



AlphaServer GS60E

Operations Manual

Order Number: EK-GS60E-OP. A01

This manual is intended for the system manager or system operator and covers the basic operation of the *Compaq AlphaServer* GS60E system.

Compaq Computer Corporation

First Printing, August 1999

The information in this publication is subject to change without notice.

COMPAQ COMPUTER CORPORATION SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN, NOR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL.

This publication contains information protected by copyright. No part of this publication may be photocopied or reproduced in any form without prior written consent from Compaq Computer Corporation.

The software described in this document is furnished under a license agreement or nondisclosure agreement and may be used or copied only in accordance with the terms of the agreement.

© 1999 Compaq Computer Corporation.

All rights reserved. Printed in the U.S.A.

COMPAQ and the Compaq logo are trademarks or registered trademarks of Compaq Computer Corporation. AlphaServer, DIGITAL, OpenVMS, and StorageWorks are trademarks or registered trademarks of Digital Equipment Corporation. Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation. UNIX is a registered trademark in the U.S. and other countries, licensed exclusively through X/Open Company Ltd. Other product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

FCC Notice: The equipment described in this manual 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 are required to correct the interference.

Shielded Cables: If shielded cables have been supplied or specified, they must be used on the system in order to maintain international regulatory compliance.

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.

Avertissement! Cet appareil est un appareil de Classe A. Dans un environnement résidentiel, cet appareil peut provoquer des brouillages radioélectriques. Dans ce cas, il peut être demandé à l'utilisateur de prendre les mesures appropriées.

Contents

Preface	vii
---------------	-----

Chapter 1 Introduction

1.1 System Overview.....	1-2
1.2 Console Firmware and Utilities Overview	1-4
1.3 System Architecture	1-6

Chapter 2 AlphaServer GS60E System

2.1 System Characteristics	2-2
2.2 System Front View.....	2-4
2.3 System Rear View	2-6
2.4 System Components	2-8
2.4.1 TLSB Card Cage	2-8
2.4.2 Cabinet Control Logic Panel.....	2-10
2.4.3 Console Load Device.....	2-12
2.4.4 Power System	2-14
2.5 Controls and Indicators	2-16
2.5.1 Operator Control Panel Keypress and Pushbuttons	2-16
2.5.2 Operator Control Panel Indicator Lights	2-18
2.5.3 Circuit Breaker and AC Power Indicators.....	2-20
2.5.4 Cooling System	2-22

Chapter 3 I/O Subsystem

3.1 I/O Subsystem and Module Overview	3-2
3.2 System Configuration Information	3-4
3.3 DWLPB PCI and KFE72 Adapters.....	3-10

Chapter 4 Booting an Operating System

4.1 Preparation.....	4-2
4.1.1 Set os_type Environment Variable.....	4-2
4.1.2 Set console Environment Variable.....	4-3
4.1.3 Set auto_action Environment Variable.....	4-4
4.2 Selecting a Boot Device.....	4-6

4.2.1	Show Config and Show Device Commands	4-8
4.2.2	Boot Environment Variables.....	4-11
4.3	Booting OpenVMS.....	4-12
4.4	Booting Tru64 UNIX	4-14

Chapter 5 System Troubleshooting

5.1	Troubleshooting During Power-Up.....	5-2
5.2	Troubleshooting During Booting	5-4
5.3	Troubleshooting a PCI Shelf.....	5-6
5.4	Troubleshooting a StorageWorks Shelf.....	5-8
5.5	Self-Test Overview	5-10
5.6	Sample Self-Test Display	5-12
5.7	Self-Test Lines NODE# and TYPE	5-14
5.8	Self-Test Lines ST and BPD.....	5-16
5.9	Self-Test Lines C0, C1, C2, ... Cn	5-18
5.10	Self-Test Lines ILV and GB	5-20
5.11	Self-Test Identification Line	5-22
5.12	Show Commands	5-24
5.12.1	Show Configuration	5-24
5.12.2	Show Network.....	5-26
5.12.3	Show Device	5-28
5.13	Test Command.....	5-30
5.13.1	Testing the System	5-32
5.13.2	Testing a Module or Devices	5-34
5.14	Error Reports	5-40

Chapter 6 SRM Console Commands

6.1	Overview.....	6-2
6.2	SRM Command Syntax	6-4
6.3	SRM Console Special Characters	6-6
6.4	SRM Environment Variables.....	6-9
6.5	SRM Console Commands.....	6-14
6.5.1	Boot	6-14
6.5.2	Building the EEPROM.....	6-15
6.5.3	Building the Nonvolatile RAM.....	6-16
6.5.4	Building the SEEPROM.....	6-17
6.5.5	Clear EEPROM.....	6-18
6.5.6	Clear <envar>	6-19
6.5.7	Clear Screen.....	6-20
6.5.8	Continue	6-21
6.5.9	Crash	6-23
6.5.10	Create.....	6-24

6.5.11	Date	6-25
6.5.12	Deposit.....	6-26
6.5.13	Examine.....	6-29
6.5.14	Halt.....	6-32
6.5.15	Help or Man.....	6-33
6.5.16	Init.....	6-35
6.5.17	Prcache	6-36
6.5.18	Run	6-37
6.5.19	Set EEPROM	6-39
6.5.20	Set <envar>.....	6-40
6.5.21	Set SEEPROM	6-41
6.5.22	Set Time.....	6-42
6.5.23	Show Configuration	6-43
6.5.24	Show CPU.....	6-44
6.5.25	Show Device	6-45
6.5.26	Show EEPROM.....	6-46
6.5.27	Show <envar>.....	6-47
6.5.28	Show Memory.....	6-48
6.5.29	Show Network.....	6-49
6.5.30	Show SEEPROM.....	6-50
6.5.31	Show Time.....	6-51
6.5.32	Start	6-52
6.5.33	Stop.....	6-53
6.5.34	Test.....	6-54
6.5.35	Type.....	6-55
6.5.36	Vga.....	6-57
6.5.37	Comment (#).....	6-58

Appendix A Open VMS and Tru64 UNIX Boot Options

Appendix B Diagnostics and Utilities

B.1	Booting LFU with OpenVMS and Tru64 UNIX Systems.....	B-2
B.2	List	B-4
B.3	Update	B-6
B.4	Exit.....	B-10
B.5	Display and Verify Commands.....	B-12
B.6	Create	B-14

Appendix C Running Configuration Utilities from the SRM Console

C.1	Configuring a RAID Storage Array	C-2
C.2	ISP1020/1040 Configuration Utility	C-11

Examples

3-1	System Self-Test Display	3-4
3-2	Sample Show Configuration Command	3-6
3-3	Sample Show Device Command	3-8
4-1	Setting os_type for OpenVMS	4-1
4-2	Setting os_type for Tru64 UNIX	4-1
4-3	Set Console to Serial for Tru64 UNIX and Open VMS Systems	4-3
4-4	Setting the auto_action Environment Variable	4-4
4-5	Show Config and Show Device Commands	4-8
4-6	Viewing and Setting Boot Environment Variables	4-11
4-7	OpenVMS Boot	4-12
4-8	Tru64 UNIX Boot	4-14
5-1	Self-Test Results	5-12
5-2	Self-Test Results: NODE # and TYP	5-14
5-3	Self-Test Results: ST and BPD	5-16
5-4	Self-Test Results: C0, C1, C2, ... Cn	5-18
5-5	Self-Test Results: ILV and GB	5-20
5-6	Self-Test Results: Identification Line	5-22
5-7	Sample System Hardware Configuration	5-24
5-8	Sample Output of Show Network Command	5-26
5-9	Sample Output of Show Device Command	5-28
5-10	Sample Test Command	5-30
5-11	Sample Test Command, System Test	5-32
5-12	Sample Test Command, Memory Test	5-34
5-13	Sample Summary Error Report	5-36
6-1	Boot Command	6-14
6-2	Building the EEPROM	6-15
6-3	Building the Nonvolatile RAM	6-16
6-4	Building the SEEPROM	6-17
6-5	Clear EEPROM	6-18
6-6	Clear <envar>	6-19
6-7	Clear Screen Command	6-20
6-8	Continue Command	6-21
6-9	Crash Command	6-23
6-10	Create Command	6-24
6-11	Date Command	6-25

6-12	Deposit Command.....	6-26
6-13	Examine Command.....	6-29
6-14	Halt Command.....	6-32
6-15	Help Command.....	6-33
6-16	Init Command.....	6-35
6-17	Prcache Command.....	6-36
6-18	Run Command.....	6-37
6-19	Prcache Command.....	6-36
6-20	Set EEPROM Command.....	6-39
6-21	Set <envar> Command.....	6-40
6-22	Set SEEPROM Command.....	6-41
6-23	Set Time Command.....	6-42
6-24	Show Configuration Command.....	6-43
6-25	Show CPU Command.....	6-44
6-26	Show Device Command.....	6-45
6-27	Show EEPROM Command.....	6-46
6-28	Show <envar> Command.....	6-47
6-29	Show Memory Command.....	6-48
6-30	Show Network Command.....	6-49
6-31	Show SEEPROM Command.....	6-50
6-32	Show Time Command.....	6-51
6-33	Start Command.....	6-52
6-34	Stop Command.....	6-53
6-35	Test Command.....	6-54
6-36	Type Command.....	6-56
6-37	Vga Command.....	6-57
6-38	Comment (#) Command.....	6-58
B-1	Booting LFU from CD-ROM.....	B-2
B-2	List Command.....	B-4
B-3	Update Command.....	B-6
B-4	Exit Command.....	B-10
B-5	Display and Verify Commands.....	B-12
B-6	Create Command.....	B-14
C-1	ISP1020/1040 Configuration Utility Opening Screen.....	C-10
C-2	ISP1020/1040 Configuration Utility Second Screen.....	C-10
C-3	Edit Host Adapter Parameters Example.....	C-11

Figures

1-1	AlphaServer GS60E System	1-2
1-2	Accessing Firmware at the Console Device	1-4
1-3	Sample System Architecture.....	1-6
2-1	Sample System Footprint.....	2-2
2-2	System Front View	2-4
2-3	System Rear View	2-6
2-4	TLSB Card Cage.....	2-8
2-5	Cabinet Control Logic Panel	2-10
2-5	Accessing the Console Load Device	2-12
2-7	Power System.....	2-14
2-8	Operator Control Panel Keyswitch and Pushbuttons	2-16
2-9	Operator Control Panel LEDs.....	2-18
2-10	Circuit Breaker and AC Power Indicators.....	2-20
2-11	Cabinet Airflow.....	2-22
3-1	I/O Subsystem	3-2
3-2	KFTHA I/O Port Module.....	3-3
3-3	Hose Numbering Scheme for KFTHA	3-5
3-4	PCI Slot Configuration Used with the KFE72.....	3-10
3-5	KFE72 and Graphics Modules.....	3-11
5-1	Power-Up Troubleshooting Flowchart	5-2
5-2	Power-Up Troubleshooting Steps.....	5-3
5-3	Booting Troubleshooting Flowchart.....	5-4
5-4	Troubleshooting Steps During Booting.....	5-5
5-5	DWLPCB PCI Shelf	5-6
5-6	Troubleshooting Steps for PCI Shelf.....	5-7
5-7	StorageWorks Indicator LEDs.....	5-8
5-8	Determining Self-Test Results.....	5-10
6-1	SRM Console Command and Feature Breakdown.....	6-2

Tables

1	AlphaServer GS60E Documentation	xii
2-1	Electrical Characteristics.....	2-3
2-2	Environmental Characteristics	2-5
2-3	Control/Status and I/O Connections	2-11
2-4	Keyswitch and Pushbutton Positions.....	2-17
2-5	Operator Control Panel LEDs.....	2-19
4-1	Boot Devices	4-6
4-2	SRM Console Device Naming Conventions.....	4-10
5-1	SCSI Disk Drive LEDs.....	5-9

5-2	System Configuration for Example 5-1	5-12
5-3	Test Command Options.....	5-31
5-4	Test Command Environment Variables	5-31
6-1	SRM Console Command Language Syntax	6-4
6-2	SRM Console Special Characters	6-6
6-3	Environment Variables.....	6-10
6-4	Deposit Command Options.....	6-27
6-5	Device Name and Address Space.....	6-28
6-6	Examine Command Options.....	6-30
6-7	Device Name and Address Space.....	6-31
6-8	Test Command Options.....	6-55
A-1	Tru64 UNIX Boot Options	A-1
A-2	OpenVMS Boot Options.....	A-2
C-1	Number of Drives You Can Use in a Drive Group for Each RAID Level .	C-7
C-2	How the Capacity of Each Drive Affects the Capacity of the Drive Group	C-7
C-3	Host Adapter Parameters.....	C-13

Preface

Intended Audience

This manual is written for the system manager or system operator who has training in systems management and is running a *Compaq AlphaServer GS60E* system.

Document Structure

This manual uses a structured documentation design. Topics are organized into small sections for efficient reference. Each topic begins with an abstract. You can quickly gain a comprehensive overview by reading only the abstracts. Next is an illustration or example, which also provides quick reference. Last in the structure are descriptive text and syntax definitions.

This manual has six chapters and three appendixes, as follows:

- **Chapter 1, Introduction**, provides a brief overview of the GS60E hardware, firmware, and architecture.
- **Chapter 2, AlphaServer GS60E System**, gives a basic introduction to your system and its parts.
- **Chapter 3, I/O Subsystem**, describes the GS60E system's I/O design.
- **Chapter 4, Booting an Operating System**, tells how to start running the OpenVMS and Tru64 UNIX operating systems.
- **Chapter 5, System Troubleshooting**, provides basic troubleshooting procedures.
- **Chapter 6, SRM Console Commands**, lists the SRM console commands with an example of each command.
- **Appendix A, OpenVMS and Tru64 UNIX Boot Options**, lists the options used with the boot commands for OpenVMS and Tru64 UNIX to control various phases of booting.
- **Appendix B, Updating Firmware**, explains how to run the Loadable Firmware Update (LFU) Utility.
- **Appendix C, Running Configuration Utilities from the SRM Console**, explains how to run the configuration utilities required when installing some options.

Conventions Used in This Document

Icons. The icons shown below are used in illustrations for designating part placement in the system described. A shaded area in the icon shows the location of the component or part being discussed.

