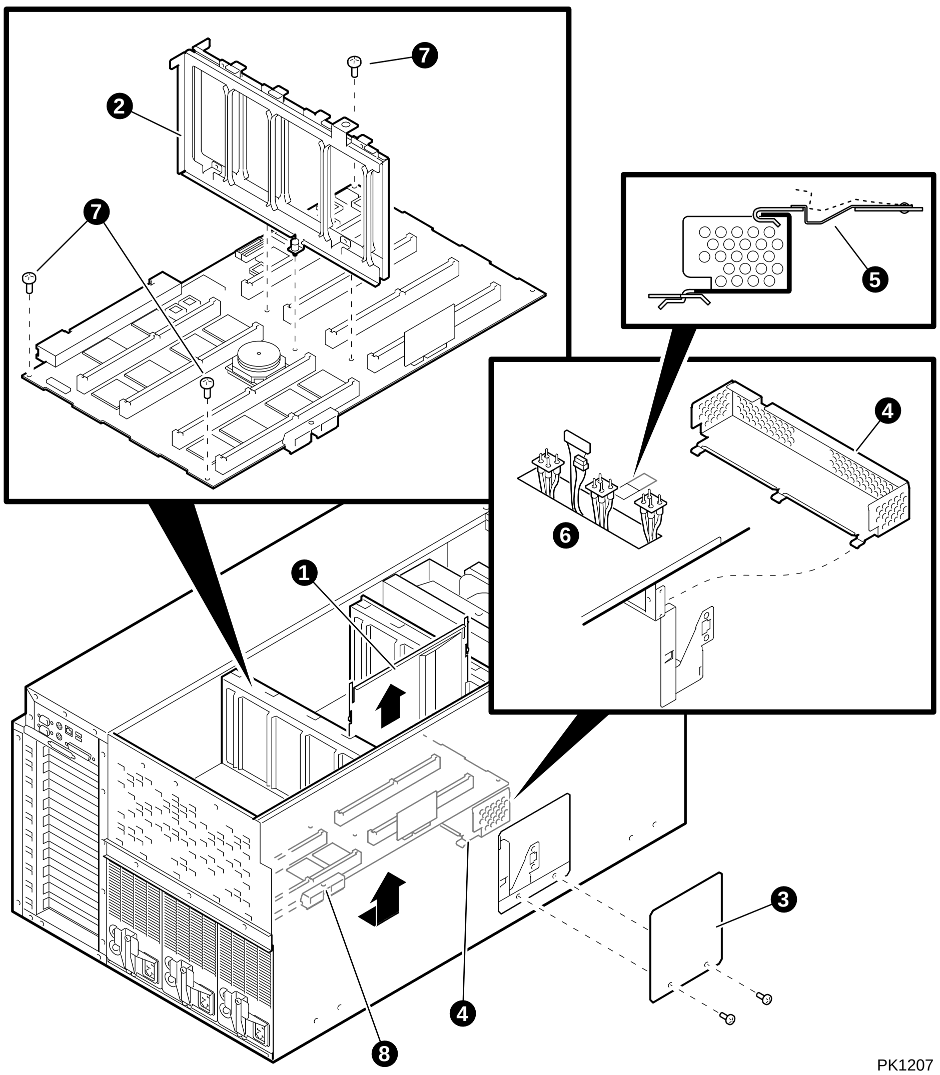
   CAUTION: *Do not remove the four additional nonwashered screws ❷. Removing them inactivates the built-in mechanism for extracting the PCI backplane from the system.*

2. Use the ejector lever ❸ in the fan area to separate the PCI backplane from the system motherboard, then lift the backplane out of the chassis.

NOTE: *When installing a new PCI backplane, align the backplane on the guide pins ❹, and press the board firmly until it is seated. Seating the PCI backplane requires considerable pressure. When seating the PCI backplane in a cabinet, a second person should brace the chassis to ensure that no excessive stress is placed on the rails.*

8.16 System Motherboard

Figure 8–22 Removing the System Motherboard



WARNING: CPUs and memory DIMMs have parts that operate at high temperatures. Wait 2 minutes after power is removed before touching any module.

CAUTION: *When removing the system motherboard, be careful not to flex the board. Flexing the board may damage the BGA component connections.*

NOTE: *Removing the system motherboard requires the removal of other FRUs. Review the removal procedures for the fans, MMBs, CPUs, and drive cage before beginning the system motherboard removal procedure.*

1. Remove the three covers from the system chassis.
2. Remove fans 3 and 4 in the PCI area (the inner fans).
3. Record the positions of the MMBs and CPUs, and remove the MMBs and CPUs.
4. Remove the CPU air flow deflectors ❶, if present.
5. Loosen the three captive Phillips screws holding the middle support bracket ❷. The screws pop up when sufficiently loosened. Pull the bracket straight out.
6. Remove the second drive cage (left cage in pedestal/rack, bottom cage in tower), if installed, or the blank panel.
7. Remove the two Phillips flat-head screws that secure the small cover ❸ to the left side (pedestal/rack) or bottom (tower) of the system and remove the panel. Set aside the screws. (Removing the small cover provides better access to the power harness bracket.)
8. Remove the power harness bracket ❹ as follows: Push up on the spring latch ❺ to release the bracket, slide the bracket forward, and remove it.

Continued on next page

9. Unplug the five connectors ❻ on the bottom of the system motherboard.
10. Remove the three Phillips screws ❼ that secure the system motherboard.
11. A white plastic flange ❽ and two holes in the sheet metal under the flange are used to help disengage the system motherboard from the PCI backplane. Insert a screwdriver through the hole in the flange into the closest hole and pry the system motherboard away from the PCI backplane. Insert the screwdriver into the second hole that is now exposed and pry again to fully disengage the system motherboard connector from the PCI backplane.
12. Extract the system motherboard.

After installing a new motherboard:

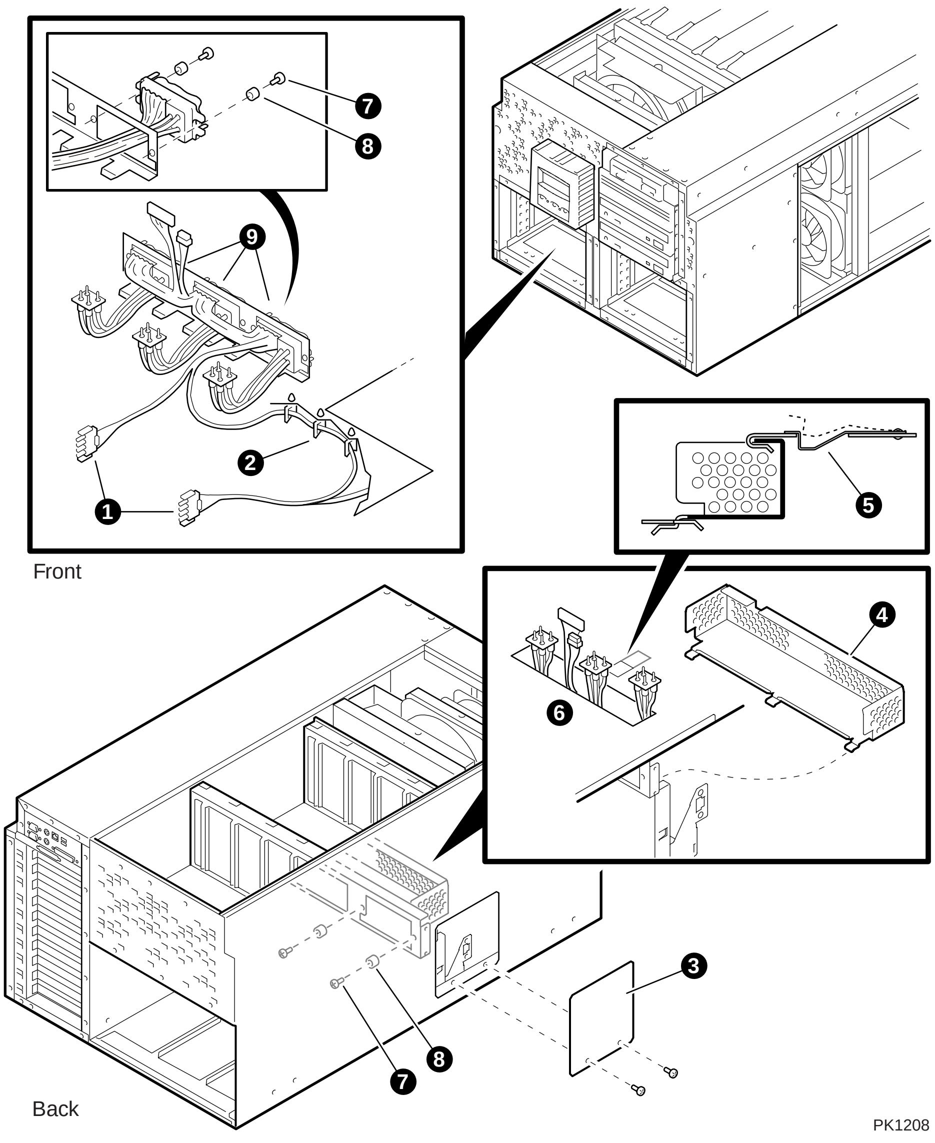
1. Power up to the P00>>> prompt.
2. Enter the **clear_error all** command.
3. Enter the **set sys_serial_num** command to set the system serial number. For example:

```
P00>>> set sys_serial_num NI900100022
```

The serial number will be propagated to all FRU devices that have EEPROMs.

8.17 Power Harness

Figure 8–23 Removing the Power Harness



NOTE: *Removing the power harness requires the removal of other system FRUs. Review the removal procedures for the power supplies, fans, and drive cage before beginning the harness removal procedure.*

1. Remove the power supplies and any blank power supply panels.
2. Remove the cover to the PCI card cage.
3. Remove fans 4 and 3 (the inner fans).
4. Unplug the connectors to each removable media device (except the floppy).
5. Remove the four screws that secure the removable media cage. Slide out the cage to access the floppy power connector. Disconnect the floppy power connector and slide the cage back in.
6. Unplug the power connector ❶ to the drive cage or cages.
7. Remove the harness from the cable clamps ❷.
8. Remove the second drive cage (left cage in pedestal/rack, bottom cage in tower), if installed, or the blank panel.
9. Remove the two Phillips flat-head screws that secure the small cover ❸ to the left side (pedestal/rack) or bottom (tower) of the system and remove the panel. Set aside the screws. (Removing the small cover provides better access to the power harness bracket.)
10. Remove the power harness bracket ❹ as follows: Push up on the spring latch ❺ to release the bracket, slide the bracket forward, and remove it .
11. Unplug the five connectors ❻ on the bottom of the system motherboard.
12. Remove the two screws ❼ and two plastic bushings ❽ on each of the three power supply connectors ❾. The screws are located deep inside the power supply cavity. Set aside the screws and bushings for reinstallation.
13. Starting with the left connector (as viewed from the rear of the system), pull the connector to the right and angle it so that you can push the left end out through the opening.
14. Remove the power harness.

Appendix A
SRM Console Commands

This appendix lists the SRM console commands that are most frequently used with the *Compaq AlphaServer* ES40 family of systems.

Table A–1 SRM Commands Used on ES40 Systems

Command	Function
alphabios	Loads and starts the AlphaBIOS console.
boot	Loads and starts the operating system.
buildfru	Initializes I^2Cbus EEPROM data structures for the named FRU.
cat el	Displays the console event log. Same as more el, but scrolls rapidly. The most recent errors are at the end of the event log and are visible on the terminal screen.
clear error	Clear errors logged in the FRU EEPROMs as reported by the show error command.
continue	Resumes program execution on the specified processor or on the primary processor if none is specified.
crash	Forces a crash dump at the operating system level.
deposit	Writes data to the specified address of a memory location, register, or device.
edit	Invokes the console line editor on a RAM file or on the user power-up script, "nvram," which is always invoked during the power-up sequence.
examine	Displays the contents of a memory location, register, or device.

Table A–1 SRM Commands Used on ES40 Systems (Continued)

Command	Function
exer	Exercises one or more devices by performing specified read, write, and compare operations.
floppy_write	Runs a write test on the floppy drive to determine whether you can write on the diskette.
grep	Searches for "regular expressions"—specific strings of characters—and prints any lines containing occurrences of the strings.
hd	Dumps the contents of a file (byte stream) in hexadecimal and ASCII.
help *command*	Displays information about the specified console command.
info	Displays registers and data structures.
init	Resets the SRM console and reinitializes the hardware.
kill	Terminates a specified process.
kill_diags	Terminates all executing diagnostics.
man	Displays information about the specified console command.
memexer	Runs a requested number of memory tests in the background.
memtest	Tests a specified section of memory.
more el	Same as cat el, but displays the console event log one screen at a time.
net -ic	Initialize the MOP counters for the specified Ethernet port.
net -s	Displays the MOP counters for the specified Ethernet port.
nettest	Runs loopback tests for PCI-based Ethernet ports. Also used to test a port on a "live" network.
prcache	Initializes and displays the status of the PCI NVRAM.

Table A–1 SRM Commands Used on ES40 Systems (Continued)

Command	Function
rmc	Invokes the remote management console from the local VGA monitor.
set *envar*	Sets or modifies the value of an environment variable.
show *envar*	Displays the state of the specified environment variable.
show config	Displays the logical configuration at the last system initialization.
show device	Displays a list of controllers and bootable devices in the system.
show error	Reports errors logged in the FRU EEPROMs .
show fru	Displays information about field replaceable units (FRUs), including CPUs, memory DIMMs, and PCI cards.
show memory	Displays information about system memory.
show pal	Displays the versions of Tru64 UNIX and OpenVMS PALcode.
show power	Displays information about system environmental characteristics, including power supplies, system fans, CPU fans, and temperature.
show_status	Displays the progress of diagnostic tests. Reports one line of information for each executing diagnostic.
show version	Displays the version of the SRM console program installed on the system.
sys_exer	Exercises the devices displayed with the show config command
sys_exer -lb	Runs console loopback tests for the COM2 serial port and the parallel port during the sys_exer test sequence.
test	Verifies the configuration of the devices in the system.
test -lb	Runs loopback tests for the COM2 serial port and the parallel port in addition to verifying the configuration of devices.

Appendix B
Jumpers and Switches

This chapter lists and describes the configuration jumpers and switches on the system motherboard and PCI board. Sections are as follows:

- RMC and SPC Jumpers on System Motherboard
- TIG/SROM Jumpers on System Motherboard
- Clock Generator Switch Settings
- Jumpers on PCI Board
- Setting Jumpers

B.1 RMC and SPC Jumpers on System Motherboard

The RMC jumpers can be used to override the RMC defaults. For example, if a high-speed modem is connected to COM1, you can disable J31 to prevent RMC from receiving characters that might cause interference. The SPC jumpers are reserved.

Figure B–1 RMC and SPC Jumpers

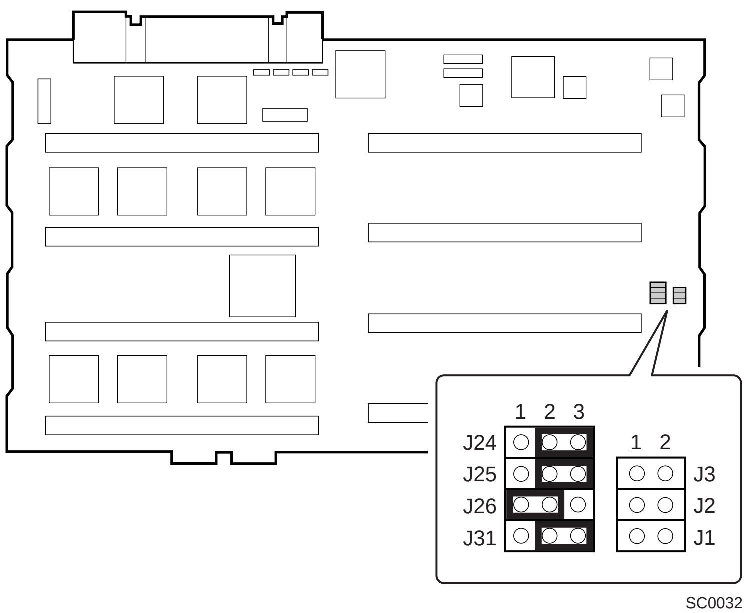


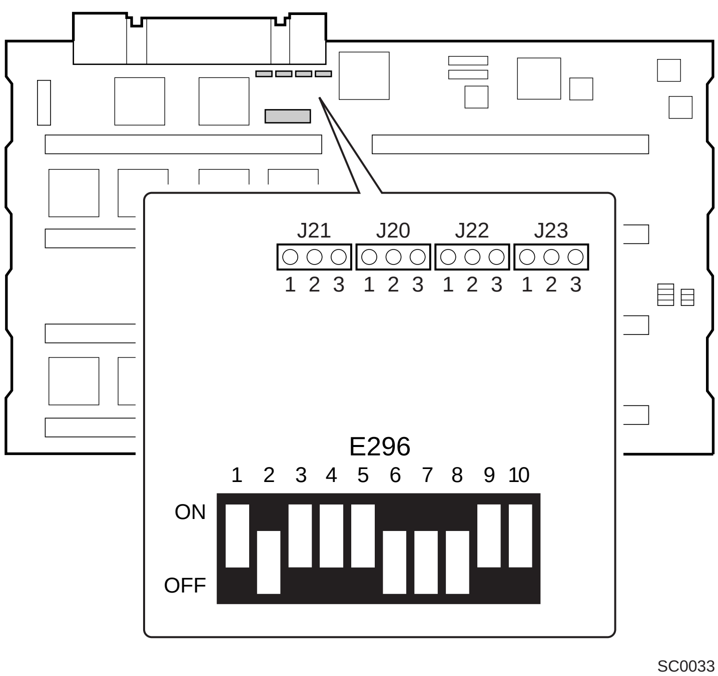
Table B–1 RMC/SPC Jumper Settings

Jumper	Description
J24	1–2: Disables RMC flash update 2–3: Enables RMC flash update (default) Disabling RMC flash update prevents other operators from erasing or updating the RMC.
J25	1–2: Sets RMC back to defaults 2–3: Normal RMC operating mode (default) If the RMC escape sequence is set to something other than the default, and you have forgotten the sequence, RMC must be reset to factory settings to restore the default escape sequence. See Chapter 8 for the reset procedure.
J26	1–2: Causes system to shut down if over-temperature limit is reached (default) 2–3: Permits system to continue running at over-temperature.
J31	1–2: Disables COM1 bypass 2–3: Allows RMC to control COM1 bypass (default) No jumper installed: Forces COM1 bypass If a high-speed modem is connected to COM1 (MMJ), removing J31 prevents RMC from receiving characters that might cause interference.
J1	Not installed (default). When installed, bypasses power-up checks of processors by system power controller.
J2	Reserved (not installed).
J3	Reserved (not installed).

B.2 TIG/SROM Jumpers on System Motherboard

TIG/SROM jumpers allow you to load the TIG if flash RAM is corrupted or load the fail-safe loader (FSL) if SRM firmware is corrupted.