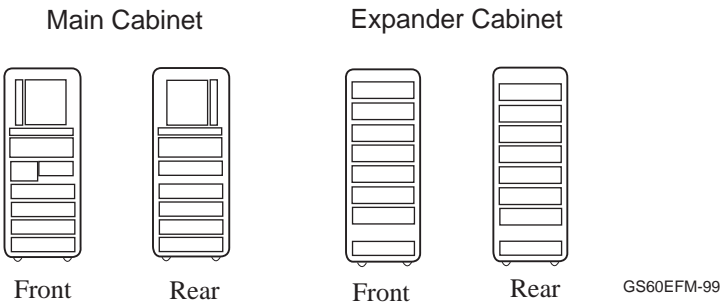


Table 1 AlphaServer GS60E Documentation

Title	Order Number
Hardware User Information and Installation	
<i>AlphaServer GS60E Installation Guide</i>	EK-GS60E-IN
<i>AlphaServer GS60E Operations Manual</i>	EK-GS60E-OP
<i>KFTHA System I/O Module Installation Guide</i>	EK-KFTHA-IN
<i>KFE72 Installation Guide</i>	EK-KFE72-IN
Service Information	
<i>AlphaServer GS60E Service Manual</i>	EK-GS60E-SV
Reference Manual	
<i>AlphaServer GS60E and GS140 Getting Started with Logical Partitions</i>	EK-TUNLP-SF
Upgrade Manuals	
<i>GS60/8200 to GS60E Upgrade Manual</i>	EK-GS60E-UP
<i>H7506 Power Supply Installation</i>	EK-H7506-IN
<i>RRDCD Installation Card</i>	EK-RRDXX-IN

Chapter 1

Installation Overview

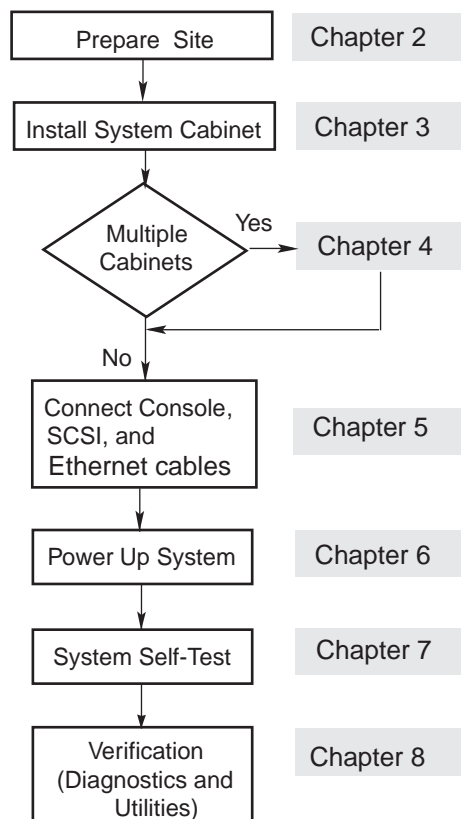
This chapter provides a quick overview of the installation procedure and relates the steps of the procedure with chapters in this manual. It consists of one section:

- Installation Flowchart

1.1 Installation Flowchart

In general, a system is installed as shown in Figure 1-1.

Figure 1-1 Installation Flowchart



GS60E11-99

In general, the procedure is as follows (Figure 1-1):

1. Prepare the site for system installation as described in Chapter 2.
2. Install the system cabinet as described in Chapter 3.
3. If the system has one or more expander cabinets, refer to Chapter 4 for installation instructions.
4. Connect the console, and, if appropriate, make the Ethernet, and SCSI connections as described in Chapter 5.
5. Power up the system as described in Chapter 6.
6. Observe and consider the results of the system self-test as described in Chapter 7.
7. Verify system operation by running various diagnostics and utilities as described in Chapter 8.

Chapter 2

AlphaServer GS60E System

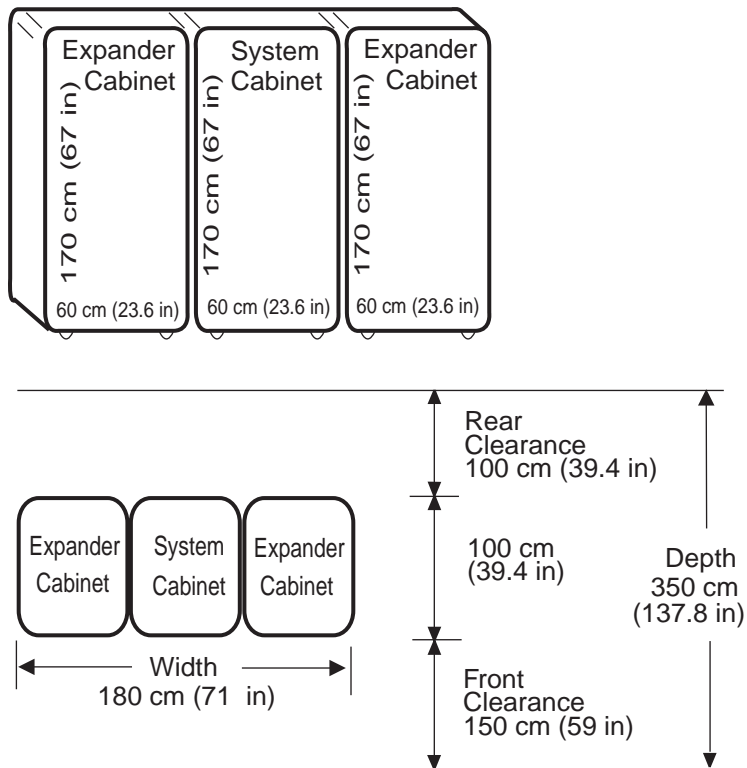
This chapter describes the system package and the location of components in the cabinet. Sections include:

- System Characteristics
- System Front View
- System Rear View
- System Components (TLSB, Cabinet Control Logic Panel, Console Load Device, and Power System)
- Controls and Indicators (Operator Control Panel, AC Power Circuit Breaker)

2.1 System Characteristics

Figure 2-1 shows the cabinet dimensions and the required clearance space. The tables list the electrical and environmental characteristics.

Figure 2-1 Sample System Footprint



OM21-99

The values in Table 2-1 and Table 2-2 apply to the system cabinet only. The values are configuration dependent.

Table 2-1 Electrical Characteristics

Electrical	Specification
Three-phase AC input voltage	120/208 Wye – North America 380-415 Wye – Europe/AP 202 Delta – Japan
Nominal frequency	50 – 60 Hz
AC current (maximum)	30A – North America 32A – Europe/AP 30A – Japan
AC power consumption (maximum)	2,450 watts

Table 2-2 Environmental Characteristics

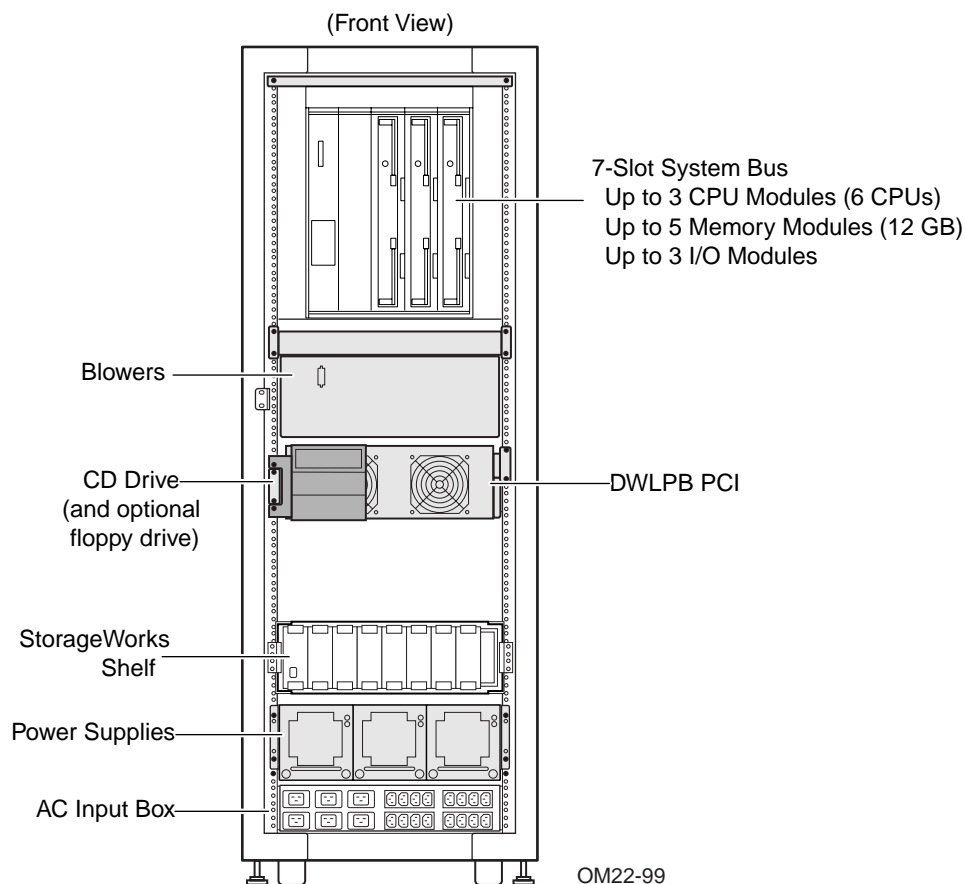
Environmental	Operating ¹	Storage
Heat dissipation (maximum)	8,300 BTU/hr	
Temperature	5°to 35°C (41°to 95°F) ¹	-40°to 66°C -40°to 151°F)
Relative humidity	10–90%	10–95%
Altitude	0–3 km (0–10,000 ft)	0–12.2 km (0–40,000 ft)

¹ Recommended operating temperature is 18°–24°C (65°–75°F) and 40–60% relative humidity.

2.2 System Front View

The system includes a console terminal and printer. It can have up to two optional expander cabinets. With the front door open, you can see the TLSB card cage, CD-ROM, blowers, PCI shelves, StorageWorks shelves, power supplies, and AC input box.

Figure 2-2 System Front View



Your Compaq customer service engineer has installed your system and verified that it is running properly. Before you turn on the system, familiarize yourself with its components:

- The **system cabinet** houses the power system, the TLSB card cage, operator control panel (located in the top center of the door), the cabinet control logic (CCL) panel, and a CD-ROM drive. Optional hardware includes a floppy drive in the CD unit, StorageWorks shelves, and PCI shelves.
- The **CD-ROM** is the console load device, used for installing operating systems and software. A second CD-ROM is required for logical partitioning.
- The **console device** is used for booting and for system management operations and must include a serial console monitor.
- The **console printer** provides a hardcopy record of system operations.
- **Optional I/O components** include PCI shelves and StorageWorks shelves. These shelves are installed in the system or expander cabinets to provide space for I/O and disk options.
- **Optional expander cabinets** (up to two) provide additional space for PCI I/O devices and disk drives. Each expander cabinet has two power supplies and provides space for a maximum of four DWLPBs and three BA36Rs or three DWLPBs and four BA36Rs.
- A system documentation kit.

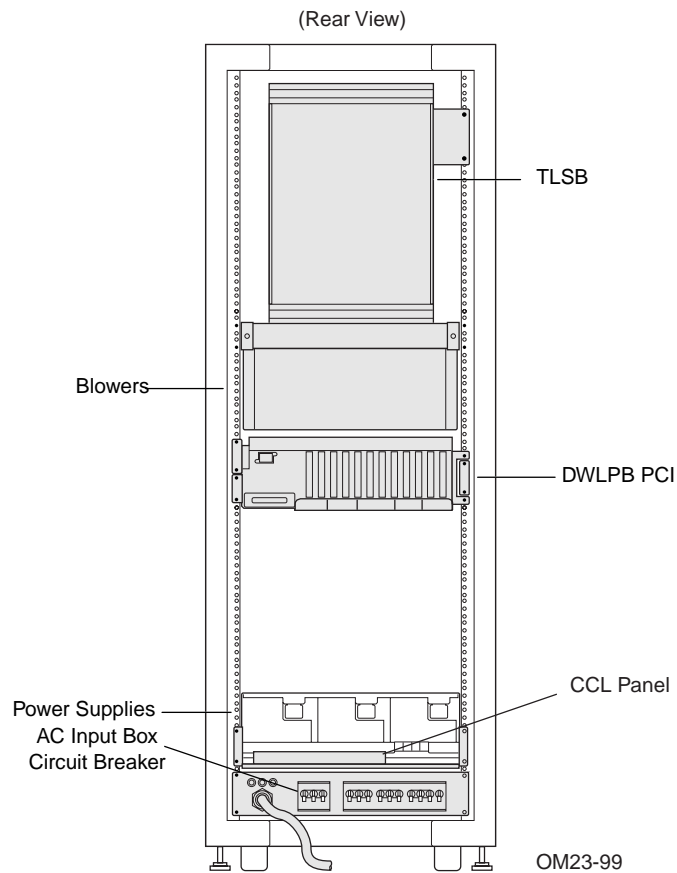
The following components are visible from the inside front of the cabinet:

- TLSB card cage
- CD-ROM drive
- Blowers
- PCI or StorageWorks shelves
- Power supplies
- AC input box

2.3 System Rear View

With the rear door open, Compaq customer service engineers can access the CCL panel, the AC input box circuit breaker, and the AC power cord.

Figure 2-3 System Rear View



The following components are visible from the inside rear of the cabinet:

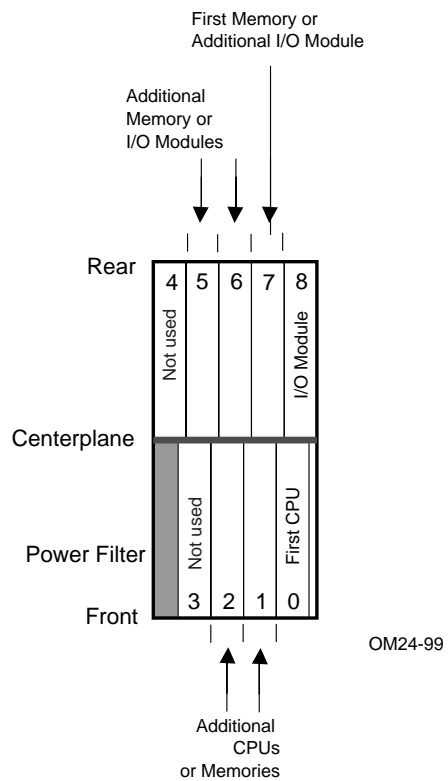
- TLSB card cage
- Blowers
- PCI or StorageWorks shelves
- Power Supplies
- CCL panel
- AC input box circuit breaker

2.4 System Components

2.4.1 TLSB Card Cage

The TLSB card cage is a 7-slot card cage that contains slots for up to three CPU modules, up to five memory array modules, and up to three I/O modules. The TLSB bus interconnects the CPU, memory, and I/O modules.

Figure 2-4 TLSB Card Cage



The TLSB card cage is located in the upper part of the system cabinet. The TLSB card cage contains seven module slots (slots 3 and 4 are not used). The slots are numbered 0 through 2 from right to left in the front of the cabinet and slots 5 through 8 right to left in the rear of the cabinet (see Figure 2-4). The minimum configuration is a processor module in slot 0, an I/O module in slot 8, a memory module in slot 7, and terminator modules in all other slots.

Module Placement Rules

Configure modules in this order:

1. Place the processor modules first. Start at slot 0 and work up to slot 2.
2. Place the KFTHA modules next. The first KFTHA module goes in slot 8, a second in slot 7, and a third in slot 6.
3. Place memory modules last. The first memory module goes in the highest numbered open slot, the next in the lowest numbered open slot, and so on, alternating between highest- and lowest-numbered open slots.
4. Fill all remaining open slots with terminator modules.

About the TLSB Card Cage

Modules used in this system are:

Terminator

1 Gbyte memory (MS7CC-EA)

2 Gbyte memory (MS7CC-FA)

4 Gbyte memory (MS7CC-GA)

KFTHA (4 hose cables)

Dual processor (KN7CG-AB)

The maximum number of processor modules is three.

The maximum number of memory modules is five. Memory modules may be placed in slots 1, 2, 5, 6, and 7 only. The maximum amount of memory is 12 Gbytes. All memory modules support two-way interleaving. Mixed sizes of memory modules may be installed in the TLSB card cage.

Each system must have a minimum of one KFTHA I/O module, installed in slot 8.

2.4.2 Cabinet Control Logic Panel

Console terminal I/O and expander remote power control/status connections are located on the cabinet control logic (CCL) panel. The CCL is accessible from the lower rear of the system, above the AC input box. Table 2-3 lists the other CCL panel connections.

Figure 2-5 Cabinet Control Logic Panel

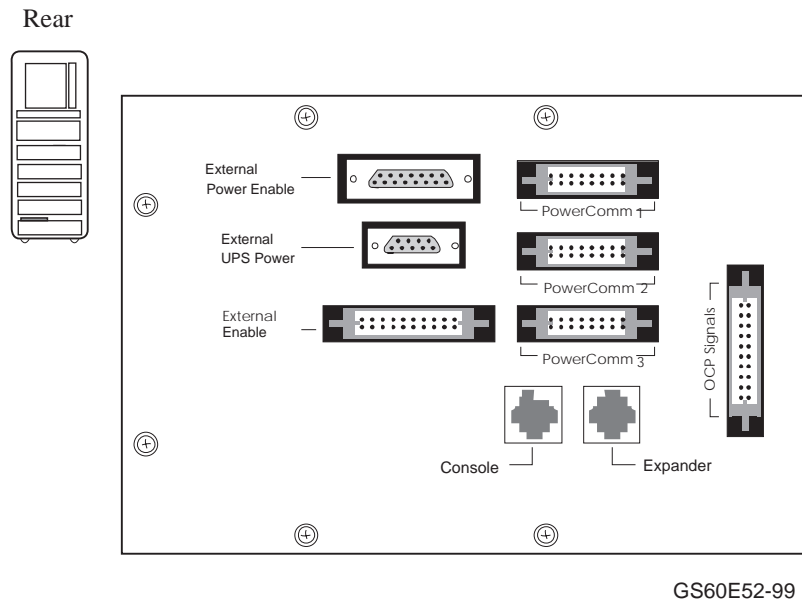


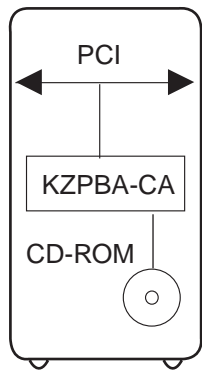
Table 2-3 Control/Status and I/O Connections

Connector Name	Function
Console	Serial console device connection for <i>OpenVMS</i> and <i>Tru64 UNIX</i> systems.
Expander	Expander cabinet power supply control cable. NOTE: The expander cabinet connector is not intended to be connected to a public telecommunications network.
Power Comm1	Power supply 1 signal and control connection.
Power Com2	Power supply 2 signal and control connection.
Power Comm3	Power supply 3 signal and control connection.
OCP Signals	Operator Control Panel signal cable.
External Power Enable	Enables power to PCI and StorageWorks shelves.
External UPS Power	Reserved for future use.
External Enable	Reserved for future use.

2.4.3 Console Load Device

The CD-ROM drive (see Figure 2-2) is the in-cabinet console load device.

Figure 2-6 Accessing the Console Load Device



GS60E81-99

The console load device is used for:

- Installing or updating software
- Loading a backup utility program
- Interchanging user data
- Updating module firmware

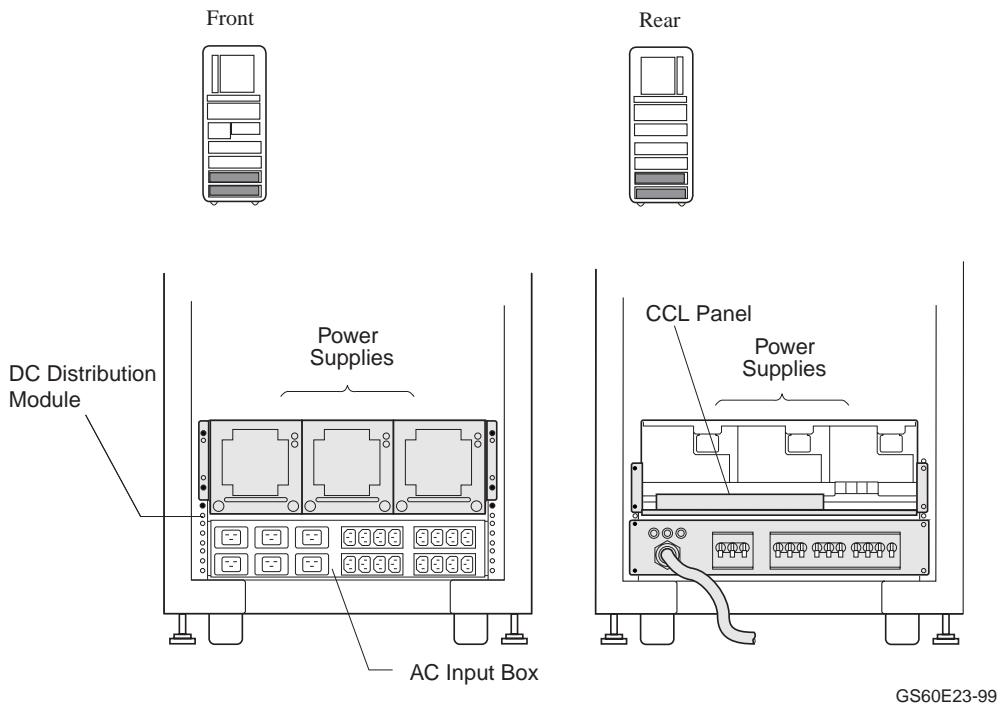
The CD-ROM drive is the console load device. It is installed in the system cabinet and is used to access software and online documentation. Access to the CD-ROM is provided through the PCI subsystem through a KFTHA module.

A second CD-ROM must be installed when logical partitioning is used (see *RRDCD Installation Card*, EK-RRDXX-IN and *AlphaServer GS60E and GS140 Getting Started with Logical Partitions*, EK-TUNLP-SF).

2.4.4 Power System

The power system includes an AC input box, a power subrack (containing a DC distribution module and power supplies), cabinet control logic panel, power distribution cables, and signal interconnect cables.

Figure 2-7 Power System



Three H7506 power supplies (see Figure 2-7) provide the necessary power and power redundancy required for all internal system components. The power supplies require three-phase AC.

The AC input box is located at the bottom of the system cabinet (when viewing the system cabinet from the rear). The 48 VDC power supplies are located above the AC input box and visible when viewing the system cabinet from the front.

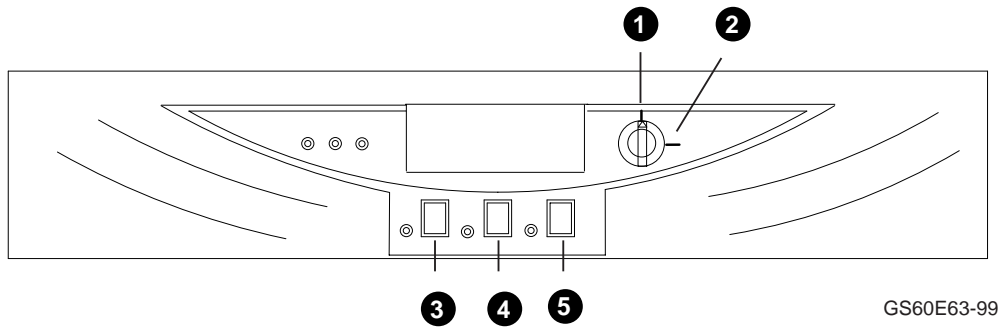
The AC input box provides the interface for the system to the AC utility power. The main input circuit breaker, on the AC input box, contains a circuit breaker trip indicator to indicate an open circuit breaker. The DC distribution module connects the AC input box and power supplies. It distributes the 48 VDC power. See Section 2.5.3.

2.5 Controls and Indicators

2.5.1 Operator Control Panel Keyswitch and Pushbuttons

The operator control panel (OCP), located in the upper center of the front of the cabinet, contains a keyswitch, pushbuttons, and indicator lights. The keyswitch regulates power going to the system.

Figure 2-8 Operator Control Panel Keyswitch and Pushbuttons



The OCP keyswitch and indicators are shown in Figure 2-8.

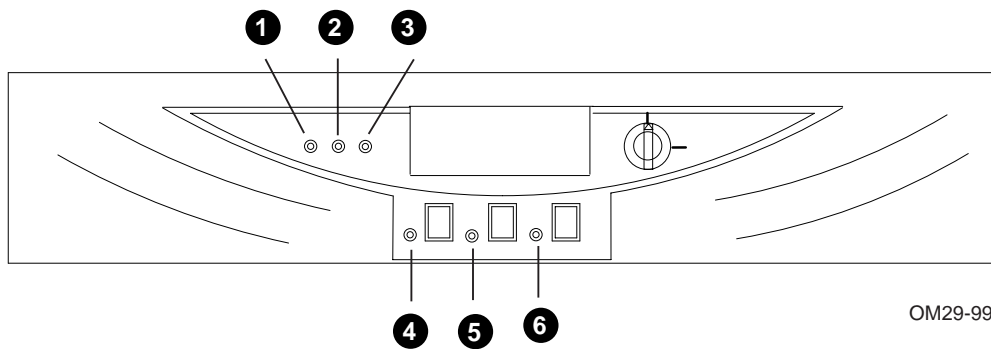
Table 2-4 Keyswitch and Pushbutton Positions

Position	Effect
❶ – Off	Removes 48 VDC power from the system. Power is still supplied to the CCL module. This switch is useful for field service when they wish to power down the system in an orderly way, prior to switching the power off completely while replacing or installing a new piece of hardware.
❷ – On	Position used while the machine executes programs. On <i>Tru64 UNIX</i> or <i>OpenVMS</i> systems, you can interrupt program execution and enter SRM console mode by typing Ctrl/P at the console device when the switch is in this position.
❸ – On/Off Button	With the keyswitch in the On position, this button will power the system on or off. With the keyswitch in the Off position, this button is disabled.
❹ – Secure Button	Pressing this button disables the primary console.
❺ – Reset Button	Pressing this button causes the system to reset.

2.5.2 Operator Control Panel Indicator Lights

The OCP has six status indicator lights: Run, Power, Fault, On Secure, and Reset. These lights indicate the operating status of the system.

Figure 2-9 Operator Control Panel LEDs



Six status indicator LEDs (see Figure 2-9) show the state of the system. Table 2-5 describes the conditions indicated by the lights.

Table 2-5 Operator Control Panel LEDs

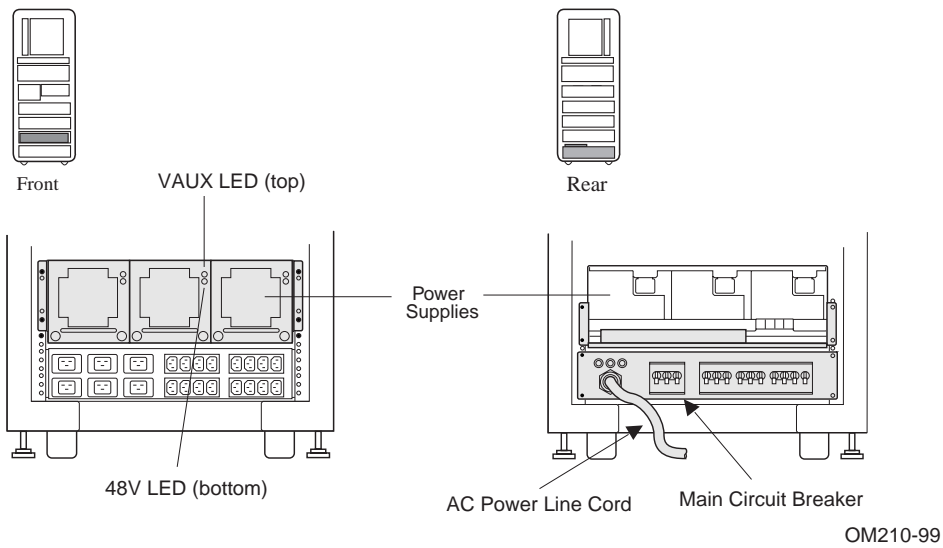
Light	Color	State	Meaning
❶ – Run	Green	On	Power is supplied to entire system; the blower is running. System has exited console.
❷ – Power	Green	On	System is powered on.
❸ – Fault	Yellow	On	Fault on system bus.
❹ – On	Green	On	Power is supplied to the whole system.
❺ – Secure	Green	On	Indicates input from the console device is prevented.
❻ – Reset	Yellow	On	Indicates a system reset has occurred, clearing captured error information.