Figure B–2 TIG/SROM Jumpers



NOTE: *See Chapter 3 for instructions on activating the FSL.*

Table B–2 TIG/SROM Jumper Descriptions

Jumper	Description
J21	1–2: Load TIG from flash RAM (default) 2–3: Load TIG from serial ROM. This setting allows you to load the TIG if the flash RAM is corrupted.
J20	Must be in default positions over pins 1 and 2 to enable FSL. FIR_FUNC2 (bit 2) 1–2 = 0, 2–3 = 1
J22	Jumper for enabling fail-safe loader (FSL) FIR_FUNC1 (bit 1) 1–2= 0, 2–3= 1
J23	Must be in default positions over pins 1 and 2 to enable FSL. FIR_FUNC0 (bit 0) 1–2= 0, 2–3 = 1

Firmware Function Table (FIR_FUNC)

Bits 210	Meaning
000	Normal
001	Prevent flash loads. Load from SROM.
010	Load from floppy
111	Lock console. Prevents the writing of flash from CPUs.

Switchpack E296 sets the clock speed for the system motherboard. The settings should not be changed.

SW1	SYS_EXT_DELAY1 (off)
SW2	SYS_EXT_DELAY0 (on)
SW3	SYS_FILL_DELAY (off)
SW4	CPU_CFWD_PSET (off)
SW5	PCI_CLK_DIV_IN1 (off)
SW6	PCI_CLK_DIV_IN0 (on)
SW7	Y_DIV3 (on)
SW8	Y_DIV2 (on)
SW9	Y_DIV1 (off)
SW10	Y_DIV0 (off)

B.3 Clock Generator Switch Settings

Switchpack E16 on the system motherboard sets the frequency of the main clock on the system motherboard. The settings should not be changed.

Figure B–3 CSB Switchpack E16

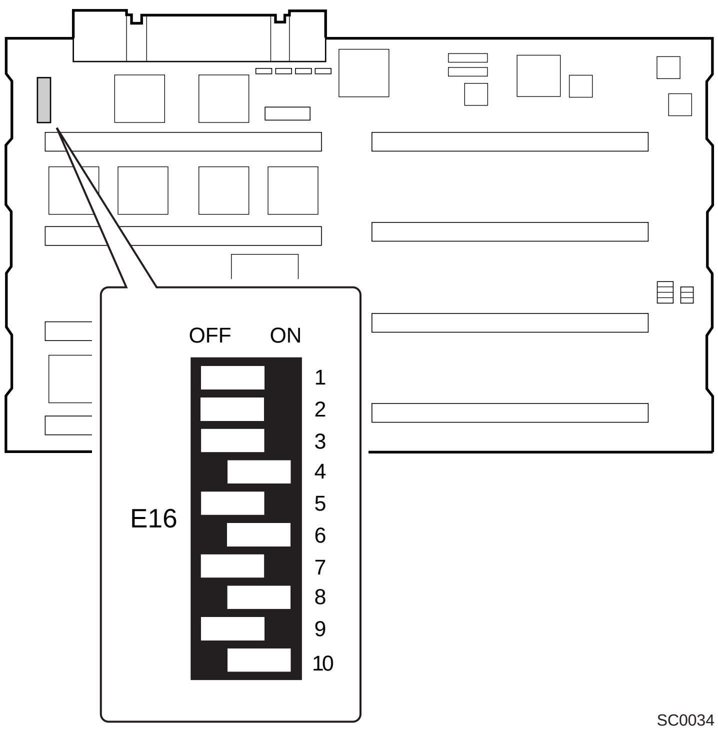


Table B–3 Clock Generator Settings

SW1	M0 (on)
SW2	M1 (on)
SW3	M2 (on)
SW4	M3 (off)
SW5	M4 (on)
SW6	M5 (off)
SW7	M6 (on)
SW8	N0 (off)
SW9	N1 (on)
SW10	XTAL_SEL (OFF)

B.4 Jumpers on PCI Board

You can set J31 on the PCI board to force DTR so that a modem will not be disconnected if the system is power cycled. Check J13 if the system is losing time or the operating system comes up with a very inaccurate time.

Figure B–4 PCI Board Jumpers

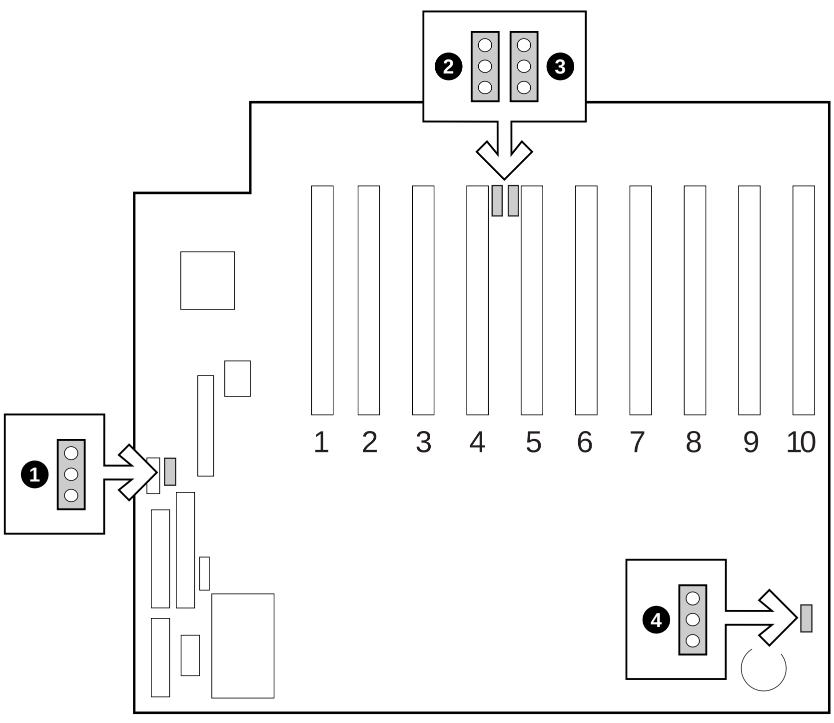


Table B–4 PCI Board Jumper Descriptions

	Jumper	Description
❶	J31	1–2: Do not force COM1 DTR 2–3: Force COM1 DTR (default) This jumper allows you to force DTR. The default position prevents disconnection of the modem on a power cycle.
❷	J20	1–2: Enable PCI 0 power management events (PME). 2–3: Disable PCI 0 PME (default) This jumper is reserved.
❸	J21	1–2: Enable PCI 1 PME 2–3: Disable PCI 1 PME (default) This jumper is reserved.
❹	J13	1–2: Enable VBAT to real-time clock (RTC) chip (default) 2–3: Disable VBAT to RTC chip. The default setting ties the battery into the real-time clock (RTC) chip. If you lose time between power cycles or if the operating system boots with a very inaccurate time, check the J13 setting. If disabled, set it to enabled. If enabled, the battery should be changed. The battery is a 3V 190 milliamp coin cell battery, part number 12-41476-06. The RTC chip also stores some environment variable settings. If you set a bad environment variable such that you cannot bring up the system, you can disable J13. For example, if you forgot the password set for AlphaBIOS, set J13 to disabled so that you can access AlphaBIOS.

NOTE: *The operating systems use different algorithms for system time. If you switch between operating systems(for example, between UNIX and OpenVMS), be sure to reset the time at the operating system level.*

B.5 Setting Jumpers

Review the material in the previous sections of this chapter before setting any system jumpers. Before setting jumpers, shut down the system and remove the power cord from each power supply.

CAUTION: *Static electricity can damage integrated circuits. Always use a grounded wrist strap (29-26246) and grounded work surface when working with internal parts of a computer system.*

Remove jewelry before working on internal parts of the system.

Setting Jumpers

1. Shut down the operating system.
2. Shut down power on all external options connected to the system.
3. Turn off power to the system.
4. Unplug the power cord from each power supply.
5. Remove enclosure panels and chassis covers to gain access to the system motherboard or PCI board.
   - If you are setting RMC jumpers, remove CPU 1 to gain access to the jumpers.
   - If you are setting TIG/SROM jumpers, remove MMB 1 to gain access to the jumpers.
   - If you are setting PCI jumpers, you typically do not need to remove any PCI cards. However, if you have a full-length card in slot 10, remove it.
6. Locate the jumper you need to set. Refer to the illustrations in this chapter. Set the jumpers as needed.
7. Reinstall any modules you removed.
8. Reinstall the chassis covers and enclosure panels.

Plug the power cords into the supplies.

Appendix C
DPR Address Layout

This appendix shows the address layout of the dual-port RAM (DPR). Use the SRM **examine dpr:*address*** command (where *address* is the offset from the base of the DPR) or use the RMC **dump** command to view locations in the DPR. See Appendix D for definitions of locations written when environmental error events occur.

C.1 DPR Address Layout

Table C–1 DPR Address Layout

Location (Hex)	Logical Indicator	Written By	Used For
0	0	SROM	EV6 BIST status 1=good 0=bad
1	1	SROM	Bit[7]=Master Bits[0,1]=CPU_ID
2	2	SROM	Test STR status 1=good 0=bad
3	3	SROM	Test CSC status 1=good 0=bad
4	4	SROM	Test Pchip 0 PCTL status 1=good 0=bad
5	5	SROM	Test Pchip 1 PCTL status 1=good 0=bad
6	6	SROM	Test DIMx status 1=good 0=bad
7	7	SROM	Test TIG bus status
8	8	SROM	Dual-Port RAM test DD= started
9	9	SROM	Status of DPR test 1=good 0=bad
A	A	SROM	Status of CPU speed function FF=good 0=bad
B	B	SROM	Lower byte of CPU speed in MHz
C	C	SROM	Upper byte of CPU speed in MHz
D:F	-	-	Reserved
10:15		SROM	Power On Time Stamp for CPU 0—written as BCD Byte 10 = Hours (0-23) Byte 11 = Minutes (0-59) Byte 12 = Seconds (0-59) Byte 13 = Day of Month (1-31) Byte 14 = Month (1-12) Byte 15 = Year (0-99)

Table C–1 DPR Address Layout (Continued)

Location (Hex)	Logical Indicator	Written By	Used For	
16		SROM	SROM Power On Error Indication for CPU is "alive." For example; 0 = no error, 2 = Secondary time-out Error, 3 = Bcache Error	
17:1D			Unused	
1E		SROM	Last "sync state" reached; 80=Finished GOOD	
1F		SROM	Size of Bcache in MB	
20:3F	20		Repeat for CPU1 of CPU0 0-1F	
40:5F	20		Repeat for CPU2 of CPU0 0-1F	
60:7F	20		Repeat for CPU3 of CPU0 0-1F	
80	80	SROM	Array 0 (AAR 0) Configuration	
			Bits<7:4>	Bits<3:0>
			4 = non split - lower set only	0 = Configured - Lowest array
			5 = split - lower set only	1 = Configured - Next lowest array
			9 = split - upper set only	2 = Configured - Second highest array
			D = split - 8 DIMMs	3 = Configured - Highest array
			F = Twice split - 8 DIMMs	4 = Misconfigured - Missing DIMM(s)
				8 = Miconfigured - Illegal DIMM(s)
				C = Misconfigured - Incompatible DIMM(s)

Table C–1 DPR Address Layout (Continued)

Location (Hex)	Logical Indicator	Written By	Used For
81	81	SROM	Array 0 (AAR 0)Size (x64 Mbytes) 0 = no good memory 1 = 64 Mbyte 2 = 128 Mbyte 4 = 256 Mbyte 8 = 512 Mbyte 10 = 1 Gbyte 20 = 2 Gbyte 40 = 4 Gbyte 80 = 8 Gbyte
82	82	SROM	Array 1 (AAR 1) Configuration
83	83	SROM	Array 1 (AAR 1) Size (x64 Mbytes)
84	84	SROM	Array 2 (AAR 2) Configuration
85	85	SROM	Array 2 (AAR 2) Size (x64 Mbytes)
86	86	SROM	Array 3 (AAR 3) Configuration
87	87	SROM	Array 3 (AAR 3) Size (x64 Mbytes)
88:8B		SROM	Byte to define failed DIMMs for MMBs 88 - MMB 0 89 - MMB 1 8A - MMB 2 8B - MMB 3 Bit set indicates failure. Bit definitions (bit 0 = DIMM 1, bit 1 = DIMM2, bit 2 = DIMM 3, bit 7 = DIMM 8)
8C:8F	8C-8F	SROM	Byte to define misconfigured DIMMs for MMBs 8C – MMB 0 8D – MMB 1 8E – MMB 2 8F – MMB 3 Bit definitions (bit 0 = DIMM 1, bit 1 = DIMM2, bit 2 = DIMM 3, bit 7 = DIMM 8)
90	90	RMC	Power Supply/VTERM present
91	91	RMC	Power Supply PS_POK bits
92	92	RMC	AC input value from Power Supply

Table C–1 DPR Address Layout (Continued)

Location (Hex)	Logical Indicator	Written By	Used For
93:96	93	RMC	Temperature from CPU(x) in BCD
97:99	97	RMC	Temperature Zone(x) from 3 PCI temp sensors
9A:9F	9A	RMC	Fan Status; Raw Fan speed value
A0:A9	A0	RMC	Failure registers used as part of the 680 machine check logout frame. See Appendix D.
AA		RMC	Fan status (bit 0 = fan 1, bit 1 = fan 2, 1- indicates good; 0 indicates fan failure
AB		RMC	Status of RMC to read I^2C bus of MMB0 DIMMs Definition: Bit 7 - DIMM 8 0=OK 1=Fail Bit 6 - DIMM 7 Bit 5 - DIMM 6 .... Bit 0 - DIMM 1
AC		RMC	Status of RMC to read I^2C bus of MMB1 DIMMs
AD		RMC	Status of RMC to read I^2C bus of MMB2 DIMMs
AE		RMC	Status of RMC to read I^2C bus of MMB3 DIMMs
AF		RMC	Status of RMC to read MMB and CPU I^2C buses Definition: Bit 7 - MMB3 0=OK 1=Fail Bit 6 - MMB2 Bit 5 - MMB1 Bit 4 - MMB0 Bit 3 - CPU3 Bit 2 - CPU2 Bit 1 - CPU1 Bit 0 - CPU0
B0		RMC	Status of RMC to read CPB (PCI backplane) I^2C EEROM 0=OK 1 = fail
B1		RMC	Status of RMC to read CSB (motherboard) I^2C EEROM 0=OK 1 = fail

Table C–1 DPR Address Layout (Continued)

Location (Hex)	Logical Indicator	Written By	Used For
B2		RMC	Status of RMC to read SCSI backplane Definition: Bit 0 — SCSI backplane 0 Bit 1 — SCSI backplane 1 Bit 4 — Power supply 0 Bit 5 — Power supply 1 Bit 6 — Power supply 2
B3:B9		Unused	Unused
BA		RMC	I^2C done, BA = finished
BB		RMC	RMC Power on Error indicates error during power-up (1=Flash Corrupted)
BC		RMC	RMC flash update error status
BD		RMC	Copy of PS input Value. See Appendix D.
BE		RMC	Copy of the byte from the I/O expanders on the SPC loaded by the RMC on fatal errors. See Appendix D.
BF		RMC	Reason for system failure. See Appendix D.
C0:D8			Unused
D9		RMC	Baud rate
DA		TIG	Indicates TIG finished loading its code (0xAA indicates done)
DB:E3		RMC	Fan/Temp info from PS1
E4:EC		RMC	Fan/Temp info from PS2
ED:F5		RMC	Fan/Temp info from PS3
F6:F8		Unused	Unused
F9		Firmware	Buffer Size (0-0xFF) or 1 to 256 bytes
FA:FB	FA	Firmware	Command address qualifier FA = lower byte, FB = upper byte

Table C–1 DPR Address Layout (Continued)

Location (Hex)	Logical Indicator	Written By	Used For
FC	FC	RMC	Command status associated with the RMC response to a request from the firmware 0 = successful completion 80 = unsuccessful completion 81 = invalid command code 82 = invalid command qualifier
FD	FD	RMC	Command ID associated with the RMC response to a request from the firmware
FE	FE	Firmware	Command Code associated with a “command” sent to the RMC 1 = update I^2C EEROM 2 = update baud rate 3 = display to OCP F0 = update RMC flash
FF	FF	Firmware	Command ID associated with a “command” sent to the RMC
100:1FF	100	RMC	Copy of EEROM on MMB0 J1 DIMM 1, initially read on I^2C bus by RMC when 5 volts supply turned on. Written by Compaq Analyze after error diagnosed to particular FRU
200:2FF	200	RMC	Copy of EEROM on MMB0 J2 DIMM 2
300:3FF	300	RMC	Copy of EEROM on MMB0 J3 DIMM 3
400:4FF	400	RMC	Copy of EEROM on MMB0 J4 DIMM 4
500:5FF	500	RMC	Copy of EEROM on MMB0 J5 DIMM 5
600:7FF	600	RMC	Copy of EEROM on MMB0 J6 DIMM 6
700:7FF	700	RMC	Copy of EEROM on MMB0 J7 DIMM 7
800:8FF	800	RMC	Copy of EEROM on MMB0 J8 DIMM 8
900:9FF	900	RMC	Copy of EEROM on MMB1 J1 DIMM 1
A00:AFF	A00	RMC	Copy of EEROM on MMB1 J2 DIMM 2
B00:BFF	B00	RMC	Copy of EEROM on MMB1 J3 DIMM 3
C00:CFF	C00	RMC	Copy of EEROM on MMB1 J4 DIMM 4
D00:DFF	D00	RMC	Copy of EEROM on MMB1 J5 DIMM 5
E00:EFF	E00	RMC	Copy of EEROM on MMB1 J6 DIMM 6
F00:FFF	F00	RMC	Copy of EEROM on MMB1 J7 DIMM 7

Table C–1 DPR Address Layout (Continued)

Location (Hex)	Logical Indicator	Written By	Used For
1000:10FF	1000	RMC	Copy of EEROM on MMB1 J8 DIMM 8
1100:11FF	1100	RMC	Copy of EEROM on MMB2 J1 DIMM 1
1200:12FF	1200	RMC	Copy of EEROM on MMB2 J2 DIMM 2
1300:13FF	1300	RMC	Copy of EEROM on MMB2 J3 DIMM 3
1400:14FF	1400	RMC	Copy of EEROM on MMB2 J4 DIMM 4
1500:15FF	1500	RMC	Copy of EEROM on MMB2 J5 DIMM 5
1600:16FF	1600	RMC	Copy of EEROM on MMB2 J6 DIMM 6
1700:17FF	1700	RMC	Copy of EEROM on MMB2 J7 DIMM 7
1800:18FF	1800	RMC	Copy of EEROM on MMB2 J8 DIMM 8
1900:19FF	1900	RMC	Copy of EEROM on MMB3 J1 DIMM 1
1A00:1AFF	1A00	RMC	Copy of EEROM on MMB3 J2 DIMM 2
1B00:1BFF	1B00	RMC	Copy of EEROM on MMB3 J3 DIMM 3
1C00:1CFF	1C00	RMC	Copy of EEROM on MMB3 J4 DIMM 4
1D00:1DFF	1D00	RMC	Copy of EEROM on MMB3 J5 DIMM 5
1E00:1EFF	1E00	RMC	Copy of EEROM on MMB3 J6 DIMM 6
1F00:1FFF	1F00	RMC	Copy of EEROM on MMB3 J7 DIMM 7
2000:20FF	2000	RMC	Copy of EEROM on MMB3 J8 DIMM 8
2100:21FF	2100	RMC	Copy of EEROM from CPU0
2200:22FF	2200	RMC	Copy of EEROM from CPU1
2300:23FF	2300	RMC	Copy of EEROM from CPU2
2400:24FF	2400	RMC	Copy of EEROM from CPU3
2500:25FF	2500	RMC	Copy of MMB 0 J5 FRU EEROM
2600:26FF	2600	RMC	Copy of MMB 1 J7 FRU EEROM
2700:27FF	2700	RMC	Copy of MMB 2 J6 FRU EEROM
2800:28FF	2800	RMC	Copy of MMB 3 J8 FRU EEROM
2900:29FF	2900	RMC	Copy of EEROM on CPB (PCI backplane)
2A00:2AFF	2A00	RMC	Copy of EEROM on CSB (motherboard)
2B00:2BFF	2B00	RMC	Last EV6 Correctable Error—ASCII character string that indicates correctable error occurred, type, FRU, and so on. Backed up in CSB (motherboard) EEROM. Written by Compaq Analyze