NOTE: *With the keyswitch in the On position, if all six LEDs are blinking, one or more of the power supplies has failed or there is a missing power supply. With the keyswitch in the Off position, the LEDs will also blink but do not provide power supply status.*

2.5.3 Circuit Breaker and AC Power Indicators

The main circuit breaker is located on the left side of the AC input box, at the bottom rear of the system cabinet. The AC input box and power supplies have indicator LEDs.

Figure 2-10 Circuit Breaker and AC Power Indicators



The circuit breaker and power indicators are at the rear of the cabinet.

Circuit Breaker

The circuit breaker controls power to the entire system, including the power supplies, blowers, and in-cabinet options. Current overload causes the breaker to trip to the Off position, so that power to the system is turned off.

For normal operation, the circuit breaker must be in the On position, with the handle pushed up. To shut the circuit breaker off, push the handle down.

AC Power Indicators

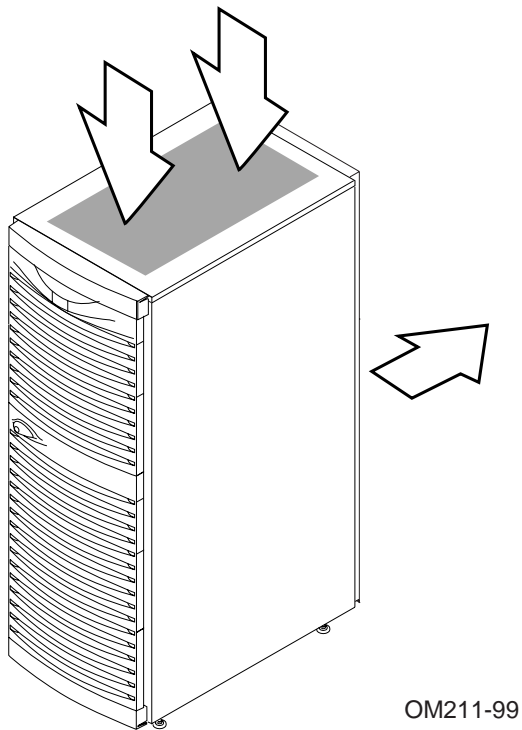
Three lights above the AC power line cord (see Figure 2-10) indicate that AC power is supplied to the the power distribution module.

The power supplies have two LEDs that indicate normal conditions and faults. When the system (keyswitch) is off, plugged in, and the circuit breakers are on, power is present only within the AC box and power supplies. The green VAUX LEDs on the power supplies should be illuminated. When the system is on, the VAUX and 48V LEDs should light.

2.5.4 Cooling System

The cooling system cools the power system, the TLSB card cage, and shelves.

Figure 2-11 Cabinet Airflow



The cooling system is designed to keep system components at an optimal operating temperature. It is important to keep the front and rear doors free of obstructions, leaving a minimum clearance space of 1.5 meters (59 inches) in the front and 1 meter (39 inches) in the rear between cabinets (see Figure 2-1) to maximize airflow.

Two blowers, located in the center of the cabinet, draw air downward through TLSB card cage. Air is exhausted at the middle of the cabinet, to the rear. The blower speed varies based on the system's ambient temperature.

CAUTION: Anything placed on top of the cabinet could restrict airflow. This will cause the system to power down.

Chapter 3

I/O Subsystem

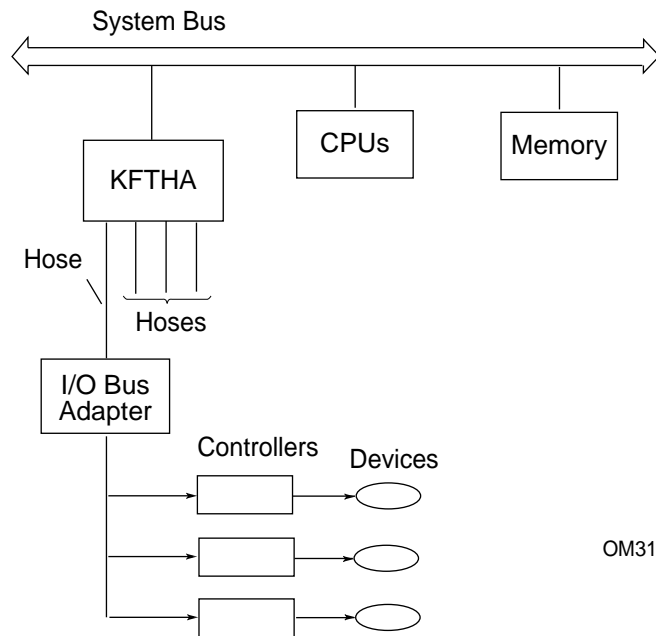
This chapter describes the *AlphaServer* GS60E I/O subsystem. Sections include:

- I/O Subsystem and Module Overview
- System Configuration Information
- DWLPB PCI and KFE72 Adapters

3.1 I/O Subsystem and Module Overview

Figure 3-1 illustrates CPU, memory, and I/O port module (KFTHA) interfaces to the system bus. The KFTHA has four channels or “hoses” to the PCI subsystem.

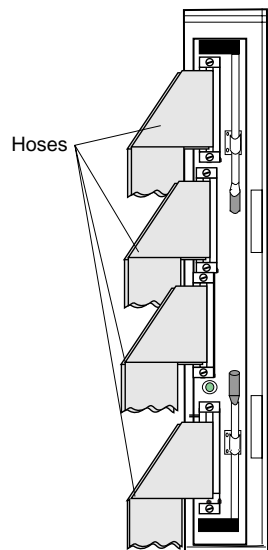
Figure 3-1 I/O Subsystem



OM31-99

The interface from the *AlphaServer GS60E* system bus to I/O is provided by the KFTHA I/O adapter module, which provides high-speed, high-volume data transfers. The KFTHA has four channels (called hoses) connecting to external I/O buses (see Figure 3-2).

Figure 3-2 KFTHA I/O Port Module



OM32-99

3.2 System Configuration Information

Basic information on the system and I/O subsystem configuration is displayed on power-up. Example 3-1 shows an *AlphaServer GS60E* system self-test display.

Example 3-1 System Self-Test Display

```

F   E   D   C   B   A   9   8   7   6   5   4   3   2   1   0   NODE #
                                A   M   M   M   .   .   P   P   P   TYP
                                o   +   +   +   .   .   ++  ++  ++  ST1  ❶
                                .   .   .   .   .   .   EE  EE  EB  BPD
                                o   +   +   +   .   .   ++  ++  ++  ST2
                                .   .   .   .   .   .   EE  EE  EB  BPD
                                o   +   +   +   .   .   ++  ++  ++  ST3
                                .   .   .   .   .   .   EE  EE  EB  BPD

                                +   +   +   +   +   +   .   .   .   .   +   C0 PCI + ❷
                                .   .   .   .   .   .   .   .   .   .   EISA +
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C1
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C2
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C3

                                B0  A1  A0  .   .   .   .   .   ILV
                                .  4GB 4GB 4GB .   .   .   .   12GB
Compaq AlphaServer GS60E 6-6/525/4, Console V5.5 4-MAY-1999 12:06:02
SR0M V2.1, OpenVMS PALcode V1.57-2, Tru64 UNIX PALcode V1.50-1
P00>>>

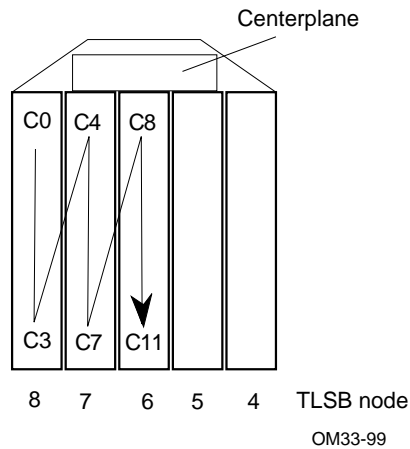
```

On power-up, the console displays the self-test results. (Chapter 5 describes the system self-test in detail.) The GS60E system shown in Example 3-1 has dual-CPU modules in nodes (slots) 0, 1, and 2 of the TSLB card cage, three 4-GB memory modules in nodes 5, 6, and 7, and a KFTHA I/O adapter in node 8 ❶.

The KFTHA adapter has one of its four connectors used. The first hose, C0 ❷, is connected to a PCI that has devices in “slots” 0 and slots 5 through 11.

Figure 3-3 shows the connector numbering scheme for the KFTHA module. Each slot has four connector numbers associated with it, numbered in increasing order from top to bottom, as shown.

Figure 3-3 Hose Numbering Scheme for KFTHA



Example 3-2 Sample Show Configuration Command

P00>>> sho conf

	①	Name	②	③	Type	④	Rev	⑤	Mnemonic
		TLSEB							
0++		KN7CG-AB		8025		0000		kn7cg-ab0	
1++		KN7CG-AB		8025		0000		kn7cg-ab1	
5+		MS7CC		5000		0000		ms7cc0	
6+		MS7CC		5000		0000		ms7cc1	
7+		MS7CC		5000		0000		ms7cc2	
8+		KFTHA		2000		0000		kftha0	
C0 PCI connected to kftha0									⑥
0+		SIO		4828086		0004		sio0	
5+		QLogic ISP1040B		10201077		0005		isp0	
6+		QLogic ISP1040B		10201077		0005		isp1	
7+		QLogic ISP1040B		10201077		0005		isp2	
8+		KZPBA		11011		0002		kzpbao	
9+		QLogic ISP1040B		10201077		0005		isp3	
A+		QLogic ISP1040B		10201077		0005		isp4	
B+		DE500-BA		191011		0030		tulipl	
Controllers on SIO									⑦
0+		DECchip 21040-AA		21011		0023		tulip0	
1+		FLOPPY		2		0000		floppy0	
2+		KBD		3		0000		kbd0	
3+		MOUSE		4		0000		mouse0	
6+		TOY		7		0000		toy0	
EISA connected to pci0 through sio0									eisa0

The SRM console command **show configuration** displays system configuration information detailing the I/O adapter connected to your system and its status. The information is displayed in five columns:

- ❶ module slot number
- ❷ module name
- ❸ module type
- ❹ module revision
- ❺ module mnemonic

TLSB information is shown first. There are six modules in the TLSB card cage:

- Slot 0 and 1 - dual-processor CPU module
- Slot 5, 6, and 7 - MS7CC memory module
- Slot 8 - KFTHA I/O port module

When there are multiple modules of a single type, the mnemonics are numbered consecutively (for example; isp0, isp1, isp2, and isp3).

The KFTHA module in TLSB slot 8 is displayed first. Hose C0 ❹ is connected to a PCI adapter containing a KFE72 adapter. There are eight modules in the PCI adapter:

- Slots 0 – SIO (standard I/O module) of the KFE72 adapter (Note that the connector module is not shown.)
- Slots 5, 6, and 7 – three ISP1040Bs (isp0, isp1, and isp2) for FWD (fast wide differential) SCSI
- Slot 8 – KZPBA (PCI UltraSCSI adapter)
- Slots 9 and A -- two ISP1040Bs (isp3 and isp4) for SE (single-ended) SCSI and FWD SCSI
- Slot B – PCI fast Ethernet (tulip1)

Any PCI bus containing the standard I/O module (SIO) will show the information shown in ❷. This information is not meaningful, however. The keyboard and mouse connectors on a KFE72 adapter occupy slot 2 of the PCI card cage. Slot 1 contains two serial ports, and slot 3 is allocated for a parallel port assembly.

Example 3-3 Sample Show Device Command

P00>>> sho dev

```
polling for units on floppy0, slot 0, bus 1, hose4...
dva0.0.0.1100.4   DVA0           RX26/RX23
polling for units on kzpba0, slot 5, bus 0, hose4...
pka.7.0.5.4      pka             Bus ID 7
dka400.4.0.5.4   DKA400         RRD47 1206

polling for units on isp0, slot 6, bus 0, hose4...
pkb.7.0.6.4      pkb             term on   Bus ID 7 5.57
dkb0.0.0.6.4     DKB0           RZ1CB-CS 0844
dkb100.1.0.6.4   DKB100         RZ1CB-CS 0844
dkb200.2.0.6.4   DKB200         RZ1CB-CS 0844
dkb300.3.0.6.4   DKB300         RZ1CB-CS 0844
dkb400.4.0.6.4   DKB400         RZ1CB-CS 0844
dkb500.5.0.6.4   DKB500         RZ1CB-CS 0844
dkb600.6.0.6.4   DKB600         RZ1CB-CS 0844

polling for units on ispl, slot 7, bus 0, hose4...
pkc.7.0.7.4      pkc             term on   Bus ID 7 5.57
polling for units on isp2, slot 8, bus 0, hose4...
pkd.7.0.8.4      pkd             term on   Bus ID 7 5.57
polling for units on isp3, slot 9, bus 0, hose4...
pke.7.0.9.4      pke             term on   Bus ID 7 5.57
polling for units on isp4, slot 10, bus 0, hose4...
pkf.7.0.10.4     pkf             term on   Bus ID 7 5.57
P00>>>
```


The SRM command **show device** is useful for locating the boot device for the Loadable Firmware Utility, or the boot device for the operating system for *OpenVMS* or *Tru64 UNIX* systems. The **show device** command is also helpful in isolating non-functioning I/O devices detected by a **test** command.

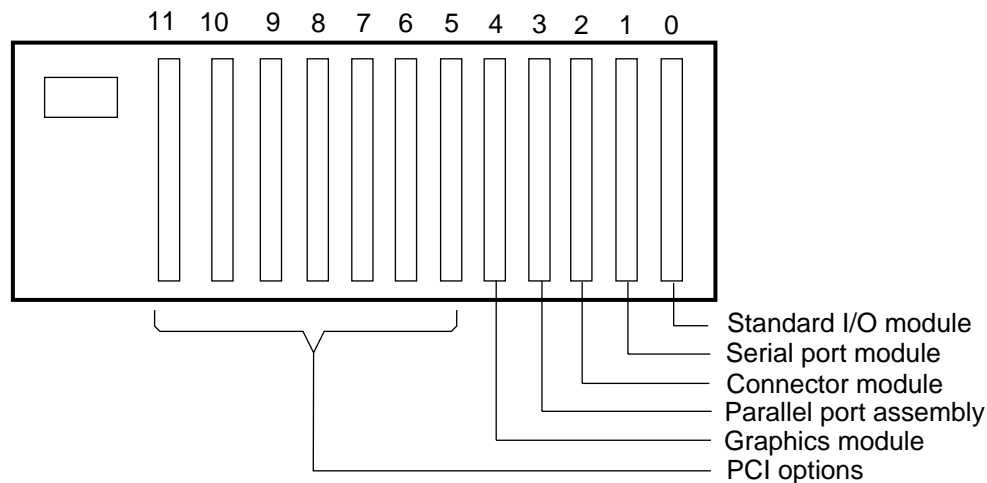
The **show device** command provides the following device information:

- ❶ Device mnemonic
- ❷ Slot number
- ❸ Bus number
- ❹ Hose number

3.3 DWLPB PCI and KFE72 Adapters

The DWLPB PCI adapter provides a complete PCI bus subsystem for use with the *AlphaServer GS60E* system. The KFE72 adapter provides I/O output needed for systems using a graphics device and must connect to a DWLPB adapter. It also provides support for logical partitions.

Figure 3-4 PCI Slot Configuration Used with the KFE72



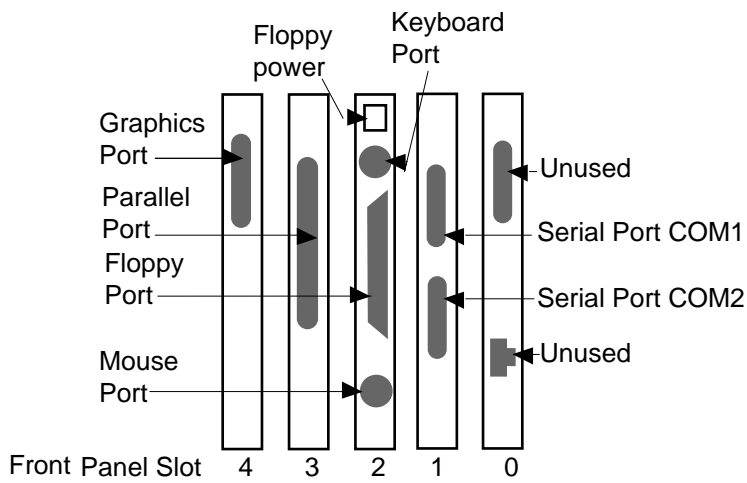
OM34-99

If a PCI shelf does not have a KFE72, it contains 12 slots for PCI adapters, numbered right-to-left from 0 to 11.

The KZPBA-CA adapter is needed for CD-ROM support.

The KFE72 consists of four modules and is supported by a DWLPB. Figure 3-4 shows that the KFE72 uses slots 0-4 for modules. Slots 5-11 are available for PCI devices.

Figure 3-5 KFE72 and Graphics Modules



OM35-99

A KFE72 may contain up to four modules and a parallel port assembly. The four modules are the standard I/O module (part number B2110-AA), a connector module (part number 54-25133-01) supplying floppy, mouse, and keyboard ports, a serial port module (part number 54-25082-01) supplying two serial ports, and the PowerStorm 3D30 graphics module (part number 54-23481-01). The parallel port assembly is cabled to the connector module and occupies a slot position as shown in Figure 3-5.

Chapter 4

Booting an Operating System

This chapter describes how to boot the *OpenVMS* and *Compaq Tru64 UNIX* operating systems. Sections include:

- Preparation
- Selecting a Boot Device
- Booting *OpenVMS*
- Booting *Tru64 UNIX*

4.1 Preparation

There are some steps you must take preparatory to booting and other steps that can make booting thereafter easier.

4.1.1 Set os_type Environment Variable

For factory-installed software (FIS), a default operating system is defined. If you do not have factory-installed software, or you wish to change the default operating system, you use the set os_type command.

Example 4-1 Setting os_type for OpenVMS

```
P00>>> set os_type vms
```

Example 4-2 Setting os_type for Tru64 UNIX

```
P00>>> set os_type unix
```

These examples show the SRM command used to define the operating system as *OpenVMS* and *Tru64 UNIX*, respectively.

4.1.2 Set console Environment Variable

For OpenVMS systems and Tru64 UNIX systems, the console environment variable should be set to serial (the default).

Example 4-3 Set Console to Serial for Tru64 UNIX and OpenVMS Systems

```
P00>>> set console serial
```

The **set console** command is needed for systems having a graphics monitor as part of their console interface. The default setting for the **console** environment variable is **serial**.

By specifying **set console graphics** or **set console both** the SRM console will build the proper parameters in a data structure called the HWRPB, or Hardware Restart Parameter Block, so that an operating system will build with graphics enabled.

Both operating systems require a serial console terminal

4.1.3 Set auto_action Environment Variable

You can direct the SRM console to automatically boot the operating system upon system power-up, Restart, or init. The default is to halt the system after system self-test and leave the console device in SRM console mode.

Example 4-4 Setting the auto_action Environment Variable

```
P00>>> show auto_action
auto_action    halt
P00>>> set auto_action boot
```


The **auto_action** environment variable is set to **halt** by default. In this case, at system restart (power-up, reset, or **init** command), the SRM console software will display the system self-test and display the SRM console prompt to accept commands.

If you set the **auto_action** environment variable to **boot**, at system restart, the SRM console software will display the system self-test and then automatically boot the operating system according to the default boot parameters you defined.

For *OpenVMS* and *Tru64 UNIX* systems, you can return to the SRM console once the system has booted by typing Ctrl/P. This halts the operating system on the boot processor and displays the SRM console prompt.

4.2 Selecting a Boot Device

The OpenVMS and Tru64 UNIX operating systems can be booted from a number of devices: the CD-ROM drive, a local system disk, a disk connected to the system through a CIPCA adapter, or by Ethernet from a remote disk on another system. Certain operating systems use a subset of these devices; check to see what is supported. Note that factory-installed software has a default boot device selected.

Table 4-1 Boot Devices

Device	Location
CD-ROM	In-cabinet compact disk drive, used for booting the Loadable Firmware Update (LFU) Utility. See Appendix B for <i>OpenVMS</i> and <i>Tru64 UNIX</i> systems.
Local device	Disk connected to the system through an adapter on an I/O bus.
CI disk	Disk located on the system's HSC controller connected to the system by a CIPCA adapter on the PCI bus.
Remote disk	Disk connected to another system on the Ethernet, through the Ethernet port interface or the adapter.

For systems with factory-installed software (FIS), a default boot device (*OpenVMS* or *Tru64 UNIX*) has already been selected for you. If you want to change this default, follow the instructions here.

NOTE: *The default boot device or boot path selected applies to the operating system. If you wish to boot the Loadable Firmware Update (LFU) utility, you must specify the CD-ROM device on your system, if you use the SRM console **boot** command.*

For *OpenVMS* and *Tru64 UNIX* systems, you can set or change the default boot device with the **set bootdef dev** command, as described in Sections 4.2.1 and 4.2.2. You can also specify a particular device in the SRM console command **boot**.

See Table 4-1 for the types of devices available for your operating system for booting.

4.2.1 Show Config and Show Device Commands (for Booting OpenVMS and Tru64 UNIX Systems)

If you do not wish to use the default boot device selected for factory-installed OpenVMS or Tru64 UNIX systems, the show config and show device SRM console commands display your system configuration, allowing you to identify and select an appropriate boot device.

Example 4-5 Show Config and Show Device Commands

```
P00>>> sho conf
```

```
      Name                Type   Rev  Mnemonic
-----
TLSE
0++  KN7CG-AB             8025  0000  kn7cg-ab0
1++  KN7CG-AB             8025  0000  kn7cg-ab1
5+   MS7CC                 5000  0000  ms7cc0
6+   MS7CC                 5000  0000  ms7cc1
7+   MS7CC                 5000  0000  ms7cc2
8+   KFTHA                 2000  0000  kftha0

C0 PCI connected to kftha0
0+   SIO                   4828086  0004  sio0
5+   QLogic ISP1040B      10201077  0005  isp0
6+   QLogic ISP1040B      10201077  0005  isp1
7+   QLogic ISP1040B      10201077  0005  isp2
8+   KZPBA                 11011   0002  kzpba0
9+   QLogic ISP1040B      10201077  0005  isp3
A+   QLogic ISP1040B      10201077  0005  isp4
B+   DE500-BA             191011   0030  tulip1

Controllers on SIO
0+   DECchip 21040-AA      21011   0023  tulip0
1+   FLOPPY                 2   0000  floppy0
2+   KBD                     3   0000  kbd0
3+   MOUSE                   4   0000  mouse0
6+   TOY                      7   0000  toy0

EISA connected to pci0 through sio0  eisa0
```

Example 4-5 Show Config and Show Device Commands (Continued)

```
P00>>> sho dev
polling for units on floppy0, slot 0, bus 1, hose4...
dva0.0.0.1100.4   DVA0                               RX26/RX23
polling for units on kzpba0, slot 5, bus 0, hose4...
pka.7.0.5.4      pka                               Bus ID 7
dka400.4.0.5.4   DKA400                               RRD47 1206 ❶
polling for units on isp0, slot 6, bus 0, hose4...
pkb.7.0.6.4      pkb      term on      Bus ID 7 5.57
dkb0.0.0.6.4     DKB0      RZ1CB-CS 0844
dkb100.1.0.6.4   DKB100   RZ1CB-CS 0844
dkb200.2.0.6.4   DKB200   RZ1CB-CS 0844
dkb300.3.0.6.4   DKB300   RZ1CB-CS 0844
dkb400.4.0.6.4   DKB400   RZ1CB-CS 0844
dkb500.5.0.6.4   DKB500   RZ1CB-CS 0844
dkb600.6.0.6.4   DKB600   RZ1CB-CS 0844
polling for units on ispl, slot 7, bus 0, hose4...
pkc.7.0.7.4      pkc      term on      Bus ID 7 5.57
polling for units on isp2, slot 8, bus 0, hose4...
pkd.7.0.8.4      pkd      term on      Bus ID 7 5.57
polling for units on isp3, slot 9, bus 0, hose4...
pke.7.0.9.4      pke      term on      Bus ID 7 5.57
polling for units on isp4, slot 10, bus 0, hose4...
pkf.7.0.10.4     pkf      term on      Bus ID 7 5.57
P00>>>
```

- ❶ The RRD47 is a CD-ROM disk drive, where you load firmware update CD-ROM disks. You can use the **boot** command to boot LFU from the disk. In this example, the command would be:

```
P08>>> boot dka400
```

If you are not planning to use the default boot device selected for your factory-installed software (FIS), or if your operating system is not factory installed, you select an appropriate boot device for your system. Use the **show config** and **show device** SRM console commands to display all the I/O adapters and devices on your system. The format for SRM console device names is:

ddau.n.c.s.h

where the fields are as defined in Table 4-2.

Table 4-2 SRM Console Device Naming Conventions

Field	Meaning	Description
dd	Device driver	Two-letter designator for a port or class device driver. Usually one of: dk SCSI disk fw FDDI network dr RAID set disk mk SCSI tape du DSSI disk mu DSSI tape dv Floppy disk pk SCSI class ew Ethernet network pu DSSI class
a	Adapter ID	Specifies the one-letter designator for adapter ID. One of a, b, c,...
u	Unit number	Specifies the device unit number. For Compaq MSCP devices, this is a unique monotonically increasing number. For SCSI devices, this is 100 times the bus node number, the next designator.
n	Node number	Specifies the bus node ID. Only meaningful for storage devices.
c	Channel number	Specifies the channel or PCI function number. Only meaningful for multi-channel or multi-function devices.
s	Slot number	Specifies the device's logical slot number. The output of a show config command indicates the logical slot numbers for devices in a system.
h	Hose number	Specifies the device's hose number. The output of a show config command shows the KFTHA module on the TLSB of your system.

Ideally, your system manager labeled the I/O devices on your system, so that you can identify a particular unit by its name. If not, you will have to follow cables back to the device adapter and then back to the hose to identify a particular unit.

In either case, select a device on your system that is of the type appropriate for the media your operating system software is on. It is useful to reserve one device on your system for this use, for consistency. This is done by setting the **bootdef_dev** environment variable, as described next.

4.2.2 Boot Environment Variables (for OpenVMS and Tru64 UNIX)

With OpenVMS or Tru64 UNIX systems, you can change the factory-installed default boot device, or, if your operating system is not factory-installed, define a default boot device, to be used automatically if no device is specified in the boot command itself.

Example 4-6 Viewing and Setting Boot Environment Variables (for OpenVMS and Tru64 UNIX)

```
P00>>> show boot* ❶
boot_dev
boot_file
boot_osflags          0
boot_reset            OFF
bootdef_dev
booted_dev
booted_file
booted_osflags
P00>>> set bootdef_dev dkb100 ❷
P00>>> show boot*
boot_dev
boot_file
boot_osflags          0
boot_reset            OFF
bootdef_dev           dkb100.1.0.6.1
booted_dev
booted_file
booted_osflags
```

For *OpenVMS* and *Tru64 UNIX* systems, some boot environment variables can be set by the user using the **set** SRM console command. Others are set by values you may specify in a **boot** command.

- ❶ The **show boot*** command displays the current settings of the boot environment variables. These are described in detail following.
- ❷ In Example 4-6, the user sets the default boot device environment variable **bootdef_dev** to device dkb100.

4.3 Booting OpenVMS

Check the environment variables for default boot parameters and make any changes needed. Issue the boot command.