Table C–1 DPR Address Layout (Continued)

Location (Hex)	Logical Indicator	Written By	Used For
2C00:2CFF	2C00	RMC	Last Redundant Failure—ASCII character string that indicates redundant failure occurred, type, FRU, and so on. Backed up in system CSB (motherboard) EEROM. Written by Compaq Analyze
2D00:2DFF	2D00	RMC	Last System Failure—ASCII character string that indicates system failure occurred, type, FRU, and so on. Backed up in CSB (motherboard) EEROM. Written by Compaq Analyze.
2E00:2FFF	2E00	RMC	Uncorrectable machine logout frame (512 bytes)
3000:3008		SROM	SROM Version (ASCII string)
3009:300B		RMC	Rev Level of RMC first byte is letter Rev [x/t/v] second 2 bytes are major/minor. This is the rev level of the RMC on-chip code.
300C:300E		RMC	Rev Level of RMC first byte is letter Rev [x/t/v] second 2 bytes are major/minor. This is the rev level of the RMC flash code.
300F:3010	300F	RMC	Revision Field of the DPR Structure
3011:30FF		Unused	Unused
3100:31FF		RMC	Copy of PS0 EEROM (first 256 bytes)
3200:32FF		RMC	Copy of PS1 EEROM (first 256 bytes)
3300:33FF		RMC	Copy of PS2 EEROM (first 256 bytes)
3400		SROM	Size of Bcache in MB
3401		SROM	Flash SROM is valid flag; 8 = valid, 0 = invalid
3402		SROM	System's errors determined by SROM
3403:340F		SROM/SRM	Reserved for future SROM/SRM communication
3410:3417		SROM/SRM	Jump to address for CPU0

Table C–1 DPR Address Layout (Continued)

Location (Hex)	Logical Indicator	Written By	Used For
3418		SROM/SRM	Waiting to jump to flag for CPU0
3419		SROM	Shadow of value written to EV6 DC_CTL register.
341A:341E		SROM	Shadow of most recent writes to EV6 CBOX “Write-many” chain.
341F		SROM/SRM	Reserved for future SROM/SRM communication
3420:342F		SROM/SRM	Repeat for CPU1 of CPU0 3410-341F
3430:343F		SROM/SRM	Repeat for CPU2 of CPU0 3410-341F
3440:344F		SROM/SRM	Repeat for CPU3 of CPU0 3410-341F
3450:349F		SROM/ RMC	Reserved for SROM mini-console via RMC communication area. Future design.
34A0:34A7		SROM	Array 0 to DIMM ID translation Bits<7:5>: 0 = Exists, No Error; 1 = Expected Missing; 2 = Error - Missing DIMM(s); 4 = Error - Illegal DIMM(s); 6 = Error - Incompatible DIMM(s) Bits<4:0>: Bits <2:0> = DIMM + 1 (1-8); Bits <4:3> = MMB (0-3)
34A8:34AF		SROM	Repeat for Array 1 of Array 0 34A0:34A7
34B0:34B7		SROM	Repeat for Array 2 of Array 0 34A0:34A7
34B8:34CF		SROM	Repeat for Array 3 of Array 0 34A0:34A7
34C0:34FF	34C0	SROM	Used as scratch area for SROM

Table C–1 DPR Address Layout (Continued)

Location (Hex)	Logical Indicator	Written By	Used For
3500:35FF		Firmware	Used as the dedicated buffer in which SRM writes OCP or FRU EEROM data. Firmware will write this data, RMC will only read this data.
3600:36FF	3600	SRM	Reserved
3700:37FF		SRM	Reserved
3800:3AFF		RMC	RMC scratch space
3B00:3BFF		RMC	First SCSI backplane EEROM
3C00:3CFF		RMC	Second SCSI backplane EEROM
3D00:3DFF		RMC	PS0 second 256 bytes
3E00:3EFF		RMC	PS1 second 256 bytes
3F00:3FFF		RMC	PS2 second 256 bytes

Appendix D
Registers

This appendix describes 21264 (EV6) internal processor registers; 21272 (Tsunami/Typhoon) system support chipset registers; and dual-port RAM (DPR) registers that are related to general logout frame errors. It also provides CPU and system uncorrectable and correctable machine logout frames and error state bit definitions of all the platform logout frame registers.

21264 (EV6) Registers

Ibox Status Register (I_STAT)
Memory Management Status Register (MM_STAT)
Dcache Status Register (DC_STAT)
Cbox Read Register
Exception Address Register (EXC_ADDR)
Interrupt Enable and Current Processor Mode Register (IER_CM)
Interrupt Summary Register (ISUM)
PAL Base Register (PAL_BASE)
Ibox Control Register (I_CTL)
Process Context Register (PCTX)

21272 (Tsunami/Typhoon) System Registers

21272-CA Cchip Miscellaneous Register (MISC)
21272-CA Device Interrupt Request Register (DIR*n*, *n*=0,1,2,3)
21272-CA Pchip Error Register (PERROR)
21272-CA Array Address Registers

DPR Registers

DPR Registers (for 680 correctable error state capture)
DPR Registers (for I^2C bus)
DPR Registers (power supply status from I^2C bus)
DPR 680 Fatal Registers (for 680 uncorrectable error state capture)

D.1 Ibox Status Register (I_STAT)

The Ibox Status Register (I_STAT) is read only by PAL code and is an element in the CPU or system uncorrectable and correctable machine check error logout frame.

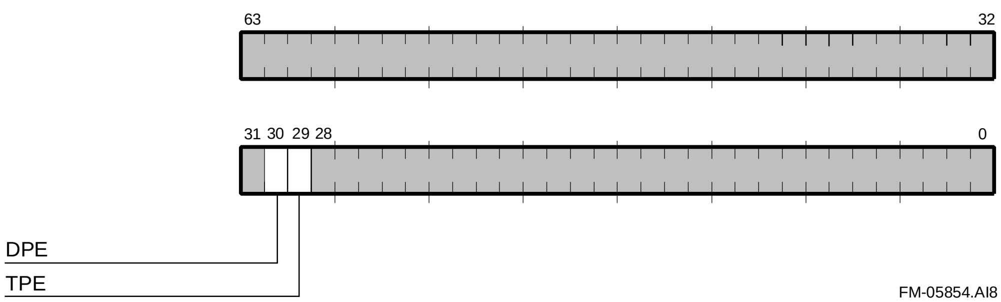


Table D–1 Ibox Status Register Fields

Name	Bits	Type	Description
Reserved	<63:31>	RO	Reserved for Compaq.
DPE	<30>	W1C	I-cache data parity error When set, indicates that the I-cache encountered a data parity error on instruction fetch.
TPE	<29>	W1C	I-cache tag parity error When set, indicates that the I-cache encountered a tag parity error on instruction fetch.
Reserved	<28:0>	RO	Reserved for Compaq.

D.2 Memory Management Status Register (MM_STAT)

The Memory Management Status Register (MM_STAT) is read only by PAL code and is an element in the CPU or system uncorrectable and correctable machine check error logout frame.

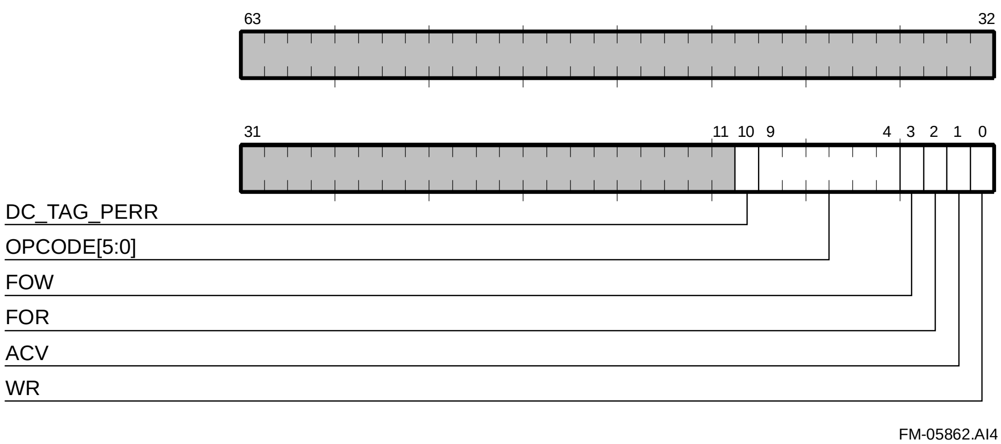


Table D–2 Memory Management Status Register Fields

Name	Bits	Type	Description
Reserved	<63:11>		Reserved for Compaq.
DC_TAG_PERR	<10>	RO	This bit is set when a D-cache tag parity error occurs during the initial tag probe of a load or store instruction. The error created a synchronous fault to the D_FAULT PALcode entry point and is correctable. The virtual address associated with the error is available in the VA register.
OPCODE	<9:4>	RO	Opcode of the instruction that caused the error. HW_LD is displayed as 3 and HW_ST is displayed as 7.
FOW	<3>	RO	Set when a fault-on-write error occurs during a write transaction and PTE[FOW] was set.
FOR	<2>	RO	Set when a fault-on-read error occurs during a read transaction and PTE[FOR] was set.
ACV	<1>	RO	Set when an access violation occurs during a transaction. Access violations include a bad virtual address.
WR	<0>	RO	Set when an error occurs during a write transaction.

D.3 Dcache Status Register (DC_STAT)

The Dcache Status Register (DC_STAT) is read only by PAL code and is an element in the CPU or system uncorrectable and correctable machine check error logout frame.

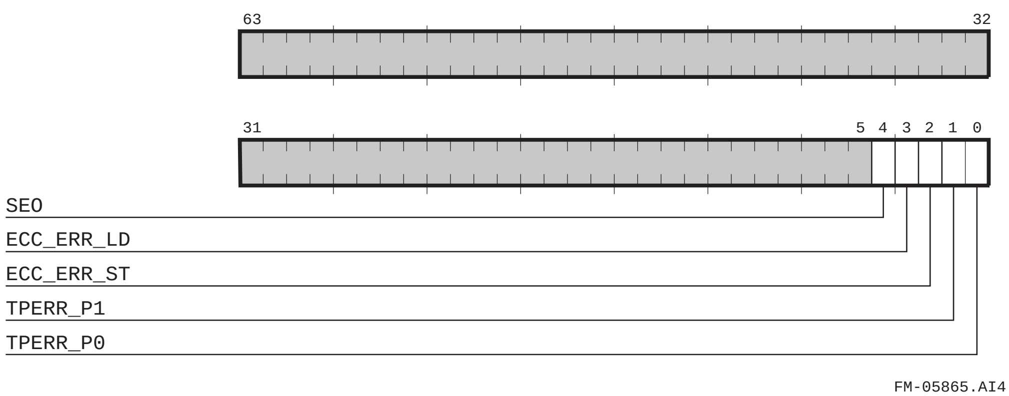


Table D–3 Dcache Status Register Fields

Name	Bits	Type	Description
Reserved	<63:5>		Reserved for Compaq.
SEO	<4>	W1C	Second error occurred. When set, indicates that a second D-cache store ECC error occurred within 6 cycles of the previous D-cache store ECC error.
ECC_ERR_LD	<3>	W1C	ECC error on load. When set, indicates that a single-bit ECC error occurred while processing a load from the D-cache or any fill.
ECC_ERR_ST	<2>	W1C	ECC error on store. When set, indicates that an ECC error occurred while processing a store.
TPERR_P1	<1>	W1C	Tag parity error— pipe 1. When set, indicates that a D-cache tag probe from pipe 1 resulted in a tag parity error. The error is uncorrectable and results in a machine check.
TPERR_P0	<0>	W1C	Tag parity error— pipe 0. When set, this bit indicates that a D-cache tag probe from pipe 1 resulted in a tag parity error. The error is uncorrectable and results in a machine check.

D.4 Cbox Read Register

The Cbox Read Register is read only by PAL code and is an element in the CPU or system uncorrectable and correctable machine check error logout frame.

Table D–4 Cbox Read Register Fields

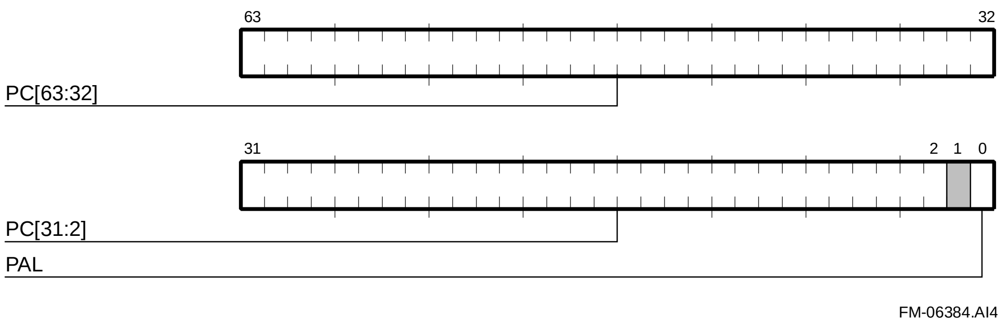
Name	Description	
C_SYNDROME_1<7:0>	Syndrome for the upper QW in the OW of victim that was scrubbed. See Appendix E.	
C_SYNDROME_0<7:0>	Syndrome for the lower QW in the OW of victim that was scrubbed. See Appendix E.	
C_STAT<4:0>	**Bits**	**Error Status**
	00000	Either no error, or error on a speculative load, of a B-cache victim read due to a D-cache/B-cache miss.
	00001	BC_PERR (B-cache tag parity error)
	00010	DC_PERR (duplicate tag parity error)
	00011	DSTREAM_MEM_ERR
	00100	DSTREAM_BC_ERR
	00101	DSTREAM_DC_ERR
	0011X	PROBE_BC_ERR
	01000	Reserved
	01001	Reserved
	01010	Reserved
	01011	ISTREAM_MEM_ERR

Table D–4 Cbox Read Register Fields (Continued)

Name	Description	
C_STAT<4:0> (continued)	**Bits**	**Error Status**
	01100	ISTREAM_BC_ERR
	01101	Reserved
	0111X	Reserved
	10011	DSTREAM_MEM_DBL
	10100	DSTREAM_BC_DBL
	11011	ISTREAM_MEM_DBL
	11100	ISTREAM_BC_DBL
C_STS<3:0>	If C_STAT equals *xxx*_MEM_ERR or *xxx*_BC_ERR, then C_STAT contains the status of the block as follows; otherwise, the value of C_STAT is X.	
	Bit Value	**Status of Block**
	7–4	Reserved
	3	Parity
	2	Valid
	1	Dirty
	0	Shared
C_ADDR<6:42>	Address of the last reported ECC or parity error. If C_STAT value is DSTREAM_DC_ERR, only bits <6:19> are valid.	

D.5 Exception Address Register (EXC_ADDR)

The exception address register (EXC_ADDR) is a read-only register that is updated by hardware when it encounters an exception or interrupt.



EXC_ADDR[0] is set if the associated exception occurred in PAL mode. The exception actions are:

- If the exception was a fault or a synchronous trap, EXC_ADDR contains the PC of the instruction that triggered the fault or trap.
- If the exception was an interrupt, EXC_ADDR contains the PC of the next instruction that would have executed if the interrupt had not occurred.

D.6 Interrupt Enable and Current Processor Mode Register (IER_CM)

The interrupt enable and current processor mode register (IER_CM) contains the interrupt enable and current processor mode bit fields.

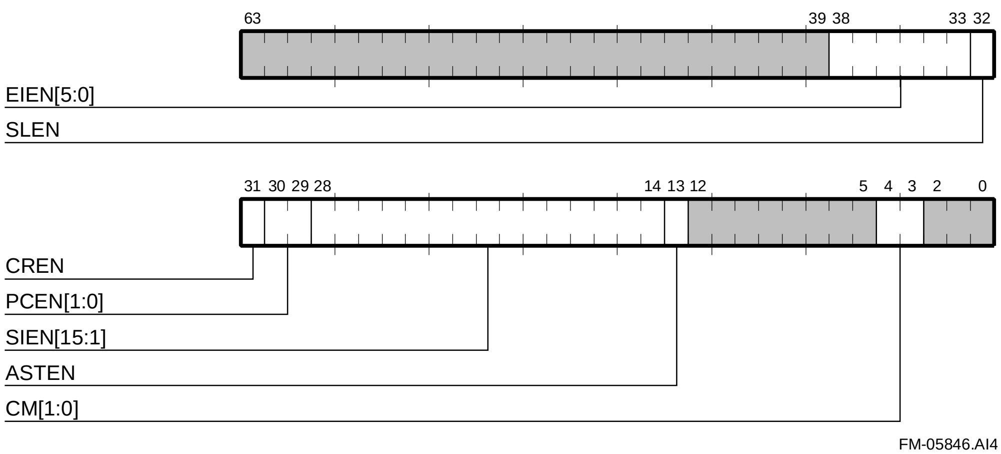


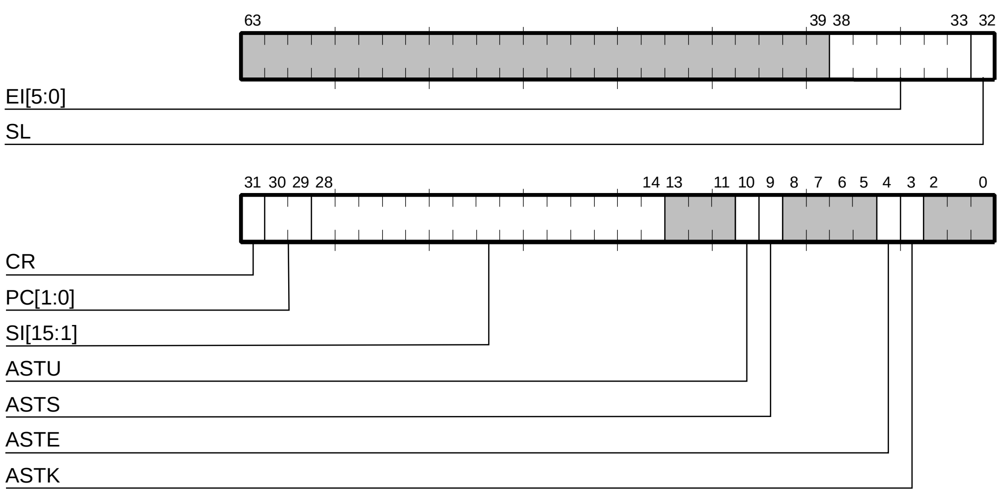
Table D–5 IER_CM Register Fields

Name	Extent	Type	Description
Reserved	[63:39]		
EIEN[5:0]	[38:33]	RW	External Interrupt Enable
SLEN	[32]	RW	Serial Line Interrupt Enable
CREN	[31]	RW	Corrected Read Error Interrupt Enable
PCEN[1:0]	[30:29]	RW	Performance Counter Interrupt Enables
SIEN[15:1]	[28:14]	RW	Software Interrupt Enables
ASTEN	[13]	RW	AST Interrupt Enable
			When set, enables those AST interrupt requests that are also enabled by the value in ASTER.
Reserved	[12:5]		
CM[1:0]	[4:3]	RW	Current Mode
			00 Kernel
			01 Executive
			10 Supervisor
			11 User
Reserved	[2:0]		

D.7 Interrupt Summary Register (ISUM)

The interrupt summary register (ISUM) is a read-only register that records all pending hardware, software, and AST interrupt requests that have their corresponding enable bit set.