Example 4-7 OpenVMS Boot

```
P00>>> show boot* ❶
boot_dev
boot_file
boot_osflags      0
boot_reset       OFF
bootdef_dev
booted_dev
booted_file
booted_osflags
P00>>> set bootdef_dev dkb100 ❷
P00>>> show boot*
boot_dev
boot_file
boot_osflags      0
boot_reset       OFF
bootdef_dev       dkb100.1.0.6.1
booted_dev
booted_file
booted_osflags

P00>>> b ❸
(boot dkb100.1.0.6.1 -flags 0)
SRM boot identifier: scsi 1 6 0 1 100 ef00 81011
boot adapter: kzpsa0 rev 0 in bus slot 6 off of kftha0 in TLSB slot 8
block 0 of dkb100.1.0.6.1 is a valid boot block
reading 904 blocks from dkb100.1.0.6.1
bootstrap code read in
Building FRU table.....
FRU table size = 0x2ff8
base = 200000, image_start = 0, image_bytes = 71000
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
```

Continued on next page

Example 4-7 OpenVMS Boot (Continued)

```
6-AUG-1999 11:36:18.27
OpenVMS (TM) Alpha Operating System, Version V7.1-2    ④
  Configuring devices...
%EWA0, Twisted-Pair(10baseT) mode set by console
:
*****
  You can install or upgrade the OpenVMS Alpha operating system
  or you can install or upgrade layered products that are included
  on the OpenVMS Alpha operating system CD-ROM.

  You can also execute DCL commands and procedures to perform
  "standalone" tasks, such as backing up the system disk.

Please choose one of the following:
  1) Upgrade, install or reconfigure OpenVMS Alpha Version V7.1-2
  2) Display products and patches that this procedure can install
  3) Install or upgrade layered products and patches
  4) Show installed products
  5) Reconfigure installed products
  6) Remove installed products
  7) Execute DCL commands and procedures
  8) Shut down this system

Enter CHOICE or ? for help: (1/2/3/4/5/6/7/8/?)
```

- ① The **show boot*** command shows the values of all the environment variables beginning with the characters "boot."
- ② Noting that no default boot device has been set, the user in this example issues the command **set bootdef_dev** to the desired disk. This disk will be used in subsequent boot commands that specify no device.
- ③ The boot command is issued and the SRM console firmware begins the process of booting the operating system. The SRM console displays information regarding the boot device and proceeds to load the bootstrap code. Once this process is finished, the SRM console transfers control to the bootstrap code.
- ④ The date and time and operating system banner display.

Example 4-7 shows that several choices can be made when booting *OpenVMS* in this manner. Make the appropriate choice and continue. (Note that by choosing 7, *Execute DCL commands and procedures*, you can back up your system disk. Standalone backup is replaced by this method of booting *OpenVMS*.)

4.4 Booting Tru64 UNIX

Use the show device command to locate the CD-ROM from which to boot the operating system. Issue the boot command.

Example 4-8 Tru64 UNIX Boot

```
P00>>> sho dev ❶
polling for units on floppy0, slot 0, bus 1, hose4...
dva0.0.0.1100.0   DVA0                RX26/RX23
polling for units on kzpba0, slot 5, bus 0, hose4...
pka.7.0.5.4      pka                  Bus ID 7
dka400.4.0.5.4   DKA400              RRD47 1206
polling for units on isp0, slot 6, bus 0, hose4...
pkb.7.0.6.4      pkb                  term on   Bus ID 7 5.57
dkb0.0.0.6.4     DKB0                RZ1CB-CS 0844
dkb100.1.0.6.4   DKB100             RZ1CB-CS 0844
dkb200.2.0.6.4   DKB200             RZ1CB-CS 0844
dkb300.3.0.6.4   DKB300             RZ1CB-CS 0844
dkb400.4.0.6.4   DKB400             RZ1CB-CS 0844
dkb500.5.0.6.4   DKB500             RZ1CB-CS 0844
dkb600.6.0.6.4   DKB600             RZ1CB-CS 0844
polling for units on ispl, slot 7, bus 0, hose4...
pkc.7.0.7.4      pkc                  term on   Bus ID 7 5.57
polling for units on isp2, slot 8, bus 0, hose4...
pkd.7.0.8.4      pkd                  term on   Bus ID 7 5.57
polling for units on isp3, slot 9, bus 0, hose4...
pke.7.0.9.4      pke                  term on   Bus ID 7 5.57
polling for units on isp4, slot 10, bus 0, hose4...
pkf.7.0.10.4     pkf                  term on   Bus ID 7 5.57
P00>>>

P02>>>boot dka400 ❷
(boot dka400.4.0.5.4 -flags A) ❸
SRM boot identifier: scsi 4 5 0 4 400 ef00 11000
boot adapter:kzpba0 rev 2 in bus slot 5 off of kfthal in TLSB slot 7
block 0 of dka400.4.0.5.4 is a valid boot block
reading 16 blocks from dka400.4.0.5.4
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 2000
initializing HWRPB at 2000
GCT base = 20012e000
:
:
setting affinity to the primary CPU
:
jumping to bootstrap code

Digital UNIX boot - Mon Aug 16 12:39:50 EDT 1999
```

```
Loading vmunix ...
[I/O subsystem reset information, memory information displayed, I/O
  bus adapters displayed, configured devices displayed,
network
  configuration information, partitioning information
displayed]
The system is ready.
```

The Installation Guide contains more information about installing DIGITAL UNIX.

- 1) Default Installation
- 2) Custom Installation
- 3) UNIX Shell

Enter your choice:

4

- ① **Show device** displays information about each I/O device. Polling checks the DWLPB for device configurations. The next line contains four columns. The first column contains the device type and unit number, node number, device channel number, the internal PCI node number, and I/O channel number, separated by periods. The second column displays the name of the device given by the device controller. The third column shows the device type, and the fourth column shows the revision of the device. The CD drive in this case is the RRD47 and its mnemonic is DKD400.
- ② In the **boot** command, **dk** is the device code of the boot device, **a** is the boot device controller designation, and **400** specifies the hexadecimal unit number of the boot device. The **4** is the node number, **0** is the channel number, **5** is the PCI node number, and **4** is the I/O channel number. The **-flags A** causes UNIX to come up in multiuser mode. Type the **boot** command.
- ③ The system boots from the CD-ROM.
- ④ Select the type of installation desired and follow the instructions after which the operating system banner appears and the user is given the **login:** prompt.

Chapter 5

System Troubleshooting

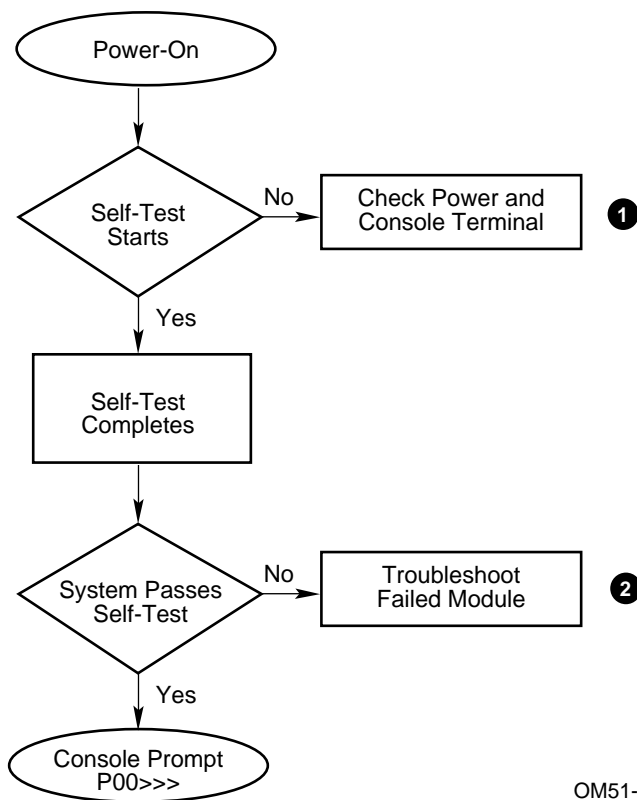
This chapter describes system problems and their symptoms during power-up, booting, and normal operation. Advanced troubleshooting is discussed in the *Service Manual*. Topics discussed include:

- Troubleshooting During Power-Up
- Troubleshooting During Booting
- Troubleshooting a PCI Shelf
- Troubleshooting a StorageWorks Shelf
- Self-Test Overview
- Sample Self-Test Display
- Self-Test Lines NODE # and TYP
- Self-Test Lines ST and BPD
- Self-Test Lines C0, C1, C2, ... Cn
- Self-Test Lines ILV and GB
- Self-Test Identification Line
- Show Commands
- Test Command
- Error Reports

5.1 Troubleshooting During Power-Up

Figure 5-1 shows the power-up sequence and steps to take if a problem occurs.

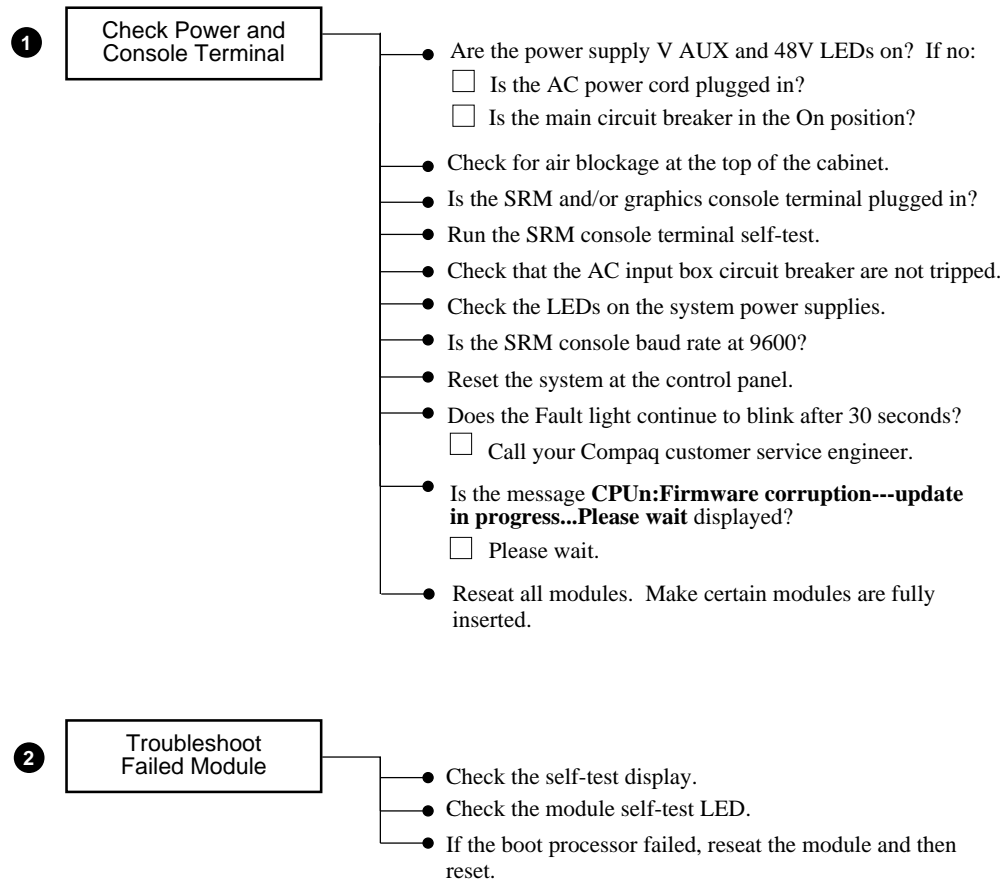
Figure 5-1 Power-Up Troubleshooting Flowchart



OM51-99

Troubleshooting steps during power-up are described in Figure 5-2:

Figure 5-2 Power-Up Troubleshooting Steps

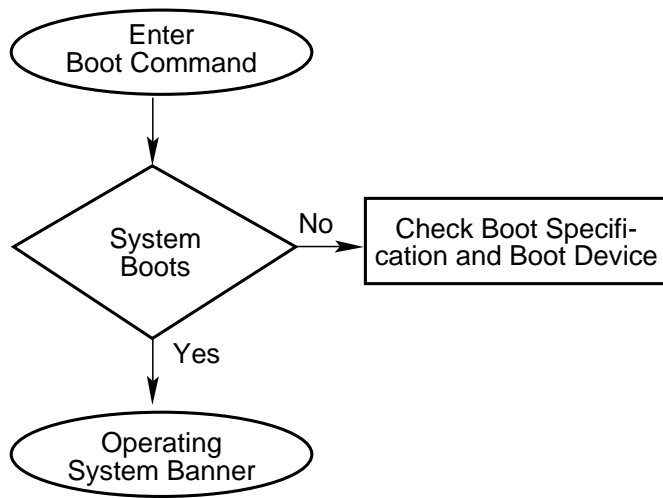


OM52-99

5.2 Troubleshooting During Booting

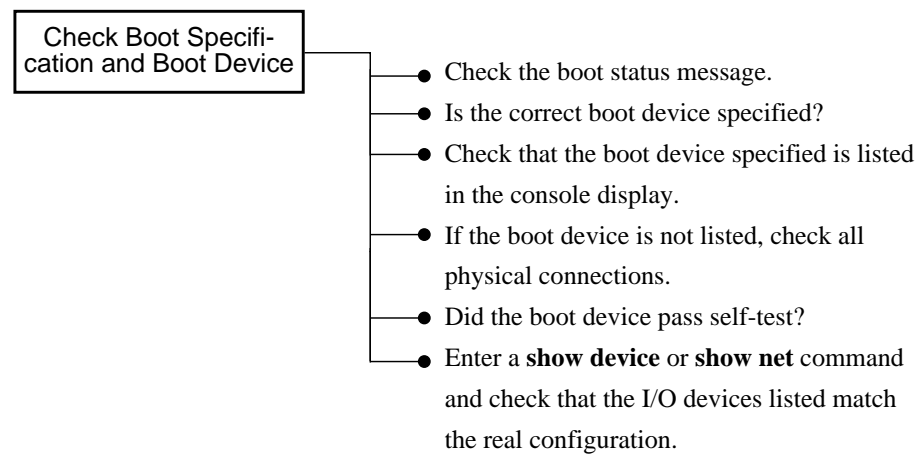
When booting fails for a GS60E system, you can check several parameters. Figure 5-3 shows the boot sequence. Figure 5-4 shows the steps to take if a problem occurs during booting. If you are unable to correct the problem, call your Compaq customer service engineer.