If a new interrupt (hardware, serial line, crd, or performance counters) occurs simultaneously with an ISUM read, the ISUM read returns zeros. That condition is normally assumed to be a passive release condition. The interrupt is signaled again when the PALcode returns to native mode. The effects of this condition can be minimized by reading ISUM twice and ORing the results.



FM-05849.AI4

Table D–6 ISUM Register Fields

Name	Extent	Type	Description
Reserved	[63:39]		
EI[5:0]	[38:33]	RO	External Interrupts
SL	[32]	RO	Serial Line Interrupt
CR	[31]	RO	Corrected Read Error Interrupts
PC[1:0]	[30:29]	RO	Performance Counter Interrupts PC0 when PC[0] is set. PC1 when PC[1] is set.
SI[15:1]	[28:14]	RO	Software Interrupts
Reserved	[13:11]		
ASTU, ASTS	[10],[9]	RO	AST Interrupts For each processor mode, the bit is set if an associated AST interrupt is pending. This includes the mode's ASTER and ASTRR bits and whether the processor mode value held in the IER_CM register is greater than or equal to the value for the mode.
Reserved	[8:5]		
ASTE, ASTK	[4],[3]	RO	AST Interrupts For each processor mode, the bit is set if an associated AST interrupt is pending. This includes the mode's ASTER and ASTRR bits and whether the processor mode value held in the IER_CM register is greater than or equal to the value for the mode.
Reserved	[2:0]		

D.8 PAL Base Register (PAL_BASE)

The PAL base register (PAL_BASE) is a read-write register that contains the base physical address for PALcode. Its contents are cleared by chip reset but are not cleared after waking up from sleep mode or from fault reset.

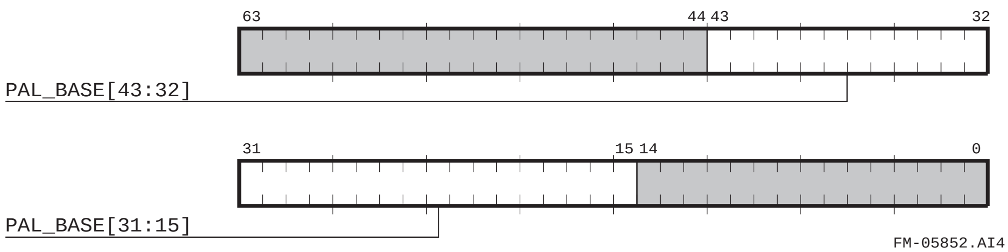


Table D–7 PAL_BASE Register Fields

Name	Extent	Type	Description
Reserved	[63:44]	RO, 0	Reserved for COMPAQ.
PAL_BASE[43:15]	[43:15]	RW	Base physical address for PALcode.
Reserved	[14:0]	RO, 0	Reserved for COMPAQ.

D.9 Ibox Control Register (I_CTL)

The Ibox control register (I_CTL) is a read-write register that controls various Ibox functions. Its contents are cleared by chip reset.

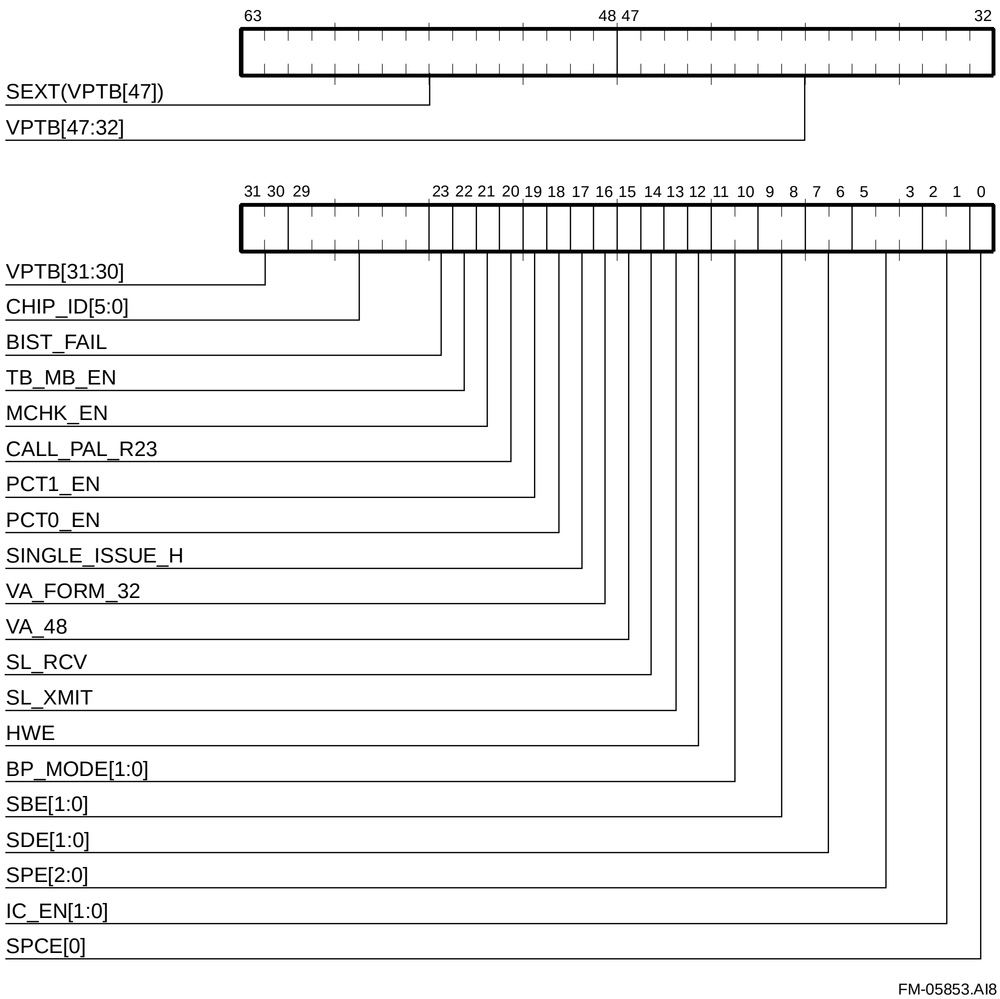


Table D–8 I_CTL Register Fields

Name	Extent	Type	Description
SEXT(VPTB[47])	[63:48]	RW,0	Sign extended VPTB[47].
VPTB[47:30]	[47:30]	RW,0	Virtual Page Table Base.
CHIP_ID[5:0]	[29:24]	RO	This is a read-only field that supplies the revision ID number for the 21264 part. 21264 pass 1 ID is 000000_2. 21264 pass 2 ID is 000001_2 21264 pass 2.2 ID is 000010_2. 21264 pass 2.3 ID is 000011_2 21264 pass 2.4 ID is 000101.
BIST_FAIL	[23]	RO,0	Indicates the status of BIST (set = pass, clear = fail).
TB_MB_EN	[22]	RW,0	When set, the hardware ensures that the virtual-mode loads in DTB and ITB fill flows that access the page table and the subsequent virtual mode load or store that is being retried are ‘ordered’ relative to another processor’s stores. This must be set for multiprocessor systems in which no MB instruction is present in the TB fill flow, unless there are other mechanisms present that ensure coherency.
MCHK_EN	[21]	RW,0	Machine check enable — set to enable machine checks.
CALL_PAL_R23	[20]	RW,0	CALL_PAL linkage register. If this bit is one, the CALL_PAL linkage register is R23; when zero, it is R27. Coordinate setting this bit with SDE[1:0] to ensure that the shadow register is used as the linkage register.
PCT1_EN	[19]	RW,0	Enable performance counter #1. If this bit is one, the performance counter will count if either the system (SPCE) or process (PPCE) performance counter enable is asserted.

Continued on next page

Table D–8 I_CTL Register Fields (Continued)

Name	Extent	Type	Description
PCT0_EN	[18]	RW,0	Enable performance counter #0. If this bit is one, the performance counter will count if EITHER the system (SPCE) or process (PPCE) performance counter enable is set.
SINGLE_ISSUE_H	[17]	RW,0	When set, this bit forces instructions to issue only from the bottom-most entries of the IQ and FQ.
VA_FORM_32	[16]	RW,0	This bit controls address formatting on a read of the IVA_FORM register.
VA_48	[15]	RW,0	This bit controls the format applied to effective virtual addresses by the IVA_FORM register and the Ibox virtual address sign extension checkers. When VA_48 is clear, 43-bit virtual address format is used, and when VA_48 is set, 48-bit virtual address format is used. The effect of this bit on the IVA_FORM register is identical to the effect of VA_CTL[VA_48] on the VA_FORM register. When VA_48 is set, the sign extension checkers generate an ACV if va[63:0] ≠ SEXT(va[47:0]). When VA_48 is clear, the sign extension checkers generate an ACV if va[63:0] ≠ SEXT(va[42:0]). This bit also affects DTB_DOUBLE Traps. If set, the DTB double miss traps vector to the DTB_DOUBLE_4 entry point. DTB_DOUBLE PALcode flow selection is not affected by VA_CTL[VA_48].
SL_RCV	[14]	RO	When in native mode, any transition on SL_RCV, driven from the SromData_H pin, results in a trap to the PALcode interrupt handler. When in PALmode, all interrupts are blocked. The interrupt routine then begins sampling SL_RCV under a software timing loop to input as much data as needed, using the chosen serial line protocol.

Table D–8 I_CTL Register Fields (Continued)

Name	Extent	Type	Description
SL_XMIT	[13]	WO	When set, drives a value on **SromClk_H**.
HWE	[12]	RW,0	If set, allow PALRES intructions to be executed in kernel mode. Note that modification of the ITB while in kernel mode/native mode may cause UNPREDICTABLE behavior.
BP_MODE[1:0]	[11:10]	RW,0	Branch Prediction Mode Selection. BP_MODE[1], if set, forces all branches to be predicted to fall through. If clear, the dynamic branch predictor is chosen. BP_MODE[0]. If set, the dynamic branch predictor chooses local history prediction. If clear, the dynamic branch predictor chooses local or global prediction based on the state of the chooser.
SBE[1:0]	[9:8]	RW,0	Stream Buffer Enable. The value in this bit field specifies the number of Istream buffer prefetches (besides the demand-fill) that are launched after an Icache miss. If the value is zero, only demand requests are launched.
SDE[1:0]	[7:6]	RW,0	PALshadow Register Enable. Enables access to the PALshadow registers. If SDE[1] is set, R4-R7 and R20-R23 are used as PALshadow registers. SDE[0] does not affect 21264 operation.

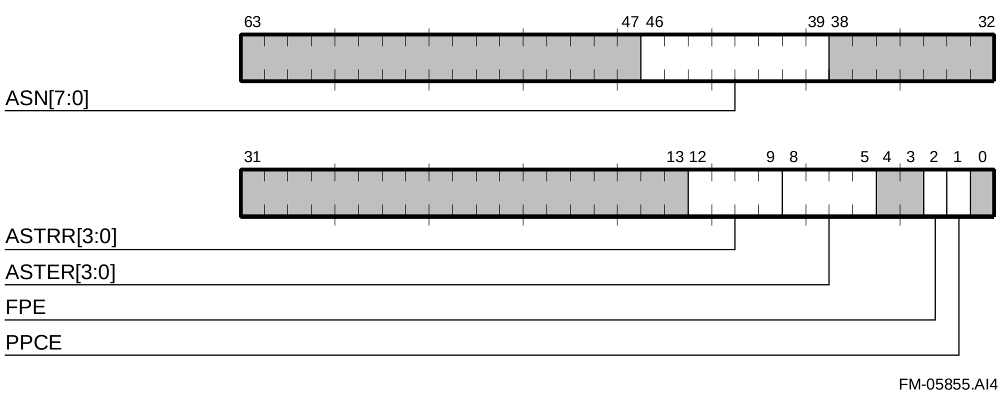
Continued on next page

Table D–8 I_CTL Register Fields (Continued)

Name	Extent	Type	Description
SPE[2:0]	[5:3]	RW,0	Super Page Mode Enable. Identical to the SPE bits in the Mbox M_CTL SPE[2:0].
IC_EN[1:0]	[2:1]	RW,3	Icache Set Enable. At least one set must be enabled. The entire cache may be enabled by setting both bits. Zero, one, or two Icache sets can be enabled. This bit does not clear the Icache, but only disables fills to the affected set.
SPCE	[0]	RW,0	System Performance Counting Enable. Enables performance counting for the entire system if individual counters (PCTR0 or PCTR1) are enabled by setting PCT0_EN or PCT1_EN, respectively. Performance counting for individual processes can be enabled by setting PCTX[PPCE].

D.10 Process Context Register (PCTX)

The process context register (PCTX) contains information associated with the context of a process.



Continued on next page

The following table lists the correspondence between IPR index bits and register fields.

IPR Index Bit	Register Field
0	ASN
1	ASTER
2	ASTRR
3	PPCE
4	FPE

Table D–9 lists the PXTX register fields.

Table D–9 PCTX Register Fields

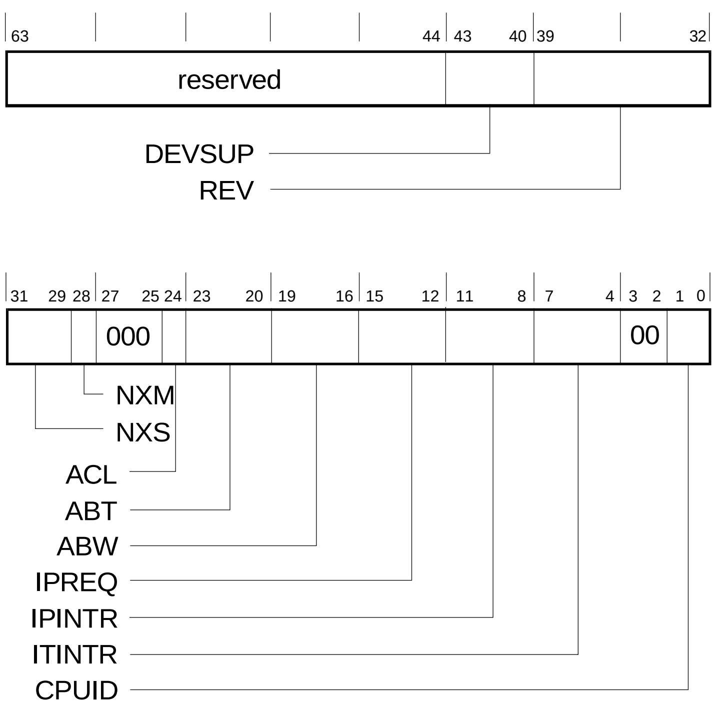
Name	Extent	Type	Description
Reserved	[63:47]		
ASN[7:0]	[46:39]	RW	Address space number.
Reserved	[38:13]		
ASTRR[3:0]	[12:9]	RW	AST request register—used to request AST interrupts in each of the four processor modes. To generate a particular AST interrupt, its corresponding bits in ASTRR and ASTER must be set, along with the ASTE bit in IER. Further, the value of the current mode bits in the PS register must be equal to or higher than the value of the mode associated with the AST request. The bit order with this field is: User Mode Supervior Mode Executive Mode Kernel Mode
ASTER[3:0]	[8:5]	RW	AST enable register—used to individually enable each of the four AST interrupt requests. The bit order with this field is: User Mode Supervisor Mode Executive Mode Kernel Mode
Reserved	[4:3]		
FPE	[2]	RW,1	Floating-point enable—if clear, floating-point instructions generate FEN exceptions. This bit is set by hardware on reset.
PPCE	[1]	RW	Process performance counting enable. Enables performance counting for an individual process with counters PCTR0 or PCTR1, which are enabled by setting PCT0_EN or PCT1_EN, respectively. Performance counting for the entire system can be enabled by setting I_CTL[SPCE].

D.11 21272-CA Cchip Miscellaneous Register (MISC)

This register is designed so that only writes of 1 affect it. When a 1 is written to any bit in the register, the programmer does not need to be concerned with read-modify-write or the status of any other bits in the register. Once NXM is set, the NXS field is locked. It is unlocked when software clears the NXM field. The ABW (arbitration won) field is locked if either ABW bit is set, so the first CPU to write it locks out the other CPU. Writing a 1 to ACL (arbitration clear) clears both ABW bits and both ABT (arbitration try) bits and unlocks the ABW field.

Address 801 A000 0040

Access RW



PK1417-99

Table D–10 21272-CA Cchip Miscellaneous Register Fields

Name	Bits	Type	Initial State	Description
RES	<63:44>	MBZ, RAZ	0	Reserved.
DEVSUP	<43:40>	WO	0	
REV	<39:32>	RO	1	Latest revision of the Cchip: 1 = Tsunami 8=Typhoon
NXS	<31:29>	RO	0	NXM source—Device that caused the NXM. Unpredictable if NXM not set. 0 = CPU0 1 = CPU1 2 = CPU2 3 = CPU3 4 = P-chip 0 5 = P-chip 1
NXM	<28>	R, W1C	0	Nonexistent memory address detected. Sets DRIR<63> and locks the NXS field until it is cleared.
RES	<27:25>	MBZ, RAZ	0	Reserved.
ACL	<24>	WO	0	Arbitration clear—writing a 1 to this bit clears the ABT and ABW fields.
ABT	<23:20>	R, W1S	0	Arbitration try—writing a 1 to these bits sets them.
ABW	<19:16>	R, W1S	0	Arbitration won—writing a 1 to these bits sets them unless one is already set, in which case the write is ignored.
IPREQ	<15:12>	WO	0	Interprocessor interrupt request—write a 1 to the bit corresponding to the CPU you want to interrupt. Writing a 1 here sets the corresponding bit in the IPINTR.

Continued on next page

Table D–10 21272-CA Cchip Miscellaneous Register Fields (Continued)

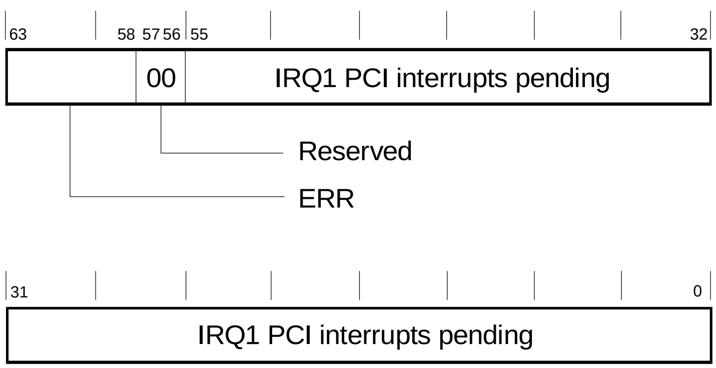
Name	Bits	Type	Initial State	Description
IPINTR	<11:8>	R, W1C	0	Interprocessor interrupt pending—one bit per CPU. Pin irq<3> is asserted to the CPU corresponding to a 1 in this field.
ITINTR	<7:4>	R, W1C	0	Interval timer interrupt pending—one bit per CPU. Pin irq<2> is asserted to the CPU corresponding to a 1 in this field.
RES	<3:2>	MBZ, RAZ	0	Reserved.
CPUID	<1:0>	RO	-	ID of the CPU performing the read.