Figure 5-3 Booting Troubleshooting Flowchart



OM53-99

Figure 5-4 Troubleshooting Steps During Booting

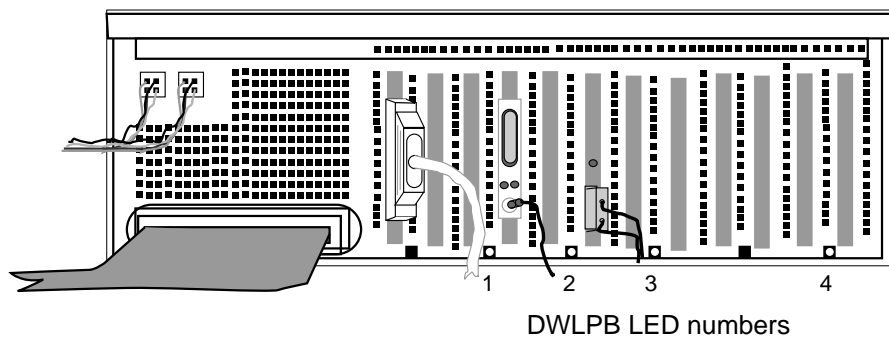


OM54-99

5.3 Troubleshooting a PCI Shelf

LEDs show the status of power supplies as well as the adapter self-test results in the PCI shelf.

Figure 5-5 DWLPB PCI Shelf



OM55-99

LED Status in PCI Shelf

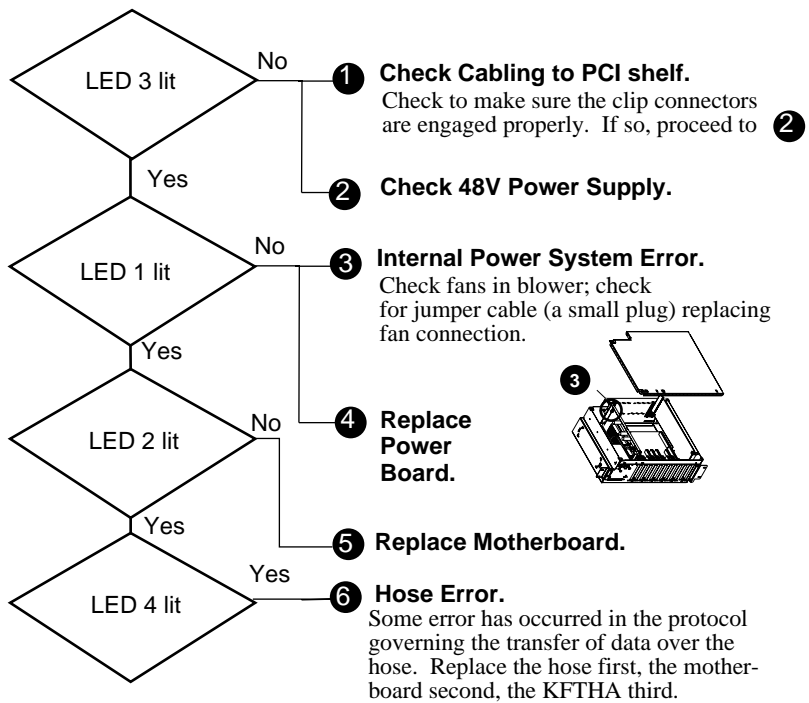
LED 1 - On-board power system OK

LED 2 - Motherboard self-test passed

LED 3 - 48 VDC power supply OK

LED 4 - Hose Error

Figure 5-6 Troubleshooting Steps for PCI Shelf

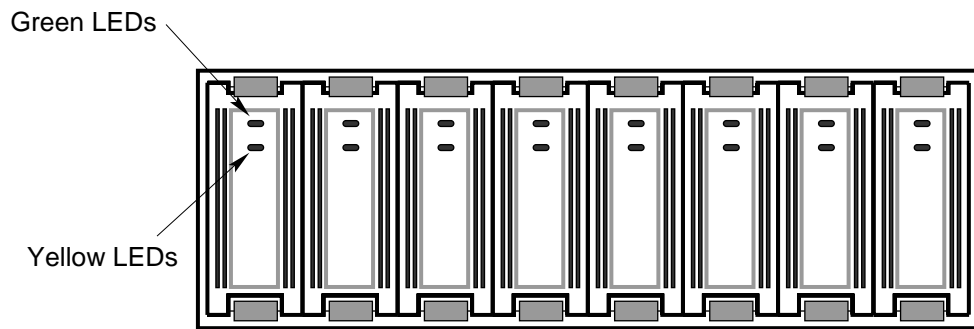


OM56-99

5.4 Troubleshooting a StorageWorks Shelf

StorageWorks LEDs are located on each disk drive. Table 5-1 lists the functions of the LEDs shown in Figure 5-7.

Figure 5-7 StorageWorks Indicator LEDs



OM57-99

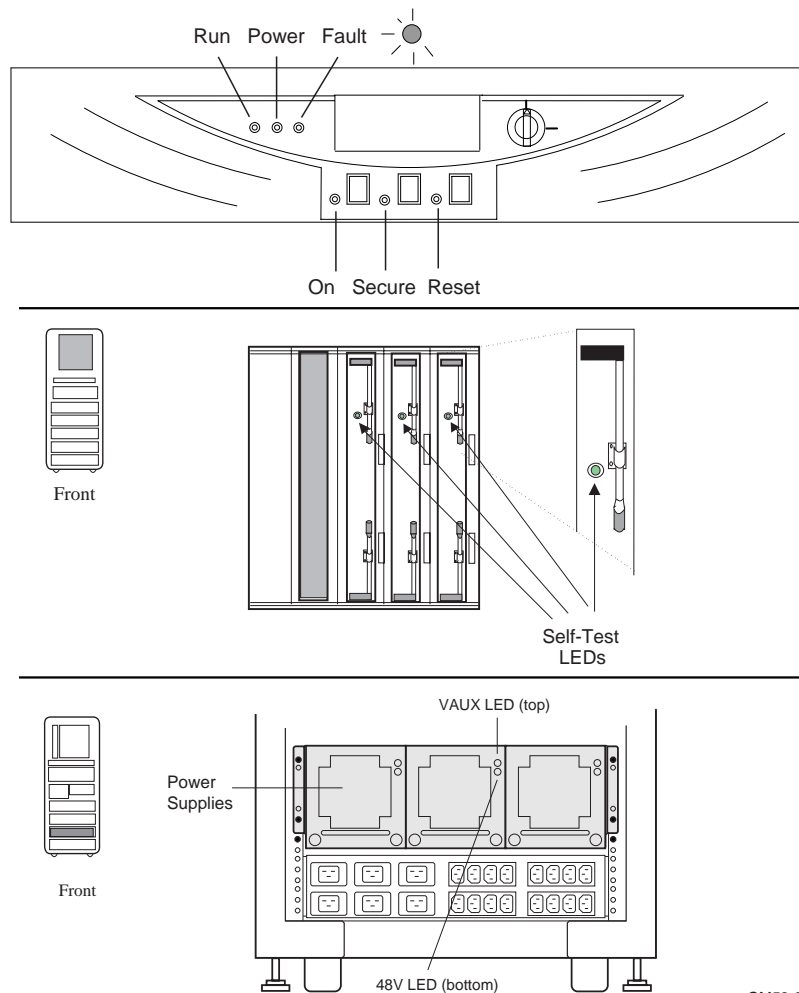
Table 5-1 SCSI Disk Drive LEDs

Indicator LED	LED State	Meaning
Green	Off	No activity
	Flashing	Activity
	On	Activity
Yellow	Off	Normal
	Flashing	Spin up/spin down
	On	Not used

5.5 Self-Test Overview

The system provides a record of testing in the self-test display. The control panel, module, and power supply LEDs also indicate success or failure.

Figure 5-8 Determining Self-Test Results



OM58-99

Following power-up and system reset, the system performs testing. Self-test results are indicated by the following:

- LEDs on the control panel
- Self-test LEDs on the modules and power supplies
- Self-test display

During system self-test, the yellow Fault light on the control panel lights. If a module fails self-test, the Fault light remains lit. If all modules pass self-test, the Fault light goes off.

Each CPU, memory, and I/O adapter module has a green LED that lights when the module passes self-test. These LEDs can be viewed through the module enclosure from the front and rear of the cabinet when the doors are open. If a module fails self-test, its green LED does not light. Although a dual-processor module's green LED will light only when both processors pass self-test, it is still possible to use the module if there is a – in the self-test display for one of the CPUs.

The self-test display is discussed in detail in this chapter.

5.6 Sample Self-Test Display

The primary console reports the results of self-test. The pass (+) or fail (-) status of each module is indicated.

Example 5-1 Self-Test Results

```

F   E   D   C   B   A   9   8   7   6   5   4   3   2   1   0   NODE # ①
                                A   M   M   M   .   .   P   P   P   TYP  ②
                                o   +   +   +   .   .   ++  ++  ++  ST1   ③
                                .   .   .   .   .   .   EE  EE  EB  BPD  ④
                                o   +   +   +   .   .   ++  ++  ++  ST2   ⑤
                                .   .   .   .   .   .   EE  EE  EB  BPD  ⑥
                                o   +   +   +   .   .   ++  ++  ++  ST3   ⑦
                                .   .   .   .   .   .   EE  EE  EB  BPD  ⑧

                                +   +   +   +   +   +   .   .   .   .   +   C0 PCI + ⑨
                                .   .   .   .   .   .   .   .   .   .   .   EISA +
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C1 ⑩
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C2 ①
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C3 ②

                                B0  A1  A0  .   .   .   .   ILV ③
                                .  4GB 4GB 4GB .   .   .   .  12GB ④
AlphaServer GS60E 6-6/525/4, Console V5.5 4-AUG-1999 12:06:03
SR0M V2.1, OpenVMS PALcode V1.57-2, Tru64 UNIX PALcode V1.50-1 ⑤
System Serial = NI84177052, OS = OpenVMS, 3:11:57 August 27, 1999
:
:
P00>>>

```

Table 5-2 System Configuration for Example 5-1

Module	TLSB Node #	Module Type
KN7CG	0, 1, 2	Dual processor.
MS7CC	5, 6, 7	Memory (4 Gbytes).
KFTHA	8	I/O adapter leading to one I/O channel.

During the first round of testing each module runs its own self-test and the results are shown on the ST1 line (❸). The boot processor is then determined (indicated by a B on the first BPD line). This processor then generates the results of testing to this point (❹).

If one processor on a CPU module fails self-test (ST1), the failing CPU is displayed as a minus under the node # (for example, - +). The processor failure is also indicated by the control panel Fault light remaining lit after power-up. If both processors fail, the module is disconnected from the backplane to prevent faulty system operation and the module self-test LED will be out.

Next, the processors run a second round of tests using the memory modules. In this testing the processor that had been designated as boot processor could fail (in which case a - would have been displayed in the ST1 line), so the boot processor is again determined. Results are displayed on the ST2 line (❺).

Finally, the processors run a third round of tests, the multiprocessing tests. Depending on configuration, there may be some delay at the ST3 line (around 20–60 seconds). Once again the boot processor is determined. The status of the boot processor and secondary processors is then displayed on the third BPD line (❻). The I/O adapter tests are then run.

Results of the I/O adapter self-test are displayed next.

The boot processor next configures memory and displays the configuration. Note that it is the boot processor determined at (❸) that displays the lines after the third BPD line. The final line before the console prompt contains the boot processor's SRM console version number, the date, and the SROM revision (Ⓢ).

Each numbered item in the self-test display is described in detail in the following sections of this chapter.

5.7 Self-Test Lines NODE # and TYP

The first two lines of the self-test printout provide the node number identification (NODE #) and the type of module (TYP).

Example 5-2 Self-Test Results: NODE # and TYP

```

F   E   D   C   B   A   9   8   7   6   5   4   3   2   1   0   NODE # ①
                                A   M   M   M   .   .   P   P   P   TYP ②
                                o   +   +   +   .   .   ++  ++  ++  ST1
                                .   .   .   .   .   .   EE  EE  EB  BPD
                                o   +   +   +   .   .   ++  ++  ++  ST2
                                .   .   .   .   .   .   EE  EE  EB  BPD
                                o   +   +   +   .   .   ++  ++  ++  ST3
                                .   .   .   .   .   .   EE  EE  EB  BPD

                                +   +   +   +   +   +   .   .   .   .   +   C0 PCI +
                                .   .   .   .   .   .   .   .   .   .   .   EISA +
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C1
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C2
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C3

                                B0  A1  A0  .   .   .   .   ILV
                                .  4GB 4GB 4GB .   .   .   . 12GB
Compaq AlphaServer GS60E 6-6/525/4, Console V5.5 4-AUG-1999 12:06:07
SR0M V2.1, OpenVMS PALcode V1.57-2, Tru64 UNIX PALcode V1.50-1
System Serial = NI84177052, OS = OpenVMS, 3:11:57 August 27, 1999
:
:
P00>>>

```

- ❶ The NODE # line lists the node numbers on the TLSB and I/O buses.

The nodes on this line are numbered in hexadecimal. Note that GS60E uses seven slots of the TLSB card cage: 0, 1, 2, 5, 6, 7, and 8 (slots 3 and 4 are not used). The slot numbers and node numbers are identical. Nodes 0 through 2 reflect the right-to-left position of the TLSB slots as you view the TLSB from the front of the cabinet and nodes 5 through 8 reflect the right-to-left position of the TLSB slots as you view the TLSB from the rear of the cabinet.

Each PCI bus has 12 slots, numbered 0 through B.

- ❷ The TYP line in the printout indicates the type of module at each TLSB node:

- An adapter: the KFTHA port module (A)
- A memory module (M)
- A processor (P) module

A period (.) indicates that the slot is not populated or that the module is not reporting.

5.8 Self-Test Lines ST and BPD

The next six lines provide test information on the processors (ST1, ST2, and ST#) and boot processor designation (BPD).