D.12 21272-CA Cchip CPU Device Interrupt Request Register (DIRn, n=0,1,2,3)

These registers indicate which interrupts are pending to the CPUs and indicate the presence of an I/O error condition.

Address 801 A000 0280 CPU0
801 A000 02C0 CPU1
801 A000 0680 CPU2
801 A000 06C0 CPU3

Access RO



PK1418-99

Continued on next page

Table D–11 21272-CA Device Interrupt Request Register Fields

Name	Bits	Type	Initial State	Description
ERR	<63:58>	RO	0	IRQ0 error interrupts <63> Cchip detected MISC <NXM> <62> Recommended hookup to Pchip0 error <61> Recommended hookup to Pchip1 error
RES	<57:56>	RO	0	Reserved
NXS	<55:0>	RO	0	IRQ1 PCI interrupts pending to the CPU

D.13 21272-CA Pchip Error Register (PERROR)

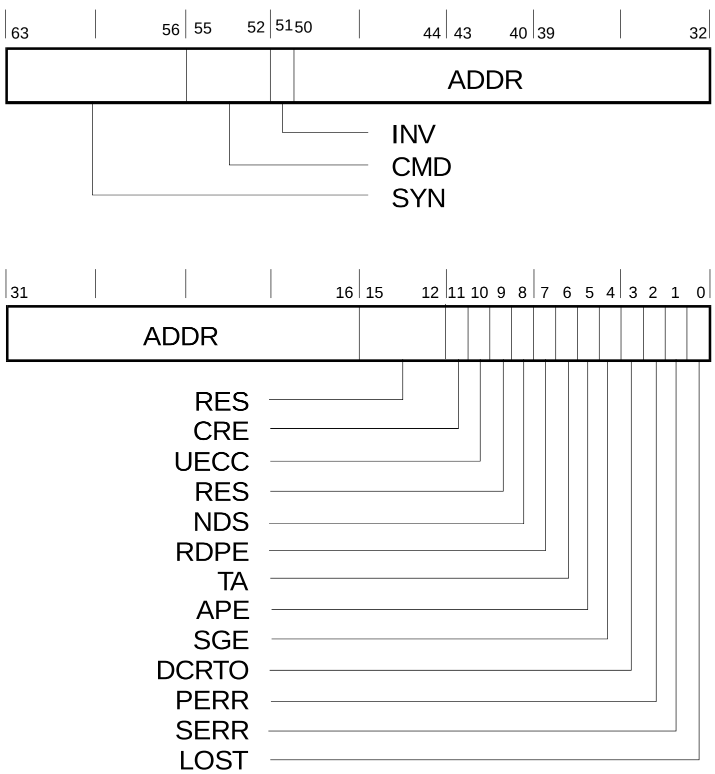
If any bits <11:0> are set, this register is frozen. Only bit <0> can be set thereafter. All other values are held until all bits <11:0> are clear. When an error occurs and one of the <11:0> bits is set, the associated information is captured in bit <63:16>. After the information is captured, the INV bit is cleared, but the information is not valid and should not be used if INV is set.

Address 801 8000 03C0 P0 ERROR

803 8000 03C0 P1 ERROR

Continued on next page

Access RW



PK1419-99

Table D–12 21272-CA Pchip Error Register Fields

Name	Bits	Type	Initial State	Description	
SYN	<63:56>	RO	0	ECC syndrome of error if CRE or UECC.	
CMD	<55:52>	RO	0	PCI command of transaction when error detected if not CRE and not UECC. If CRE or UECC, then:	
				Value	**Command**
				0000	DMA read
				0001	DMA read-modify-write
				0011	SGTE read
				Others	Reserved
INV	<51>	RO Rev1 RAZ Rev0	0	Info Not Valid—only meaningful when one of bits <11:0> is set. Indicates the validity of <SYN>, <CMD>, and <ADDR> fields.	
				Value	**Mode**
				0	Info fields are valid.
				1	Info fields are not valid.
ADDR	<50:16>	RO	0	If CRE or UECC, then ADDR<50:19> = system address <34:3> of erroneous quadword and ADDR<18:16> = 0. If not CRE and not UECC, then ADDR<50:48> = 0; ADDR<47:18> = starting PCI address <31:2> of transaction when error was detected; ADDR<17:16> = 00 → not a DAC operation; ADDR<17:16> = 01 → via DAC SG Window 3; ADDR<17> = 1 → via Monster Window	

Continued on next page

Table D–12 21272-CA Pchip Error Register Fields (Continued)

Name	Bits	Type	Initial State	Description
RES	<15:12>	MBZ, RAZ	0	Reserved
CRE	<11>	R, WIC	0	Correctable ECC error.
UECC	<10>	R, WIC	0	Uncorrectable ECC error.
RES	<9>	MBZ, RAZ	0	Reserved.
NDS	<8>	R, WIC	0	No **b_devsel_l** as PCI master.
RDPE	<7>	R,W1C	0	PCI read data parity error as PCI master.
TA	<6>	R, W1C	0	Target abort as PCI master.
APE	<5>	R, W1C	0	Address parity error detected as potential PCI target.
SGE	<4>	R, W1C	0	Scatter-gather had invalid page table entry.
DCRTO	<3>	R, W1C	0	Delayed completion retry timeout as PCI target.
PERR	<2>	R, W1C	0	**b_perr_l** sampled asserted.
SERR	<1>	R, W1C	0	**b_serr_l** sampled asserted.
LOST	<0>	R, W1C	0	Lost an error because it was detected after this register was frozen or while in the process of clearing this register.

D.14 21272-CA Array Address Registers (AAR0–AAR3)

The Array Address Registers define the base address and size for each memory array.

Table D–13 21272-CA Array Address Register (AAR)

Field	Bits	Type	Init	Description	
RES	<63:35>	MBZ,RAZ	0	Reserved.	
ADDR	<34:24>	RW	0	Base address – Bits <34:24> of the physical byte address of the first byte in the array. (<34:32> are used in Typhoon only; <34:28> are valid)	
RES	<23:17>	MBZ,RAZ	0	Reserved.	
DBG	16	RW	0	Enables this memory port to be used as a debug interface.	
ASIZ	<15:12>	RW	0	Array size (<15> is used in Typhoon only).	
				Value	**Size**
				0000	0 (bank disabled)
				0001	16MB
				0010	32MB
				0011	64MB
				0100	128MB
				0101	256MB
				0110	512MB
				0111	1GB
				1000	2GB (Typhoon only)
				1001	4GB (Typhoon only)
				1010	8GB (Typhoon only)
				1011 1111	Reserved.
RES	<11:10>	MBZ,RAZ	0	Reserved.	
TSA	<9>	RW	0	Twice-split array (Typhoon only)	
SA	<8>	RW	0	Split array.	

Continued on next page

Table D–13 21272-CA Array Address Register (AAR) (Continued)

Field	Bits	Type	Init	Description	
RES	<7:4>	MBZ,RAZ	0	Reserved.	
ROWS	<3:2>	RW	0	Number of row bits in the SDRAMs.	
				Value	**Number of Bits**
				0	11
				1	12
				2	13
				3	Reserved
BNKS	<1:0>	RW	0	Number of bank bits in the SDRAMs	
				Value	**Number of Bits**
				0	1
				1	2
				2	3 (Typhoon only)
				3	Reserved

D.15 DPR Registers for 680 Correctable Machine Check Logout Frames

DPR Locations A0:A9 represent the information that the console will read when a 680 machine check logout frame is loaded. They provide the interrupt information obtained by the RMC through the LM78 sensors. When an error occurs, the RMC writes the bits and delivers an IRQ to the SRM console. The SRM reads the bits and clears them. On the next 680 error, the RMC writes the error into the A0:A9 locations.

Table D–14 DPR Locations A0:A9

DPR Location	Description	
A0	If bit is set the associated fault is active.	
	Bit 0	+3.3v out of tolerance
	1	+5 v out of tolerance
	2	+12 v out of tolerance
	3	Vterm out of tolerance
	4	PCI backplane Zone 0 temp sensor is over temp
	5	BTI (overtemp signals from all CPU and LM78 sensors)
	6	Fan 1 fault (below the minimum RPM)
	7	Fan 2 fault (below the minimum RPM)
A1	Bit 0	CTERM out of tolerance
	2	–12 v out of tolerance

Continued on next page

Table D–14 DPR Locations A0:A9 (Continued)

DPR Location	Description
A2	If bit is set the associated fault is active. Bit 0 CPU0_VCORE out of tolerance 1 CPU0_VIO out of tolerance 2 CPU1_VCORE out of tolerance 3 CPU1_VIO out of tolerance 4 PCI backplane LM78 1 is over temp 5 Not Used 6 Fan 4 fault 7 Fan 5 fault
A3	Reserved
A4	If bit is set the associated fault is active. Bit 0 CPU2_VCORE out of tolerance 1 CPU2_VIO out of tolerance 2 CPU3_VCORE out of tolerance 3 CPU3_VIO out of tolerance 4 PCI backplane LM78 2 is over temp 5 Not used 6 Fan 3 fault 7 Fan 6 fault
A5	Bit 7 AC_input value high limit Bit 6 AC_input value low limit Bit 5 Minimum fan speed is not reached Bit 4 Current from +12 volt rail is out of tolerance Bit 3 Current from 5.5 volt rail is out of tolerance Bit 2 Current from 3.3 volt rail is out of tolerance Bit 1-0 Failing power supply number (0,1,2 are valid)

Table D–14 DPR Locations A0:A9 (Continued)

DPR Location	Description
A6	These bits indicate a door has been opened. Bit 0 unused 1 CPU door is open 2 Fan door is open 3 PCI door is open 5 System CPU door is open 6 System fan door is open 7 System PCI door is open
A7	Temperature Warning Mask Bit 0 CPU0 temp warning 1 CPU1 temp warning 2 CPU2 temp warning 3 CPU3 temp warning 4 Temp Zone 0 (LM78 0 on PCI backplane) 5 Temp Zone 1 (LM78 1 on PCI backplane) 6 Temp Zone 2 (LM78 2 on PCI backplane)
A8	Fan Controller Fault. This indicates a fan is not responding to a different RPM range as set by the RMC. (It is used to indicate that the fan failed to reach its maximum RPM at power-up). Bit 0 Fan 1 1 Fan 2 3 Fan 3 4 Fan 4 5 Fan 5 6 Fan 6
A9	These bits indicate which temperature zone the rise or fall in temperature occurred in. Bit 0 CPU fans spin at the maximum speed Bit 1 CPU fans reduce the speed from the maximum speed Bit 2 PCI fans spin at the maximum speed Bit 3 PCI fans reduce the speed from the maximum speed

D.16 DPR Power Supply Status Registers

The RMC reads nine bytes of information from each of the three power supplies. The first byte is read from an I/O expander port, the second four bytes and the last four bytes are read from the A–D converter.

Table D–15 Nine Bytes Read from Power Supply

DPR Location	Definition
DB/E4/ED	Reads I/O expander on Power Supply 0, 1, 2 Bit 0 PS_ID0_L 1 PS_ID1_L 2 Reserved (Pulled up so bit is always enabled) 3 Thermal_Shutdown_H 4:7 Tied to High within PS
DC/E5/EE	3.3V_current. Each step equals 0.255 (0xFF x 0.33203 = 85A)
DD/E6/EF	5 V_current. Each step equals 0.255 (0xFF x 0.33203 = 85A)
DE/E7/F0	12 V_current. Each step equals 0.033 (0xFF x 0.07813 = 20A)
DF/E8/F1	Fan_Speed (0x8B = 7 V)
E0/E9/F2	AC_INPUT value in hex. Each step equals 1.07422VAC (0xFF x 1.07422 = 275VAC)
E1/EA/F3	Power_supply_internal_temperature (hot) Byte represents a temp value 1 bit = 0.756• C
E2/EB/F4	Power_supply_inlet_temperature 1 bit = 0.266• C
E3/EC/F5	Spare

NOTE: *The DPR locations refer to power supplies. For example, DB/E4/ED = power supply 0/1/2. The same is true for all locations listed in the table.*

D.17 DPR 680 Fatal Registers

The RMC is powered by an auxiliary 5V supply that is independent from the system power subsystem. When any catastrophic failures (such as overtemperature failure) occur, this error state is captured as shown in Table D–16. The information is used to populate the console data log uncorrectable error frame in Environ_QW_8.

Table D–16 DPR 680 Fatal Registers

DPR Location	Definition
BD	Copy of the power supply AC input value Bit 0 PS0 1 indicates AC input is valid; 0 indicates invalid Bit 1 PS1 Bit 2 PS2
BE	Snapshot of the fault I/O expander, which indicates PS, VTERM, CPU regulator fault if bit is set. Bit 0 PS0 Bit 1 PS1 Bit 2 PS2 Bit 3 VTERM Bit 4 CPU0 Bit 5 CPU1 Bit 6 CPU2 Bit 7 CPU3
BF	RMC shutdown code Bit 0 Unused Bit 1 No CPU in CPU slot 0 Bit 2 Invalid CPU SROM voltage setting or checksum Bit 3 TIG load initialization or sequence fail Bit 4 Overtemperature failure Bit 5 CPU door open Bit 6 CPU fans 5 and 6 failed Bit 7 CTERM failure

D.18 CPU and System Uncorrectable Machine Check Logout Frame

The SRM console or the Windows NT HAL builds the uncorrectable machine check logout frames and passes them to the OS error handlers. The OS error handlers further process and subsequently log the formatted error event into the system binary error log.

Table D–17 CPU and System Uncorrectable Machine Check Logout Frame

63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0	Offset(Hex)
Retryable/Second Error Flags				Frame Size(00C8)				00000000
System Area Offet(00A0)				EV6 Area Offset(0018)				00000008
Machine Check Frame Revision(1)				Machine Check Code				00000010
EV6 Ibox Status (I_STAT<31:29>)								00000018
EV6 Dcache Status (DC_STAT<4:0>)								00000020
EV6 Cbox (C_ADDR<43:6>)								00000028
EV6 Cbox (C_SYNDROME_1<7:0>)								00000030
EV6 Cbox (C_SYNDROME_0<7:0>)								00000038
EV6 Cbox (C_STAT<4:0>)								00000040
EV6 Cbox (C_STS<3:0>)								00000048
EV6 TB Miss or Fault Status(MM_STAT<10:0>)								00000050
EV6 Exception Address (EXC_ADDR)								00000058
EV6 Interrupt Enablement and Current Processor Mode (IER_CM)								00000060
EV6 Interrupt Summary Register (ISUM)								00000068
EV6 Reserved 0								00000070
EV6 PAL Base Address (PAL_BASE)								00000078
EV6 Ibox Control (I_CTL)								00000080
EV6 Ibox Process Context (PCTX)								00000088
EV6 Reserved 1								00000090
EV6 Reserved 2								00000098
Software Error Summary Flags								000000A0
Cchip CPUx Device Interrupt Request Register (DIRx System Primary CPU Fault Watcher)								000000A8
Cchip Miscellaneous Register (MISC)								000000B0
Pchip 0 Error Register (P0_PERROR)								000000B8
Pchip 1 Error Register (P1_PERROR)								000000C0

NOTE: *For CPU uncorrectable offsets B0–B8 will be zeroed and system uncorrectable offsets 18–98 will be zeroed.*

D.19 Console Data Log Event Environmental Error Logout Frame (680 Uncorrectable)

Compaq Analyze uses the logout frame in Table D–18 for its decomposition of all 680 system environmental uncorrectable error frames.

Table D–18 Console Data Log Event Environmental Error Logout Frame (680 Uncorrectable)

63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0	Offset (Hex)
Revision (1)		Type (3)		Class (12)		Length (80)		00000000
Processor WHAMI								00000008
Retryable/Second Error Flags				Frame Size 0070)				00000010
System Area Offet(0020)				EV6 Area Offset(0020[1])				00000018
Machine Check Frame Revision				Machine Check Code (206)				00000020
Software Error Summary Flags								00000028
Cchip CPUx Device Interrupt Request Register (DIRx System Primary CPU Fault Watcher)								00000030
Environ_QW_1 (TIG System Management Information Register (SMIR))								00000038
Environ_QW_2 (TIG CPU Information Register (CPUIR))								00000040
Environ_QW_3 (TIG Power Supply Information Register (PSIR))								00000048
Environ_QW_4 (System_PS/Temp/Fan_Fault - LM78_ISR)								00000050
Environ_QW_5 (System_Doors)								00000058
Environ_QW_6(System_Temperature_Warning)								00000060
Environ_QW_7(System_Fan_Control_Fault)								00000068
Environ_QW_8(Fatal_Power_Down_Codes)								00000070
Environ_QW_9(Environmental Reserved 1)								00000078

NOTE: *Only Environ_QW_8 contains valid error state capture. All other Environ_QW_1-7, 9 will be zeroed.*

[1] Per Alpha SRM requirement.

D.20 CPU and System Correctable Machine Check Logout Frame

The SRM console or the Windows NT HAL builds the correctable machine check logout frames and passes them to the OS error handlers. The OS error handlers further process and subsequently log the formatted error event into the system binary error log. The operating systems contain built-in throttling mechanisms to handle high-volume bursting of these correctable error conditions.

Table D–19 CPU and System Correctable Machine Check Logout Frame

63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0	Offset (Hex)
Retryable / Second Error Flags				Frame Size(0080)				00000000
System Area Offet(0058)				EV6 Area Offset(0018)				00000008
Machine Check Frame Revision(1)				Machine Check Code				00000010
EV6 Ibox Status (I_STAT<31:29>)								00000018
EV6 Dcache Status (DC_STAT<4:0>)								00000020
EV6 Cbox (C_ADDR<43:6>)								00000028
EV6 Cbox (C_SYNDROME_1<7:0>)								00000030
EV6 Cbox (C_SYNDROME_0<7:0>)								00000038
EV6 Cbox (C_STAT<4:0>)								00000040
EV6 Cbox (C_STS<3:0>)								00000048
EV6 TB Miss or Fault Status(MM_STAT<10:0>)								00000050
Software Error Summary Flags (See section 1.4.2)								00000058
Cchip CPUx Device Interrupt Request Register (DIRx System Primary CPU Fault Watcher)								00000060
Cchip Miscellaneous Register (MISC)								00000068
Pchip 0 Error Register (P0-PERROR)								00000070
Pchip 1 Error Register (P1-PERROR)								00000078

NOTE: *For CPU correctable offsets 68–78 will be zeroed and system uncorrectable offsets 18–50 will be zeroed.*

D.21 Environmental Error Logout Frame (680 Correctable)

Table D–20 shows Environ_QW_1:7 and Environ_QW_8 error state capture information from DPR locations A0:A9 and BD:BF, respectively.

Table D–20 Environmental Error Logout Frame

63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0	Offset (Hex)
Retryable/Second Error Flags				Frame Size 0070)				00000000
System Area Offet(0018)				EV6 Area Offset(0018[1])				00000008
Machine Check Frame Revision(1)				Machine Check Code (206)				00000010
Software Error Summary Flags								00000018
Cchip CPUx Device Interrupt Request Register (DIRx System Primary CPU Fault Watcher)								00000020
Environ_QW_1 (TIG System Management Information Register (SMIR))								00000028
Environ_QW_2 (TIG CPU Information Register (CPUIR))								00000030
Environ_QW_3 (TIG Power Supply Information Register (PSIR))								00000038
Environ_QW_4 (System_PS/Temp/Fan_Fault - LM78_ISR)								00000040
Environ_QW_5 (System_Doors)								00000048
Environ_QW_6(System_Temperature_Warning)								00000050
Environ_QW_7(System_Fan_Control_Fault)								00000058
Environ_QW_8(Fatal_Power_Down_Codes)								00000060
Environ_QW_9(Environmental Reserved 1)								00000068

NOTE: *Only Environ_QW_1–7 contain valid error state capture. All other Environ_QW_8,9 will be zeroed.*

[1] Per Alpha SRM requirement.

D.22 Platform Logout Frame Register Translation

Compaq Analyze uses information from all logout frames for its decomposition of all error events. The error state bit definitions of all platform logout frame registers is shown in Table D–21.

Table D–21 Bit Definition of Logout Frame Registers

Register Identification	Bit Field	Text Translation Description			
C_SYNDROME_0	<7:0>	Syndrome for lower quadword in octaword of victim that was scrubbed as follows :			
		<7:0>(Hex)	Data Bit	<7:0>(Hex)	Data Bit
		CE	00	4F	32
		CB	01	4A	33
		D3	02	52	34
		D5	03	54	35
		D6	04	57	36
		D9	05	58	37
		DA	06	5B	38
		DC	07	5D	39
		23	08	A2	40
		25	09	A4	41
		26	10	A7	42
		29	11	A8	43
		2A	12	AB	44
		2C	13	AD	45
		31	14	B0	46
		34	15	B5	47
		0E	16	8F	48
		0B	17	8A	49
		13	18	92	50
		15	19	94	51
		16	20	97	52
		19	21	98	53
		1A	22	9B	54
		1C	23	9D	55
		E3	24	62	56
		E5	25	64	57
		E6	26	67	58
		E9	27	68	59
		EA	28	6B	60
		EC	29	6D	61

Continued on next page

Table D–21 Bit Definition of Logout Frame Registers (Continued)

Register Identification	Bit Field	Text Translation Description			
C_SYNDROME_0 (continued)		<7:0>(Hex)	Data Bit	<7:0>(Hex)	Data Bit
		F1	30	70	62
		F4	31	75	63
		01	CB0	10	CB4
		02	CB1	20	CB5
		04	CB2	40	CB6
		08	CB3	80	CB7
C_SYNDROME_1	<7:0>	Syndrome for upper quadword in octaword of victim that was scrubbed (same as specified above)			
C_STAT	<4:0>	<4:0>(Hex)	Detected Error [1]		
		00	No Error unless DC_STAT<3> = 1 indicating bcache/dcache victim read ECC error.		
		01	SNGL_BC_TAG_PERR		
		02	SNGL_DC_DUPLICATE_TAG_PERR		
		03	SNGL_DSTREAM_MEM_ECC_ERROR		
		04	SNGL_DSTREAM_BC_ECC_ERR		
		05	SNGL_DSTREAM _DC_ECC_ERR		
		06 or 07	SNGL_BC_PROBE _HIT_ERR		
		0B	SNGL_ISTREAM_MEM_ECC _ERR		
		0C	SNGL_ISTREAM_BC _ECC_ERR		
		13	DBL_DSTREAM_MEM_ECC_ERR		
		14	DBL_DSTREAM_BC_ECC_ERR		
		1B	DBL_ISTREAM_MEM_ECC_ERR		
		1C	DBL_ISTREAM_BC_ECC_ERR		
C_STS	<7:4>	Reserved			
	<3:0>	Captured status of the Bcache in INIT mode (<3>= Parity, <2> = Valid, <1> = Dirty, <0> = Shared).			
C_ADDR	<42:6>	Address of last reported ECC or parity error. If C_STAT<4:0> = 05(Hex) then only C_ADDR<19:6> are valid.			

[1] SNGL: Single-bit error leading to correctable error; DBL: double-bit error leading to uncorrectable error.

Table D–21 Bit Definition of Logout Frame Registers (Continued)

Register Identification	Bit Field	Text Translation Description
I_STAT	<63:41>	Reserved
	<40>	ProfileMe Mispredict Trap
	<39>	ProfileMe Trap
	<38>	ProfileMe Load-Store Order Trap
	<37:34>	ProfileMe Trap Types
	<33>	ProfileMe Icache Miss
	<32:30>	ProfileMe Counter 0 Overcount
	<29>	Set = icache encountered a parity error on instruction fetch and a reply trap is performed which generates a correctable read interrupt.
	<28:0>	Reserved
DC_STAT	<4:0>	00001(Bin) = Dcache tag probe pipeline 0 error; 00010(Bin) = Dcache tag probe pipeline 1 error; 00100(Bin) = Dcache data ECC error during store; 01000(Bin) = Dcache, Bcache or System fill data ECC error during load; 10000(Bin) = Dcache data store ECC error occurred within 6 cycles of the previous Dcache store ECC error.
MM_STAT	<3:0>	0001(Bin)= Write reference triggered error; 0010(Bin) = Reference caused an access violation; 0100(Bin) = PTE[FOR] bit set during read reference error; 1000(Bin) = PTE[FOW] bit set during write reference error.
	<10>	Set = Dcache tag parity correctable error during initial tag probe of load/store instruction.
	<9:4>	Opcode of instruction which triggered error.

Table D–21 Bit Definition of Logout Frame Registers (Continued)

Register Identification	Bit Field	Text Translation Description
EXC_ADDR	<0>	Set = exception or interrupt occurred in PAL mode
	<63:2>	Contains the PC address of the instruction that would have executed if the error interrupt did not occur.
IER_CM	<4:3>	00(Bin) = Kernel Mode, 01(Bin) = Executive Mode, 10(Bin) = Supervisor Mode, 11(Bin) = User Mode
	<13>	Set = enables those AST interrupt requests by ASTER
	<28:14>	Software interrupt enables
	<30:29>	Performance counter interrupt enables
	<31>	Set = Correctable read error interrupt enabled
	<32>	Set = Serial Line Interrupt Enabled
	<38:33>	External IRQ<5:0> enable
I_SUM	<4:3>	AST Kernel and Executive Interrupts pending ; <3> Set = Kernel Mode AST interrupt pending, <4> Set =Executive Mode AST interrupt pending
	<10:9>	AST Supervisor and User Interrupts pending ; <9> Set =Supervisor Mode AST interrupt pending, <10> Set =User Mode AST interrupt pending
	<28:14>	Software interrupts pending
	<32>	Serial line interrupt pending
	<31>	Set = Corrected read interrupt pending
	<30:29>	Performance counter interrupts pending
	<38:33>	External interrupts pending
PAL_BASE	<43:15>	Contains the physical base address for PALcode

Table D–21 Bit Definition of Logout Frame Registers (Continued)

Register Identification	Bit Field	Text Translation Description
I_CTL	<2:1>	01(Bin) and 10(Bin) for Icache set 1 or 2 enabled, respectively
	<7:6>	01(Bin) and 10(Bin) for R8-R11 & R24-R27 and R4-R7 & R20-R23 are used for PAL shadow registers, respectively
	<13>	Set = forces bad Icache tag parity
	<14>	Set = forces bad Icache data parity
	<15>	Clear and set for 43 bit or 48 bit virtual address format, respectively
	<20>	Clear or set for R23 or R27 used as CALL_PAL linkage register, respectively
	<21>	Set to enable machine check processing
	<29:24>	Revision ID number for EV6 Chip as follows: 01(Hex) = Pass 1.0; 02(Hex) = Pass 2.2; 03(Hex) = Pass 2.3; 0x04 (Hex) = Pass 3.0.
	<47:30>	Virtual page table base address
PCTX		Ibox process context register as follows :
	<0>	Reserved/RAZ
	<1>	If set, both performance counters are enabled
	<2>	If clear , floating-point instructions generate FEN exceptions
	<4:3>	Reserved/RAZ
	<8:5>	Enable AST U,S,E,K interrupt requests
	<12:9>	Request AST U,S,E.K interrupts
	<38:13>	Reserved/RAZ
	<46:39>	Address Space Number
	<63:47>	Reserved/RAZ
Software Error Summary Flags		PAL,HAL, and OS Error handler signaling software flags
	<0>	Set = Pchip0 P_Error<9:0> error has occurred.
	<1>	Set = Pchip1 P_Error<9:0> error has occurred.
	<2>	Set = Pchip0 or Pchip1 P_Error <11/10> uncorrectable/correctable error, or CPU correctable error, or CPU uncorrectable error has occurred.
	<63:3>	Unused

Table D–21 Bit Definition of Logout Frame Registers (Continued)

ID	Bit Field	Text Translation Description
MISC	<43:40>	Suppress IRQ1 interrupts to 1(Hex) for CPU0, 2(Hex) for CPU1, 4(Hex) for CPU2, and 8(Hex) for CPU3 Cchip
	<39:32>	Cchip Revision Level : 00-07(Hex) for C2, 08-0F(Hex) for C4
	<31:29>	0(Hex) for CPU0, 1(Hex) for CPU1, 2(Hex) for CPU2, 3(Hex) for CPU3, 4(Hex) for Pchip0, 5(Hex) for Pchip1, as device (source) which caused the NXM
	<28>	Set = NXM address detected, <31:29> are locked, DRIR <63> is set
	<24>	Write 1 = Arbitration Clear
	<23:20>	=1(Hex) for CPU0, 2(Hex) for CPU1, 4(Hex) for CPU2, and 8(Hex) for CPU3 Arbitration Trying
	<19:16>	=1(Hex) for CPU0, 2(Hex) for CPU1, 4(Hex) for CPU2, and 8(Hex) for CPU3 Arbitration Won
	<15:12>	=1(Hex) for CPU0, 2(Hex) for CPU1, 4(Hex) for CPU2, and 8(Hex) for CPU3 to set interprocessor interrupt request.
	<11:8>	=1(Hex) for CPU0, 2(Hex) for CPU1, 4(Hex) for CPU2, and 8(Hex) for CPU3 interprocessor interrupt (IRQ<3>) pending
	<7:4>	=1(Hex) for CPU0, 2(Hex) for CPU1, 4(Hex) for CPU2, and 8(Hex) for CPU3 interval timer interrupt (IRQ<2>) pending
	<1:0>	=00(Bin) for CPU0, 01(Bin) for CPU1, 10(Bin) for CPU2, 11(Bin) for CPU3 ID performing the read.

Table D–21 Bit Definition of Logout Frame Registers (Continued)

ID	Bit Field	Text Translation Description
DIRx	<63>	Internal Cchip asynchronous error [i.e.NXM] (IRQ0)
	<62>	P0_Pchip error (IRQ0)
	<61>	P1_Pchip error (IRQ0))
	<60>	P2_Pchip error (future designs) (IRQ0)
	<59>	P3_Pchip error (future designs) (IRQ0)
	<58>	OCP or RMC Halt(IRQ0)
	<57:56>	Unused
	<55>	INTR -PCI_ISA Device Interrupt error(IRQ1)
	<54>	SMI- System Mgmt Interrupt error(IRQ1)
	<53>	NMI - Non-Maskable Interrupt-fatal error (IRQ1)
	<52>	Unused
	<51>	Unused
	<50>	Environmental Temp,Doors,Fans errors (IRQ1)
	<49>	Unused
	<48>	Unused
	<47:44>	Pchip1_SLOT5[3:0]-System PCI Slot 9 INTa,b,c,d (IRQ1)
	<43:40>	Pchip1_SLOT4[3:0]-System PCI Slot 8 INTa,b,c,d (IRQ1)
	<39:36>	Pchip1_SLOT3[3:0]-System PCI Slot 7 INTa,b,c,d (IRQ1)
	<35:32>	Pchip1_SLOT2[3:0]-System PCI Slot 6 INTa,b,c,d (IRQ1)
	<31:28>	Pchip1_SLOT1[3:0]-System PCI Slot 5 INTa,b,c,d (IRQ1)
	<27:24>	Pchip1_SLOT0[3:0]-System PCI Slot 4 INTa,b,c,d (IRQ1)
	<23:20>	Pchip0_SLOT4[3:0]-System PCI Slot 3 INTa,b,c,d (IRQ1)
	<19:16>	Pchip0_SLOT3[3:0]-System PCI Slot 2 INTa,b,c,d (IRQ1)
	<15:12>	Pchip0_SLOT2[3:0]-System PCI Slot 1 INTa,b,c,d (IRQ1)
	<11:8>	Pchip0_SLOT1[3:0]-System PCI Slot 0 INTa,b,c,d (IRQ1) ***Note:Pchip0_SLOT0 = PCI/ISA Cypress/Acer Bridge***
	<7:0>	Unused

Table D–21 Bit Definition of Logout Frame Registers (Continued)

Register Identification	Bit Field	Text Translation Description
P0 & 1_ERROR	<63:56>	ECC Syndrome of CRE or UECC error - Same as EV6.
	<55:52>	When CRE or UECC failing transaction: 0000(Bin) = DMA Read; 0001(Bin) = DMA RMW; 0011(Bin) = S/G Read. PCI command of transaction when error not CRE or UECC : 0000(Bin) = PCI IACKCycle ; 0001(Bin) = PCI Special Cycle ; 0010(Bin) = PCI I/O Read; 0011(Bin) = PCI I/O Write; 0100(Bin) = Reserved ; 0101(Bin) = PCI PTP Write ; 0110(Bin) = PCI Memory Read ; 0111(Bin) = PCI Memory Write from CPUx; 1000(Bin) = PCI CSR Read;
	<51>	If clear = valid <63:56>,<55:52>, and <50:16> error information if any <11:0> bits are set, otherwise invalid.
	<50:16>	If <11> or <10> =set and <51> =clear, <50:19> = System address <34:3> of erred quadword and <18:16> = 000(Bin); else if any one of <9:0> =set and <51> = clear, <50:48> = 000(Bin),<47:18> = starting PCI address <31:2> of erred transaction, <17:16> = 00(Bin) if not DAC; 01(Bin) if DAC SG Windows 3; 1x(Bin) if Monster Window
	<15:12>	MBZ, RAZ
	<11>	Set = Correctable ECC Error (M or T[2])
	<10>	Set = Uncorrectable ECC Error (M or T)
	<9>	Reserved – MBZ/RAZ
	<8>	Set = No device select as PCI (M) error
	<7>	Set = PCI read data parity error as PCI (M)
	<6>	Set = Target abort error detected as PCI (M)
	<5>	Set = Address parity error detected as potential PCI
	<4>	Set = Invalid S/G page table entry detected as PCI
	<3>	Set = Delayed completion retry time-out error as PCI
	<2>	Set = PERR# error as PCI (M)
	<1>	Set = SERR# error as PCI (M or T)
	<0>	Set = Error occurred / lost after this register locked

[2] M refers to PCI Master; T refers to PCI Target

Table D–21 Bit Definition of Logout Frame Registers (Continued)

Register Identification	Bit Field	Text Translation Description
SMIR	<7>	Inverted Sys_Rst = System is being reset
(Environ_QW_1)	<6>	Inverted PCI_Rst1 = PCI Bus #1 is in reset
	<5>	Inverted PCI_Rst0 = PCI Bus #0 is in reset
	<4>	Set = System temperature over 50 degrees C failure
	<3>	unused
	<2>	Set = Sys_DC_Notok failure detected
	<1>	Inverted OCP_RMC_Halt = OCP or RMC halt detected
	<0>	Set = System Power Supply failure detected
CPUIR	<7>	Set = CPU3 regulator or configuration sequence fail
(Environ_QW_2)	<6>	Set = CPU2 regulator or configuration sequence fail
	<5>	Set = CPU1 regulator or configuration sequence fail
	<4>	Set = CPU0 regulator or configuration sequence fail
	<3>	Set = CPU3 regulator is enabled
	<2>	Set = CPU2 regulator is enabled
	<1>	Set = CPU1 regulator is enabled
	<0>	Set = CPU0 regulator is enabled
PSIR	<7>	Not Used
(Environ_QW_3)	<6>	Set = Power Supply 2 failed and was enabled
	<5>	Set = Power Supply 1 failed and was enabled
	<4>	Set = Power Supply 0 failed and was enabled
	<3>	Not Used
	<2>	Set = Power Supply 2 is enabled
	<1>	Set = Power Supply 1 is enabled
	<0>	Set = Power Supply 0 is enabled

Table D–21 Bit Definition of Logout Frame Registers (Continued)

Register Identification	Bit Field	Text Translation Description
System_PS/Temp/	<0>	Set = PS +3.3V out of tolerance
Fan_Fault_	<1>	Set = PS +5V out of tolerance
LM78_ISR	<2>	Set = PS +12V out of tolerance
(Environ_QW_4)	<3>	Set = VTERM out of tolerance
	<4>	Set = Temperature zone 0 (PCI Backplane slots 1-3 area) over limit failure
	<5>	Set = LM75 CPU0-3 Temperature over limit failure
	<6>	Set = System Fan 1 failure
	<7>	Set = System Fan 2 failure
	<8>	Set = CTERM out of tolerance
	<9>	Unused
	<10>	Set = -12V out of tolerance
	<15:11>	Unused
	<16>	Set = CPU0_VCORE +2V out of tolerance
	<17>	Set = CPU0_VIO +1.5V out of tolerance
	<18>	Set = CPU1_VCORE +2V out of tolerance
	<19>	Set = CPU1_VIO +1.5V out of tolerance
	<20>	Set = Temperature zone 1 (PCI Backplane slots 7-10 area) over limit failure
	<21>	Unused
	<22>	Set = System Fan 4 failure
	<23>	Set = System Fan 5 failure
	<31:24>	Unused
	<32>	Set = CPU2_VCORE +2V out of tolerance
	<33>	Set = CPU2_VIO +1.5V out of tolerance
	<34>	Set = CPU3_VCORE +2V out of tolerance
	<35>	Set = CPU3_VIO +1.5V out of tolerance
	<36>	Set = Temperature zone 2 (PCI Backplane slots 4-6 area) over limit failure
	<37>	Unused
	<38>	Set = System Fan 3 failure
	<39>	Set = System Fan 6 failure
	<41:40>	00(Bin) = Power supply 0; 01 (Bin) = power supply 1; 10 (Bin) = power supply 2; 11(Bin) = Reserved that has caused the <42:47> warning condition.
	<42>	Set = Power supply 3.3V rail above high amperage warning
	<43>	Set = Power supply 5.0V rail above high amperage warning
	<44>	Set = Power supply 12V rail above high amperage warning
	<45>	Set = Power supply high temperature warning
	<46>	Set = Power supply AC input low limit warning
	<47>	Set = Power supply AC input high limit warning
	<63:48>	Unused

Table D–21 Bit Definition of Logout Frame Registers (Continued)

Register Identification	Bit Field	Text Translation Description
System_Doors	<0>	Unused
(Environ_QW_5)	<1>	Set = System CPU door is open
	<2>	Set = System Fan door is open
	<3>	Set = System PCI door is open
	<4>	Unused
	<5>	Set = System CPU door is closed
	<6>	Set = System Fan door is closed
	<7>	Set = System PCI door is closed
	<63:8>	Unused
System_Tempera-	<0>	Set = CPU0 temperature warning fault has occurred
ture_Warning	<1>	Set = CPU1 temperature warning fault has occurred
(Environ_QW_6)	<2>	Set = CPU2 temperature warning fault has occurred
	<3>	Set = CPU3 temperature warning fault has occurred
	<4>	Set = System temperature zone 0 warning fault has occurred
	<5>	Set = System temperature zone 1 warning fault has occurred
	<6>	Set = System temperature zone 2 warning fault has occurred
	<63:7>	Unused
System_Fan_Con-	<0>	Set = System Fan 1 is not responding to RMC Commands
trol_Fault	<1>	Set = System Fan 2 is not responding to RMC Commands
(Environ_QW_7)	<2>	Set = System Fan 3 is not responding to RMC Commands
	<3>	Set = System Fan 4 is not responding to RMC Commands
	<4>	Set = System Fan 5 is not responding to RMC Commands
	<5>	Set = System Fan 6 is not responding to RMC Commands
	<7:6>	Unused
	<8>	Set = CPU fans 5/6 at maximum speed
	<9>	Set = CPU fans 5/6 reduced speed from maximum
	<10>	Set = PCI fans 1-4 at maximum speed
	<11>	Set = PCI fans 1-4 reduced speed from maximum.

Table D–21 Bit Definition of Logout Frame Registers (Continued)

Register Identification	Bit Field	Text Translation Description
Fatal_Power_Down_Codes (Environ_QW_8)	<0>	Set = Power Supply 0 AC input fail
	<1>	Set = Power Supply 1 AC input fail
	<2>	Set = Power Supply 2 AC input fail
	<3:7>	Unused
	<8>	Set = Power Supply 0 DC fail
	<9>	Set = Power Supply 1 DC fail
	<10>	Set = Power Supply 2 DC fail
	<11>	Set = Vterm fail
	<12>	Set = CPU0 Regulator fail
	<13>	Set = CPU1 Regulator fail
	<14>	Set = CPU2 Regulator fail
	<15>	Set = CPU3 Regulator fail
	<16>	Unused
	<17>	Set = No CPU in system motherboard CPU slot 0
	<18>	Set = Invalid CPU SROM voltage setting or checksum
	<19>	Set = TIG load initialization or sequence fail
	<20>	Set = Over temperature fail
	<21>	Set = CPU door open fail
	<22>	Set = System fan 5 (CPU backup fan) fail
	<23>	Set = Cterm fail
	<63:24>	Unused

Appendix E
Isolating Failing DIMMs

This appendix explains how to manually isolate a failing DIMM from the failing address and failing data bits. It also covers how to isolate single-bit errors. The following topics are covered:

- Information for Isolating Failures
- DIMM Isolation Procedure
- EV6 Single-Bit Errors

E.1 Information for Isolating Failures

Table E–1 lists the information needed to isolate the failure. See Appendix D for the register table for the Array Address Registers (AARs). The failing address and failing data can come from a variety of different locations such as the SROM serial line, SRM screen displays, the SRM event log, and errors detected by the 21264 (EV6) chip.

Convert the address to data bits if the address is not on a 256 bit alignment (address ends in a value less than 20 or address *xxxxx*20 or address *xxxxxnn*, where *nn* is 1 through 1F). For example, using failing address 0x1004 and failing data bit 8(dec), first multiply the failing address 4 by 8 = 32. Then add 32 to the failing data bit to yield the actual failing data bit 40. This conversion yields the new failing information to be failing address 0x1000 and failing data bit = 40(dec).

Table E–1 Information Needed to Isolate Failing DIMMs

Failing Address	
Failing Data/Check bits	
Array Address Registers	**Memory Addresses**
CSC	801.A000.0000
AAR0	801.A000.0100
AAR1	801.A000.0140
AAR2	801.A000.0180
AAR3	801.A000.01C0
DPR Locations	**Memory Addresses**
DPR:80	801.1000.2000
DPR:82	801.1000.2080
DPR:84	801.1000.2100
DPR:86	801.1000.2180

E.2 DIMM Isolation Procedure

Use the procedure in this section to isolate the failing DIMM.

1. Find the failing array by using the failing address and the Array Address Registers (AARs—see Appendix D). Use the AAR base address and size to create an Address range for comparing the failing address.

   For example if AAR1 base address was 40000000 (1 GB) and its size was 10000000 (256 MB), the address range would be 40000000–4FFFFFFF (4–4.25 GB). This range would be used to compare against the failing address.

2. Use one of the following methods to determine if the Address XORing is enabled:

   - If Bit 39 of the CSC register is set to 1, XORing is disabled.
   - Examine the contents of each AAR and compare bit 23 of each AAR, bit 22 of each AAR, through bit 0 of each AAR for the same values. If the values all match—bit 23 of AAR0 matches bit 23 of AAR1 matches bit 23 of AAR2 matches bit 23 of AAR3 (and the same for bits 22-0)—then bit 39 of the CSC register was cleared.

   If Address XORING is enabled, use Table E–2 to find the real array on which the failure occurred.

Table E–2 Determining the Real Failed Array

Failing Address <8:7>	Original Array 0	Original Array 1	Original Array 2	Original Array 3
00	Real Array 0	Real Array 1	Real Array 2	Real Array 3
01	Real Array 1	Real Array 0	Real Array 3	Real Array 2
10	Real Array 2	Real Array 3	Real Array 0	Real Array 1
11	Real Array 3	Real Array 2	Real Array 1	Real Array 0

3. After finding the real array, determine whether it is the lower array set or the upper array set. Use DPR locations 80, 82, 84, and 86 listed in Table E–1. Table E–3 shows the description of these locations.

Table E–3 Description of DPR Locations 80, 82, 84, and 86

DPR Location	Description	
80	Array 0 (AAR 0) Configuration	
	Bits<7:4>	Bits<3:0>
	4 = non split—lower set only	0 = Configured—Lowest array
	5 = split—lower set only	1 = Configured—Next lowest array
	9 = split—upper set only	2 = Configured—Second highest array
	D = split—8 DIMMs	3 = Configured—Highest array
	F = Twice split—8 DIMMs	4 = Misconfigured—Missing DIMM(s)
		8 = Miconfigured—Illegal DIMM(s)
		C = Misconfigured—Incompatible DIMM(s)
82	Array 1 (AAR 1) configuration	
84	Array 2 (AAR 2) configuration	
86	Array 3 (AAR 3) configuration	

4. Use the following table to determine the proper set. Bits<27,28,29,30,31,32> are from the failing address.

Array Size	Configuration Type Bits <7:4> from DPR		
	4 & 5	9	D & F
256MB	Lower Set	Upper Set	Bit <27> == 0 – Lower Set, 1– Upper Set
512MB	Lower Set	Upper Set	Bit <28> == 0 – Lower Set, 1– Upper Set
1GB	Lower Set	Upper Set	Bit <29> == 0 – Lower Set, 1– Upper Set
2GB	Lower Set	Upper Set	Bit <30> == 0 – Lower Set, 1– Upper Set
4GB	Lower Set	Upper Set	Bit <31> == 0 – Lower Set, 1– Upper Set
8GB	Lower Set	Upper Set	Bit <32> == 0 – Lower Set, 1– Upper Set

5. Now that you have the real array, the failing Data/Check bits, and the correct set, use Table E–4 to find the failing DIMM or DIMMs.

The table shows data bits 0–255 and check bits 0–31. These data bits indicate a single-bit error. An SROM compare error would yield address and data bits from 0–63. When you convert the address to be in the correct range, the failing data would be somewhere between 0 and 255.

Continued on next page

Table E–4 Failing DIMM Lookup Table

	Array 1		Array 2		Array 3		Array 4	
Data Bits	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set
0	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
1	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
2	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
3	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
4	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
5	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
6	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
7	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
8	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
9	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
10	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
11	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
12	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
13	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
14	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
15	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
16	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
17	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
18	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
19	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
20	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
21	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
22	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
23	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
24	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
25	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
26	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
27	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
28	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
29	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
30	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
31	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7

Table E–4 Failing DIMM Lookup Table (Continued)

	Array 1		Array 2		Array 3		Array 4	
Data Bits	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set
32	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
33	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
34	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
35	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
36	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
37	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
38	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
39	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
40	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
41	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
42	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
43	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
44	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
45	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
46	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
47	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
48	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
49	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
50	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
51	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
52	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
53	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
54	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
55	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
56	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
57	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
58	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
59	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
60	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
61	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
62	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7

Continued on next page

Table E–4 Failing DIMM Lookup Table (Continued)

	Array 1		Array 2		Array 3		Array 4	
Data Bits	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set
63	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
64	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
65	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
66	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
67	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
68	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
69	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
70	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
71	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
72	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
73	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
74	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
75	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
76	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
77	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
78	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
79	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
80	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
81	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
82	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
83	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
84	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
85	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
86	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
87	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
88	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
89	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
90	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
91	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
92	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
93	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7

Table E–4 Failing DIMM Lookup Table (Continued)

	Array 1		Array 2		Array 3		Array 4	
Data Bits	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set
94	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
95	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
96	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
97	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
98	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
99	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
100	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
101	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
102	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
103	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
104	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
105	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
106	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
107	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
108	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
109	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
110	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
111	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
112	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
113	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
114	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
115	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
116	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
117	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
118	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
119	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
120	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
121	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
122	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7

Continued on next page

Table E–4 Failing DIMM Lookup Table (Continued)

	Array 1		Array 2		Array 3		Array 4	
Data Bits	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set
123	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
124	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
125	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
126	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
127	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
128	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
129	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
130	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
131	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
132	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
133	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
134	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
135	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
136	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
137	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
138	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
139	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
140	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
141	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
142	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
143	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
144	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
145	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
146	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
147	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
148	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
149	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
150	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
151	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8

Table E–4 Failing DIMM Lookup Table (Continued)

	Array 1		Array 2		Array 3		Array 4	
Data Bits	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set
152	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
153	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
154	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
155	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
156	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
157	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
158	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
159	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
160	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
161	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
162	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
163	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
164	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
165	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
166	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
167	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
168	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
169	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
170	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
171	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
172	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
173	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
174	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
175	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
176	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
177	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
178	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
179	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
180	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
181	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8

Continued on next page

Table E–4 Failing DIMM Lookup Table (Continued)

	Array 1		Array 2		Array 3		Array 4	
Data Bits	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set
182	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
183	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
184	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
185	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
186	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
187	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
188	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
189	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
190	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
191	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
192	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
193	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
194	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
195	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
196	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
197	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
198	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
199	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
200	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
201	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
202	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
203	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
204	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
205	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
206	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
207	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
208	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
209	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
210	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8

Table E–4 Failing DIMM Lookup Table (Continued)

	Array 1		Array 2		Array 3		Array 4	
Data Bits	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set
211	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
212	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
213	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
214	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
215	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
216	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
217	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
218	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
219	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
220	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
221	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
222	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
223	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
224	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
225	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
226	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
227	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
228	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
229	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
230	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
231	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
232	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
233	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
234	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
235	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
236	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
237	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
238	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
239	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
240	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
241	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8

Continued on next page

Table E–4 Failing DIMM Lookup Table (Continued)

	Array 1		Array 2		Array 3		Array 4	
Data Bits	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set
242	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
243	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
244	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
245	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
246	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
247	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
248	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
249	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
250	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
251	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
252	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
253	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
254	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
255	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8

Table E–4 Failing DIMM Lookup Table (Continued)

	Array 1		Array 2		Array 3		Array 4	
Check Bits	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set	Upper Set	Lower Set
0	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
1	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
2	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
3	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
4	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
5	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
6	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
7	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
8	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
9	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
10	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
11	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
12	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
13	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
14	M:0 D:1	M:0 D:5	M:2 D:1	M:2 D:5	M:0 D:3	M:0 D:7	M:2 D:3	M:2 D:7
15	M:1 D:1	M:1 D:5	M:3 D:1	M:3 D:5	M:1 D:3	M:1 D:7	M:3 D:3	M:3 D:7
16	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
17	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
18	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
19	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
20	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
21	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
22	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
23	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
24	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
25	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
26	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
27	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
28	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
29	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8
30	M:0 D:2	M:0 D:6	M:2 D:2	M:2 D:6	M:0 D:4	M:0 D:8	M:2 D:4	M:2 D:8
31	M:1 D:2	M:1 D:6	M:3 D:2	M:3 D:6	M:1 D:4	M:1 D:8	M:3 D:4	M:3 D:8

E.3 EV6 Single-Bit Errors

The procedure for detection down to the set of DIMMs for a single-bit error is very similar to the procedure described in the previous sections. However, you cannot isolate down to a specific data or check bit.

The 21264 (EV6) chip detects and reports a C_ADDR<42:6> failing address that is accurate to the cache block (64 bytes). The syndrome registers (Table E–5) detect data syndrome information, providing isolation down to the low or high quadword of the target octaword that the fault has been detected within. Each of the syndrome registers is able to report 64 data bits (the quadword) and 8 check bits (memory data bus ECC bits).

Table E–5 shows the syndrome hexadecimal to physical data or check bit decoding. For example, if you have an EV6 single-bit C_Syndrome_0 hexadecimal error value equal to 23, the second column indicates the decoded physical data or check bit for this encoding. Use these physical data bits in conjunction with the previously described isolation procedure to isolate the failing DIMMs.

Table E–5 Syndrome to Data Check Bits Table

Syndrome	C_Syndrome 0	C_Syndrome 1
CE	Data Bit 0 or 128	Data Bit 64 or 192
CB	Data Bit 1 or 129	Data Bit 65 or 193
D3	Data Bit 2 or 130	Data Bit 66 or 194
D5	Data Bit 3 or 131	Data Bit 67 or 195
D6	Data Bit 4 or 132	Data Bit 68 or 196
D9	Data Bit 5 or 133	Data Bit 69 or 197
DA	Data Bit 6 or 134	Data Bit 70 or 198
DC	Data Bit 7 or 135	Data Bit 71 or 199
23	Data Bit 8 or 136	Data Bit 72 or 200
25	Data Bit 9 or 137	Data Bit 73 or 201
26	Data Bit 10 or 138	Data Bit 74 or 202
29	Data Bit 11 or 139	Data Bit 75 or 203
2A	Data Bit 12 or 140	Data Bit 76 or 204
2C	Data Bit 13 or 141	Data Bit 77 or 205

Table E–5 Syndrome to Data Check Bits Table (Continued)

Syndrome	C_Syndrome 0	C_Syndrome 1
31	Data Bit 14 or 142	Data Bit 78 or 206
34	Data Bit 15 or 143	Data Bit 79 or 207
0E	Data Bit 16 or 144	Data Bit 80 or 208
0B	Data Bit 17 or 145	Data Bit 81 or 209
13	Data Bit 18 or 146	Data Bit 82 or 210
15	Data Bit 19 or 147	Data Bit 83 or 211
16	Data Bit 20 or 148	Data Bit 84 or 212
19	Data Bit 21 or 149	Data Bit 85 or 213
1A	Data Bit 22 or 150	Data Bit 86 or 214
1C	Data Bit 23 or 151	Data Bit 87 or 215
E3	Data Bit 24 or 152	Data Bit 88 or 216
E5	Data Bit 25 or 153	Data Bit 89 or 217
E6	Data Bit 26 or 154	Data Bit 90 or 218
E9	Data Bit 27 or 155	Data Bit 91 or 219
EA	Data Bit 28 or 156	Data Bit 92 or 220
EC	Data Bit 29 or 157	Data Bit 93 or 221
F1	Data Bit 30 or 158	Data Bit 94 or 222
F4	Data Bit 31 or 159	Data Bit 95 or 223
4F	Data Bit 32 or 160	Data Bit 96 or 224
4A	Data Bit 33 or 161	Data Bit 97 or 225
52	Data Bit 34 or 162	Data Bit 98 or 226
54	Data Bit 35 or 163	Data Bit 99 or 227
57	Data Bit 36 or 164	Data Bit 100 or 228
58	Data Bit 37 or 165	Data Bit 101 or 229
5B	Data Bit 38 or 166	Data Bit 102 or 230
5D	Data Bit 39 or 167	Data Bit 103 or 231
A2	Data Bit 40 or 168	Data Bit 104 or 232
A4	Data Bit 41 or 169	Data Bit 105 or 233
A7	Data Bit 42 or 170	Data Bit 106 or 234
A8	Data Bit 43 or 171	Data Bit 107 or 235
AB	Data Bit 44 or 172	Data Bit 108 or 236
AD	Data Bit 45 or 173	Data Bit 109 or 237

Continued on next page

Table E–5 Syndrome to Data Check Bits Table (Continued)

Syndrome	C_Syndrome 0	C_Syndrome 1
B0	Data Bit 46 or 174	Data Bit 110 or 238
B5	Data Bit 47 or 175	Data Bit 111 or 239
8F	Data Bit 48 or 176	Data Bit 112 or 240
8A	Data Bit 49 or 177	Data Bit 113 or 241
92	Data Bit 50 or 178	Data Bit 114 or 242
94	Data Bit 51 or 179	Data Bit 115 or 243
97	Data Bit 52 or 180	Data Bit 116 or 244
98	Data Bit 53 or 181	Data Bit 117 or 245
9B	Data Bit 54 or 182	Data Bit 118 or 246
9D	Data Bit 55 or 183	Data Bit 119 or 247
62	Data Bit 56 or 184	Data Bit 120 or 248
64	Data Bit 57 or 185	Data Bit 121 or 249
67	Data Bit 58 or 186	Data Bit 122 or 250
68	Data Bit 59 or 187	Data Bit 123 or 251
6B	Data Bit 60 or 188	Data Bit 124 or 252
6D	Data Bit 61 or 189	Data Bit 125 or 253
70	Data Bit 62 or 190	Data Bit 126 or 254
75	Data Bit 63 or 191	Data Bit 127 or 255
01	Check Bit 0 or 16	Check Bit 8 or 24
02	Check Bit 1 or 17	Check Bit 9 or 25
04	Check Bit 2 or 18	Check Bit 10 or 26
08	Check Bit 3 or 19	Check Bit 11 or 27
10	Check Bit 4 or 20	Check Bit 12 or 28
20	Check Bit 5 or 21	Check Bit 13 or 29
40	Check Bit 6 or 22	Check Bit 14 or 30
80	Check Bit 7 or 23	Check Bit 15 or 31

Index

A

AAR memory addresses, E-2
Acceptance testing, 2-11
Advanced CMOS Setup screen, 6-23
Alpha System Reference Manual, 4-26
alphabios command, 6-4
AlphaBIOS console
  Auto Start option, 6-25
  boot screen, 3-21, 6-3
  hard disk setup, 6-22
  initialization screen, 3-20
  memory test, 6-23
  running in serial mode, 6-32
  setting date and time, 6-21
  setup screen, 6-2
  startup screens, 3-20
AlphaBIOS error frames, 5-20
AlphaBIOS firmware
  obtaining, 2-14
AlphaBIOS menus, 2-10
AlphaBIOS utilities, 6-28
Architecture, 1-2
Auto start, 6-24
  UNIX or OpenVMS, 6-26
  Windows NT, 6-25
auto_action environment variable, 6-12, 6-26
Autoboot, 6-24
AUX_5V LED, 1-25
AUX_5V power supply, 1-20
Auxiliary power supply, RMC, 7-3

B

Beep codes, 3-22
Boot device, setting, 6-27
Boot problems, 2-7
Boot screen, AlphaBIOS, 3-21, 6-3
Boot selections, Windows NT
  changing default, 6-25
boot_file environment variable, 6-12
boot_osflags environment variable, 6-12
bootdef_dev environment variable, 6-12
buildfru command, 4-4
Bypass modes, 7-6
Bypassing the RMC, 7-6

C

Cables, 8-2
cat el command, 4-8
CCAT, 2-11
C-chip, 1-3
CD-ROM drive, 1-6
  part number, 8-4
Chassis
  accessing in a cabinet, 8-14
  front components, 1-6
  rear components, 1-7
  removing covers from, 8-16
Checksum error, 3-24
Chipset, 1-3
clear password command, 6-36
clear_error all command, 4-10, 8-1, 8-9
clear_error command, 4-10, 4-48
Clearing checksum errors, 4-48
Clearing errors, 4-10
Clock generator settings, B-6
CMOS Setup screen, 6-21
COM1 data flow, defining, 7-15
COM1 environment variables, 7-12
COM1 MMJ port, 1-9
com1_modem environment variable, 6-15

com1_baud environment variable, 6-14
com1_flow environment variable, 6-14
com1_mode environment variable, 6-14, 7-4
COM2 and parallel port
  loopback tests, 4-54
COM2 port, 1-9
com2_baud environment variable, 6-14
com2_flow environment variable, 6-14
com2_modem environment variable, 6-15
Command conventions, RMC, 7-14
Compaq Analyze, 2-9
  and SDD errors, 4-48
  and TDD errors, 4-48
  documentation, 5-3
  event screen, 5-5
  evidence designator, 5-10
  FRU list designator, 5-8
  invoking GUI, 5-4
  overview, 5-2
  problem found report, 5-6
Compaq Crash Analysis Tool, 2-11
Components
  common, 1-5
  system front, 1-6
  system rear, 1-7
Configuration
  CPU, 6-41
  memory, 6-42
  OpenVMS, 6-10
  power supply, 6-48, 6-49
  Tru64 UNIX, 6-10
  viewing Tru64 UNIX or OpenVMS, 6-8
  Windows NT, 6-9
Configuring devices, 6-40
Connecting to RMC
  from local terminal, 7-10
  from local VGA, 7-11
Connectors, rear, 1-8
Connectors, system motherboard, 1-12
Console data log event environmental error
  logout frame (680 uncorrectable), D-43
console environment variable, 3-6, 6-5, 6-15,
  6-28
Console event log, 3-19
  displaying, 4-8
Console programs, 6-2
Console terminal, 1-32
Console, selecting, 6-5
Consoles, switching between, 6-4
Control panel, 1-10
Controls
  Halt button, 1-11
  Power button, 1-10
  Reset button, 1-11
Covers, 8-16
  removing from pedestal, 8-19
  removing from tower, 8-18
CPU
  configuration, 6-40
  part numbers, 8-3
  slot numbers, 6-40
CPU and system correctable machine check
  logout frame, D-44
CPU and system uncorrectable machine check
  logout frame, D-42
CPU cards, 1-12, 1-14
  removing, 8-26
CPU correctable error (630), 5-14
CPU uncorrectable error (670), 5-14
cpu_enabled environment variable, 6-15
crash command, 4-11
Crash dumps, 2-11, 4-11
Crashes, troubleshooting, 2-8

D

Data buses, 1-17
Data structures, displaying, 4-26
Date and time, in AlphaBIOS, 6-21
DC_STAT, D-6
Dcache Status Register, D-6
D-chips, 1-3
DEC VET, 2-11
DECevent, 5-2
deposit and examine commands, 4-12
Devices, configuring, 6-40
Devices, verifying, 4-56
Diagnostic commands
  buildfru, 4-4
  cat el, 4-8
  clear_error, 4-10
  clear_error all, 4-10
  crash, 4-11
  deposit and examine, 4-12
  exer, 4-16
  floppy_write, 4-21

grep, 4-22
hd, 4-24
info, 4-26
kill, 4-31
kill_diags, 4-31
memexer, 4-32
memtest, 4-34
more el, 4-8
net, 4-39
net -ic, 4-39
net -s, 4-39
nettest, 4-41
set sys_serial_num, 4-45
show error, 4-46
show fru, 4-49
show_status, 4-52
sys_exer, 4-54
test, 4-56
test -lb, 4-56
Diagnostic commands list, 4-2
Diagnostics
power-up, 3-1
running in background, 4-1
showing status of, 4-52
SRM console, 4-1
Dial-in configuration, 7-24
Dial-out alert, 7-26
DIMM arrays, 6-42
DIMM isolation
failing DIMM lookup table, E-6
information required for, E-2
procedure, E-3
DIMM lookup table, E-6
DIMMs
aligning in MMB, 8-30
configuring, 6-42
part numbers, 8-3
Director, Compaq Analyze, 5-3
Display device
selecting, 6-5
verifying, 6-6
Display Error Frames screen, 5-22
Display System Configuration screen, 6-9
Displaying error frames, 5-20
Displaying FRU configuration, 4-49
Double error halts, 5-21
DPR, 1-21
clearing errors, 8-1, 8-9
error respository, 7-3
DPR layout, C-2
DPR locations
80, 82, 84, and 86, E-4
DPR locations A0:
A9, D-37
DPR memory addresses, E-2
DPR registers, D-1
680 correctable machine check logout frames, D-37
680 fatal, D-41
power supply status, D-40
dump command (RMC), 7-20

E

ECC logic, 5-13
ei*0_inet_init environment variable, 6-15
ei*0_mode environment variable, 6-15
ei*0_protocols environment variable, 6-16
Enclosure panels
removing from a pedestal, 8-13
removing from a tower, 8-11
Enclosures, 1-4
env command (RMC), 7-18
Environment variables, 6-10, 6-12
setting, 6-11
Environment, monitoring, 7-18
Environmental error logout frame (680 correctable), D-45
Environmental errors captured by SRM, 5-18
Error beep codes, 3-22
Error frame
binary dump, 5-26
clearing log in AlphaBIOS, 5-23
deleting, 5-30
displaying in AlphaBIOS, 5-23
formatted text file, 5-28
saving to floppy, 5-27
text style, 5-24
viewing text-style, 5-24
Error handling tools, 2-9
Error log event structure map, 5-17
Error log format, 5-16

Error logs, 5-1
browsing in AlphaBIOS, 5-25
Windows NT, 5-20
Error messages
power-up, 3-22
RMC, 3-28
SROM, 3-30, 3-31
Error repository, clearing, 8-1, 8-9
Escape sequence (RMC), 7-10
Ethernet external loopback, 4-54
EV6 (21264) microprocessor, 1-15
EV6 registers, D-1
Event log, 3-19
Event structure map, 5-16
ew*0_inet_init environment variable, 6-15
ew*0_mode environment variable, 6-15
ew*0_protocols environment variable, 6-16
exer command, 4-16
Exercising devices, 4-16
Exercising memory, 4-32, 4-34

F

Fail-safe loader, 2-14, 3-24
activating, 3-32, 3-33
jumpers, 3-32
Fans, 1-26
part numbers, 8-2
replacing, 8-22
Fatal error halts, 5-21
Fault detection and correction, 5-12
Firm Bypass mode, 7-8
Firmware updates, 2-13, 3-25
Flash SROM, 3-9
Floppy diskette drive, 1-6
Floppy drive
part number, 8-4
removing, 8-38
floppy_write script, 4-21
Front doors, 1-32
FRU assembly hierarchy, 4-5
FRU descriptor, 4-6
FRU EEPROMs
viewing errors logged to, 4-46
FRUs
displaying physical configuration, 4-49
hot-plug, 8-8
locations, 8-6
part numbers, 8-2
tools for removing, 8-8
Function jumpers, 3-32

G

Graphics mode, 6-28
grep command, 4-22
Greycode test, 4-35, 4-36

H

Halt button, 1-11
with login command, 6-37
halt in/out commands (RMC), 1-11, 7-23
Halt LED, 1-11
Halt, remote, 1-11, 7-23
hangup command (RMC), 7-25
Hard Disk Setup screen, 6-22
Hard disk setup, AlphaBIOS, 6-22
Hard drive, removing, 8-24
Hard drives, 1-29
Hardware configuration
AlphaBIOS, 2-10
viewing, 6-8
Windows NT, 6-9
hd command, 4-24
Heap space, resizing, 3-14
heap_expand environment variable, 6-16
Hex dump, 4-24
Hot-plug FRUs, 8-8

I

I/O connector assembly, removing, 8-40
I/O connectors, 1-8
I/O control logic, 1-18
I/O implementation, 1-19
info 0 command, 4-26
info 1 command, 4-27
info 2 command, 4-28
info 3 command, 4-29
info 4 command, 4-30
info command, 4-26
Information resources, 2-13
Initialization screen, AlphaBIOS, 3-20
Interlock switch, 8-17

Internal processor registers (21264), D-1
Interrupts, 5-14
Invoking SRM from AlphaBIOS, 6-4

J

Jumpers
PCI, B-8
RMC and SPC, B-2
setting, B-10
TIG/SROM, B-4
Jumpers and switches, B-1
Junk I/O. *See* I/O connector assembly

K

kbd_hardware_type environment variable, 6-16
Key mapping, AlphaBIOS, 6-32
Keyboard port, 1-9
Keys, 1-30
kill command, 4-31
kill_diags command, 4-31
KZPAC-*xx* RAID controllers, 6-34
kzpsa_host_id environment variable, 6-16

L

language environment variable, 6-17
LEDs
control panel, 1-10
power supply, 1-25
LFU utility, 3-25, 3-35
Line voltage, 1-25
Local mode, 7-5
login command, 6-36, 6-37
Logout frame
for console level environmental error, 5-18
Logout frame registers, bit definitions, D-47
Loopback connectors, 4-55
Loopback tests, 2-9, 4-56
COM2 and parallel ports, 4-54

M

Machine checks, 5-14
memexer command, 4-32
Memory allocation, SRM, 3-14
Memory architecture, 1-16
Memory buses, 1-3
Memory configuration, 6-42
pedestal, 6-44
tower, 6-45
Memory exercisors, 4-32, 4-34
Memory failure, 3-9
Memory interleaving, 1-17
Memory motherboards. *See* MMBs
Memory options, 1-17
Memory test, AlphaBIOS, 6-23
memory_text environment variable, 6-17
memtest command, 4-34
memtest test 1, 4-36
Microprocessor, 21264, 1-15
MM_STAT Register, D-4
MMBs, 1-17
location of, 1-12
part number, 8-3
removing, 8-28
Model 1 and Model 2 systems, 1-5
Modem port, 1-9
Modules, processor, 1-12
MOP loopback tests, 4-42
more el command, 3-19, 4-8
Motherboard logic, 1-13
Motherboard, removing, 8-47
Mouse port, 1-9

N

net command, 4-39
net -ic command, 4-39
net -s command, 4-39
nettest command, 4-41
Network ports, testing, 4-41
No MEM error, 3-26

O

OCP, 1-10
customized message, 6-7
error messages, 3-22
OCP assembly, removing, 8-34
ocp_text environment variable, 6-17
Operating system exercisers, 2-11

Operating systems
errors reported by, 2-8
switching between, 6-50
switching to UNIX or OpenVMS, 6-52
switching to Windows NT, 6-50
Operator control panel. *See* OCP
Options, supported, 2-15
os_type environment variable, 6-5, 6-17
OSLOADER program, 6-27

P

Pagers, 7-27
PAL handler, 5-12
PALcode
error routines, 5-14
exception/interrupt handling, 5-12
Parallel port, 1-9
password environment variable, 6-17
Password protection, Windows NT, 6-38
Patches, 2-14
P-chips, 1-3
PCI backplane, 1-18
cables, 8-42
part numbers, 8-3
removing, 8-44
PCI bus implementation, 1-19
PCI buses, 6-47
PCI card
installing or replacing, 8-32
PCI card cage, 8-17
PCI jumpers, B-8
PCI slot locations
for VGA controller, 1-9
pedestal, 6-46
tower, 6-47
pci_parity environment variable, 6-17
PIC processor, 1-22, 1-23, 7-3
pk*0_fast environment variable, 6-18
pk*0_host_id environment variable, 6-18
pk*0_soft_term environment variable, 6-18
Platform logout frame register translation, D-46
POK LED, 1-25
Ports, system rear, 1-9
Power button, 1-10
Power cords, 8-5
Power harness, removing, 8-51
Power LED, 1-11
power on/off commands (RMC), 1-11, 7-22
Power problems, 2-4
Power supplies, 1-24
configuring, 6-48, 6-49
installation order, 6-49
installing, 8-21
LEDs, 1-25
locations, 6-48
numbering, 6-48
redundant, 6-49
removing, 8-20
Power supply
part number, 8-4
Power-on, remote, 1-11
Power-on/off, from RMC, 7-22
Power-up diagnostics, 3-1
RMC, 3-2
SRM, 3-2
SROM, 3-2
Power-up display
SRM, 3-10
SROM, 3-8, 3-9
Power-up displays, 3-6
Power-up error messages, 3-22
Power-up memory failure, 3-9
Power-up memory test, AlphaBIOS, 6-23
Power-up procedure, 3-9
Power-up sequence, 3-3, 3-4
Processor card, 1-12, 1-14

Q

quit command (RMC), 7-10

R

RAID utility, running, 6-34
RCM tool, 2-11
Real failed array, finding, E-3
Redundant power supply, 6-49
Register translation, platform logout frames, D-46
Registers, D-1
Registers (21272)
AAR0-AAR3, D-35
MISC, D-26
PERROR, D-31

Registers (21272) DIRn, D-29
Registers (EV6)
  Cbox Read, D-8
  DC_STAT, D-6
  EXC_ADDR, D-10
  I_CTL, D-18
  I_STAT, D-2
  IER_CM, D-12
  ISUM, D-14
  MM_STAT, D-4
  PAL_BASE, D-16
  PCTX, D-23
Registers, displaying, 4-26
Remote management console. *See* RMC
Remote power-on/off, 7-22
Remote system management logic, 1-20
Removable media, 1-28
  removing 5.25-inch device, 8-36
Removable media bays, 1-6
Removal and replacement, 8-1
Removing covers from chassis, 8-16
Removing enclosure panels, 8-10
  from a pedestal, 8-13
  from a tower, 8-11
Reset button, 1-11
reset command (RMC), 1-11, 7-23
Revision and Configuration Management tool, 2-11
RMC, 1-20, 2-10, 7-1
  auxiliary power supply, 7-3
  bypass modes, 7-6
  CLI, 6-3, 7-10, 7-13
  command conventions, 7-14
  commands, 7-13
  configuring call-out, 7-24
  connecting from local VGA, 7-11
  connecting from serial terminal, 7-10
  data flow diagram, 7-4
  dial-out alert, 7-26
  dump command, 7-20
  env command, 7-18
  error information, 7-3
  escape sequence, 7-10
  exiting, 7-10
  exiting from local VGA, 7-11
  fatal error messages, 3-28
  Firm bypass mode, 7-8
  hangup command, 7-25
  jumpers, 7-30
  Local mode, 7-5
  logic, 1-23, 7-3
  operating modes, 7-4
  overview, 1-23, 7-2
  PIC processor, 7-3
  quit command, 7-10
  remote power on/off, 7-22
  remote reset, 7-23
  resetting to factory defaults, 7-30
  set com1_mode command, 7-15
  set escape command, 7-29
  Snoop mode, 7-7
  Soft Bypass mode, 7-7
  status command, 7-16
  terminal setup, 7-9
  Through mode, 7-5
  troubleshooting, 7-32
  updating, 3-34
  warning messages, 3-29
RMC and SPC jumpers, B-2
Run Maintenance Program dialog box, 6-30

S

SCB offsets, 5-14
SCSI breakouts, 1-9
SDD errors, 4-47
Security
  SRM, 6-35
  Windows NT, 6-38
Serial mode, 6-28
  setting up, 6-31
Serial number mismatch, 4-47
Serial terminal, 1-32, 6-5
  running utilities from, 6-32
  set-up, 6-31
Service tools CD, 2-13
set com1_mode command (RMC), 7-15
set console command, 6-6
set *envar* command, 6-11
set escape command (RMC), 7-29
set heap_expand command, 3-14
set ocp_text command, 6-7
set password command, 6-35
set secure command, 6-36
set sys_serial_num command, 4-45
Setup, Windows NT, 6-20

show boot* command, 6-8
show config command, 6-8
show console command, 6-6
show device command, 6-8
show *envar* command, 6-11
show error command, 4-46
  message translation, 4-48
show fru command, 4-49, 6-8
show fru E field, 4-51
show memory command, 6-8
show power command, 6-49
show_status command, 4-52
Single-bit errors (EV6), detecting, E-16
Slot locations, PCI, 6-46
Slot numbers
  CPUs, 6-40
  PCI, 6-47
Snoop mode, 7-7
Soft Bypass mode, 7-7
Software patches, 2-14
SPC logic, 1-22
Speaker, testing, 4-56
SRM console, 6-3
  diagnostic commands, 4-2
  diagnostics, 4-1
  environment variables, 6-12
  memory allocation, 3-14
  power-up display, 3-10
  problems accessing, 2-5
  problems reported by, 2-6
SRM console commands, 2-9
SRM console commands, listed, A-1
SRM console password, 6-35
SROM
  error messages, 3-30, 3-31
  power-up display, 3-8, 3-9
Stabilizer bar, 8-15
Startup screens, AlphaBIOS, 3-20
status command (RMC), 7-16
Storage
  hard drive, 1-29
  removable media, 1-28
Storage drive bays, 1-6
StorageWorks Command Console, 2-12
Supported Options List, 2-15
SWCC tool, 2-12
Switched system interconnect, 1-3
sys_exer command, 4-54
sys_serial_num environment variable, 6-19
System access, 1-30
System architecture, 1-2
System block diagram, 1-2
System card cage, 8-17
System correctable error (620), 5-15
System enclosures, 1-4
System environmental error (680), 5-15
System Error Logging Software for Alpha kit, 5-21
System motherboard, 1-12
  removing, 8-47
System power controller (SPC), 1-22
System registers (21272), D-1
System serial number
  setting, 4-45, 8-49
System uncorrectable error (660), 5-15

T

TDD errors, 4-47
Technical information, on Web, 2-15
Terminal setup (RMC), 7-9
Terminating diagnostics, 4-31
test command, 4-56
test -lb command, 4-56
Test script, 4-57
Testing floppy and tape drives, 4-55, 4-57
Testing network ports, 4-41
Through mode (RMC), 7-5
TIG chip, 1-21, 7-3
TIG/SROM jumpers, B-4
Tools and utilities, 2-9
Troubleshooting
  boot problems, 2-7
  crash dumps, 2-11
  errors reported by operating system, 2-8
  power problems, 2-4
  problem categories, 2-3
  problems getting to console, 2-5
  problems reported by console, 2-6
  RMC, 7-32
  strategy, 2-2
  tools and utilities for, 2-9
  with console event log, 3-19
Tsunami/Typhoon system registers, D-1
tt_allow_login environment variable, 6-19

U

UART ports, 7-5
Updating RMC, 3-34
USB ports, 1-9
User interfaces, 6-2
Utilities
  AlphaBIOS, 6-28
  running from serial terminal, 6-32
  running from VGA, 6-29
Utilities menu, 6-29

V

Verifying devices, 4-56
VGA console tests, 4-57
VGA controller, slot for, 6-47
VGA monitor, 1-32, 6-5
VT terminal, 6-5

W

Warning messages, RMC, 3-29
WEBES Director, 5-3
Windows NT Crash Dump Collector, 2-11
Windows NT, testing, 4-57
Write test, on floppy, 4-21