Example 5-3 Self-Test Results: ST and BPD

```

F   E   D   C   B   A   9   8   7   6   5   4   3   2   1   0   NODE #
                                A   M   M   M   .   .   P   P   P   TYP
                                o   +   +   +   .   .   ++  ++  ++  ST1   ③
                                .   .   .   .   .   .   EE  EE  EB  BPD   ④
                                o   +   +   +   .   .   ++  ++  ++  ST2   ⑤
                                .   .   .   .   .   .   EE  EE  EB  BPD   ⑥
                                o   +   +   +   .   .   ++  ++  ++  ST3   ⑦
                                .   .   .   .   .   .   EE  EE  EB  BPD   ⑧

                                +   +   +   +   +   +   .   .   .   .   +   CO PCI +
                                .   .   .   .   .   .   .   .   .   .   .   EISA +
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C1
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C2
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C3

                                B0  A1  A0  .   .   .   .   .   ILV
                                .  4GB 4GB 4GB .   .   .   .   .  12GB
Compaq AlphaServer GS60E 6-6/525/4, Console V5.5 4-AUG-1999 12:06:07
SR0M V2.1, OpenVMS PALcode V1.57-2, Tru64 UNIX PALcode V1.50-1
System Serial = NI84177052, OS = OpenVMS, 3:11:57 August 27, 1999
:
:
P00>>>

```

③ The ST1 line shows the results of self-test. The entries are:

- ++ (both processors pass)
- + – (one processor passes; one fails)
- o (does not apply)

NOTE: *If both CPUs fail on a module, there is no indication of the failure in the self-test display. The failing processor is logically disconnected from the backplane to prevent faulty system operation. The processor failure is indicated by the control panel Fault light remaining lit after power-up and the module's self-test LED being off.*

Since the I/O port module does not have a module-resident self-test, its entry for the ST1 line is always "o".

④ The BPD line indicates boot processor designation. When the system goes through self-test, the processor with the lowest ID number that passes self-test (ST1 line is +) becomes the boot processor, unless you intervene.

The results on the BPD line indicate:

- The boot processor (B)
- Processors eligible to become the boot processor (E)
- Processors ineligible to become the boot processor (D)

This BPD line is printed three times. After the first determination of the boot processor, the processors go through two more rounds of testing. Since it is possible for a processor to pass self-test (at line ST1) and fail ST2 or ST3 testing, the processors again determine the boot processor following each round of tests. The first processor to pass self-test is chosen as the boot processor.

- ⑤ During the second round of testing (ST2) all processors run additional CPU tests involving memory.
- ⑦ During the third round of testing (ST3) all processors run multi-processor tests, and the status of each processor is once again reported on the BPD line.
- ⑧ The primary CPU also tests the I/O port module at this time.

5.9 Self-Test Lines C0, C1, C2, ... Cn

The I/O channel lines (C0, C1, C2, ...) of the self-test display provide information on the node numbers and self-test status for modules in the I/O subsystems, which are connected to the system bus.

Example 5-4 Self-Test Results: C0, C1, C2, ..., Cn

```

F   E   D   C   B   A   9   8   7   6   5   4   3   2   1   0   NODE #
      A   M   M   M   .   .   P   P   P   TYP
      o   +   +   +   .   .   ++  ++  ++  ST1
      .   .   .   .   .   .   EE  EE  EB  BPD
      o   +   +   +   .   .   ++  ++  ++  ST2
      .   .   .   .   .   .   EE  EE  EB  BPD
      o   +   +   +   .   .   ++  ++  ++  ST3
      .   .   .   .   .   .   EE  EE  EB  BPD

      +   +   +   +   +   +   .   .   .   .   +   C0 PCI + 9
      .   .   .   .   .   .   .   .   .   .   .   .   .   EISA +
      .   .   .   .   .   .   .   .   .   .   .   .   .   C1 10
      .   .   .   .   .   .   .   .   .   .   .   .   .   C2
      .   .   .   .   .   .   .   .   .   .   .   .   .   C3

      B0  A1  A0  .   .   .   .   .   ILV
      .  4GB 4GB 4GB .   .   .   .   12GB
Compaq AlphaServer GS60E 6-6/525/4, Console V5.5 4-AUG-1999 12:06:07
SR0M V2.1, OpenVMS PALcode V1.57-2, Tru64 UNIX PALcode V1.50-1
System Serial = NI84177052, OS = OpenVMS, 3:11:57 August 27, 1999
:
:
P00>>>

```

The I/O channel lines (C0, C1, C2, ...) indicate the:

- I/O channel that connects the I/O adapter bus to the system bus
- DWLPB adapter self-test results
- PCI adapters' self-test results

A + indicates an adapter passed self-test, a – indicates a failure, and a period (.) indicates that that node number is not used.

- ⑨ In Example 5-4, the PCI (channel C0) and its options at nodes 0, 5, 6, 7, 8, 9, 10, and 11 passed self-test as indicated by the + symbols.
- ⑩ I/O channels C1, C2, and C3 are not used.

The **show configuration** command gives additional information on I/O subsystems and adapters.

5.10 Self-Test Lines ILV and GB

The ILV line details the interleaving of the memories, and the GB line gives the Gbytes of each memory module and the total size of the system memory.

Example 5-5 Self-Test Results: ILV and GB

```

F   E   D   C   B   A   9   8   7   6   5   4   3   2   1   0   NODE #
                                A   M   M   M   .   .   P   P   P   TYP
                                o   +   +   +   .   .   ++  ++  ++  ST1
                                .   .   .   .   .   .   EE  EE  EB  BPD
                                o   +   +   +   .   .   ++  ++  ++  ST2
                                .   .   .   .   .   .   EE  EE  EB  BPD
                                o   +   +   +   .   .   ++  ++  ++  ST3
                                .   .   .   .   .   .   EE  EE  EB  BPD

                                +   +   +   +   +   +   .   .   .   .   +   C0 PCI +
                                .   .   .   .   .   .   .   .   .   .   .   EISA +
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C1
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C2
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C3

                                B0  A1  A0  .   .   .   .   ILV ③
                                .  4GB 4GB 4GB .   .   .   . 12GB ④
AlphaServer GS60E 6-6/525/4, Console V5.5 4-AUG-1999 12:06:07
SR0M V2.1, OpenVMS PALcode V1.57-2, Tru64 UNIX PALcode V1.50-1
System Serial = NI84177052, OS = OpenVMS, 3:11:57 August 27, 1999
:
:
P00>>>

```


- ③ The ILV line contains a memory interleave value (ILV) for each memory. The default memory configuration algorithm attempts to maximize memory interleaving; arrays on a single memory module are interleaved by default. In Example 5-5, the memory modules at nodes 2 and 6 are each in a two-way system internal interleave (they are not interleaved with each other). There are two interleave sets; set A and set B. Information on memory interleaving can be obtained by entering a **show memory** command:

```
P00>>> sho mem
Set  Node  Size      Base Address      Intlv      Position
---  -
A    5      4096 Mb    00000000 00000000    8-Way      0
A    6      4096 MB    00000000 10000000    8-Way      1
```

A system with three memory modules is shown in the following example:

```
F  E  D  C  B  A  9  8  7  6  5  4  3  2  1  0  NODE #
.   B0 A3 A1 A0 . . . . ILV
.  2GB 2GB 2GB 4GB . . . . 10GB
```

In this example, there is one memory interleave set designated by the letter A. Note that there is no A2 designator on the ILV line; the memory at node 4 provides on-board interleaving, and so supplies both the A0 memory word and the A2 memory word (which is not reported on the ILV line). Also note that different size memory arrays can be interleaved into a single set by "stacking" the smaller arrays to interleave with their larger counterparts.

- ④ The line after the ILV line displays the size of each configured memory module in the system and gives the total size of system memory. In Example 5-5, the total size is 12 Gbytes.

5.11 Self-Test Identification Line

The last line of the self-test display gives the firmware revision numbers, the SROM revision numbers, and the date and time stamp of the console.

Example 5-6 Self-Test Results: Identification Line

```

F   E   D   C   B   A   9   8   7   6   5   4   3   2   1   0   NODE #
                               A   M   M   M   .   .   P   P   P   TYP
                               o   +   +   +   .   .   ++   ++   ++   ST1
                               .   .   .   .   .   .   EE   EE   EB   BPD
                               o   +   +   +   .   .   ++   ++   ++   ST2
                               .   .   .   .   .   .   EE   EE   EB   BPD
                               o   +   +   +   .   .   ++   ++   ++   ST3
                               .   .   .   .   .   .   EE   EE   EB   BPD

                               +   +   +   +   +   +   .   .   .   .   +   C0 PCI +
                               .   .   .   .   .   .   .   .   .   .   .   EISA +
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C1
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C2
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C3

                               B0   A1   A0   .   .   .   .   ILV
                               .   4GB 4GB 4GB .   .   .   .   12GB
AlphaServer GS60E 6-6/525/4, Console V5.5 4-AUG-1999 12:06:07
                               ⑤                               ⑥                               ⑦
SROM V2.1, OpenVMS PALcode V1.57-2, Tru64 UNIX PALcode V1.50-1 ⑧
System Serial=NI84177052, OS=OpenVMS, 8:43:39 August 16, 1999

P00>>> ⑨

```

- ⑤ Indicates the type of system, the number of processors, the CMOS technology and speed.
- ⑥ In Example 5-6, the primary processor indicates the version of the console firmware.
- ⑦ The date and time stamp indicates the production date of the running console.
- ⑧ The SROM Rev information indicates the primary processor's serial ROM version. The serial ROM contains the first level of console, diagnostic, and bootstrap code. This code initializes the CPU programmable features and diagnoses any faults detected along with the bootstrap path and bootstrapping code execution out to the main console program (the second level of console, diagnostic, and bootstrap code).
- ⑨ The console prompt indicates the number of the primary processor. In this example used to describe the self-test results, there are six CPUs, two on the module in node 0, two on the module in node 1, and two on the module in node 2. Therefore, possible console prompts that could appear are P00>>> through P05>>>. In this example, the first CPU on the first module was the first CPU to pass all three rounds of testing, so it became the boot processor.

5.12 Show Commands

To get system information, you can use the SRM commands, **show configuration**, **show network**, or **show device**. The **show** command output identifies the subsystem, module, or device you may want to test. The following sections explain the **show** command output.

5.12.1 Show Configuration

Enter the **show configuration** command to display the system hardware configuration.

Example 5-7 Sample System Hardware Configuration

```
P00>>> sho conf ❶
```

	Name	Type	Rev	Mnemonic	❷
	TLSB				
0++	KN7CG-AB	8025	0000	kn7cg-ab0	
1++	KN7CG-AB	8025	0000	kn7cg-ab1	
5+	MS7CC	5000	0000	ms7cc0	
6+	MS7CC	5000	0000	ms7cc1	
7+	MS7CC	5000	0000	ms7cc2	
8+	KFTHA	2000	0000	kftha0	
C0	PCI connected to kftha0			pci0	❸
0+	SIO	4828086	0004	sio0	
5+	QLogic ISP1040B	10201077	0005	isp0	
6+	QLogic ISP1040B	10201077	0005	isp1	
7+	QLogic ISP1040B	10201077	0005	isp2	
8+	KZPAA	11011	0002	kzpaa0	
9+	QLogic ISP1040B	10201077	0005	isp3	
A+	QLogic ISP1040B	10201077	0005	isp4	
B+	DE500-BA	191011	0030	tulip1	❹
	Controllers on SIO			sio0	
0+	DECchip 21040-AA	21011	0023	tulip0	
1+	FLOPPY	2	0000	floppy0	
2+	KBD	3	0000	kbd0	
3+	MOUSE	4	0000	mouse0	
6+	TOY	7	0000	toy0	

- ❶ The operator enters a **show configuration** command to show the system hardware configuration.
- ❷ The mnemonic for each processor, memory module, I/O adapter, and I/O subsystem is displayed in this column. You use mnemonics to identify the subsystem, module, or device you wish to test.
- ❸ This system has one PCI subsystem; identified in the mnemonic column as **pci0**. To test the I/O subsystem, you would issue the command **test kftha0**.
- ❹ To test an I/O adapter, such as the Ethernet adapter, you would enter the command **test tulip1**.

5.12.2 Show Network

Enter a show network command to display network devices.

Example 5-8 Sample Output of Show Network Command

```
P000>>> sho network ❶  
polling for units on tulip0, ❷ slot 12, bus 0, hose0...  
ewa0.0.0.12.0 00-00-F8-25-90-A6 Twisted-Pair ❸  
polling for units on tulip1, ❹ slot 11, bus 0, hose0...  
ewa0.0.0.11.0 08-00-2B-C3-9E-9B Twisted-Pair ❺
```

- ❶ In Example 5-8, the operator enters a **show network** command to show the network adapters in the system.
- ❷ The first Ethernet adapter is identified by the tulip0 mnemonic. This adapter is in slot 12 of the I/O card cage connected to I/O channel 0.
- ❸ The unit number for tulip0 is ewa0.0.0.12.0. The hardware address is 00-00-F8-25-90-A6.
- ❹ The second Ethernet adapter is tulip1. This adapter is in slot 11 of the I/O card cage connected to I/O channel 0.
- ❺ The unit number for tulip1 is ewa0.0.0.11.8. The hardware address is 08-00-2B-C3-9E-9B.

5.12.3 Show Device

Enter the show device command to display system devices.

Example 5-9 Sample Output of Show Device Command

```
P00>>> sho device ❶
polling for units on floppy0, slot 0, bus 1, hose4...
dva0.0.0.1100.4 ❷   DVA0 ❸   RX26/RX23 ❹
polling for units on kzpaa0, slot 5, bus 0, hose4...
pka.7.0.5.4      pka      Bus ID 7
dka400.4.0.5.4   DKA400   RRD47 1206
polling for units on isp0, slot 6, bus 0, hose4...
pkb.7.0.6.4      pkb      term on   Bus ID 7 5.57
dkb0.0.0.6.4     DKB0     RZ1CB-CS 0844 ❺
dkb100.1.0.6.4   DKB100   RZ1CB-CS 0844
dkb200.2.0.6.4   DKB200   RZ1CB-CS 0844
dkb300.3.0.6.4   DKB300   RZ1CB-CS 0844
dkb400.4.0.6.4   DKB400   RZ1CB-CS 0844
dkb500.5.0.6.4   DKB500   RZ1CB-CS 0844
dkb600.6.0.6.4   DKB600   RZ1CB-CS 0844
polling for units on ispl, slot 7, bus 0, hose4...
pkc.7.0.7.4      pkc      term on   Bus ID 7 5.57
polling for units on isp2, slot 8, bus 0, hose4...
pkd.7.0.8.4      pkd      term on   Bus ID 7 5.57
polling for units on isp3, slot 9, bus 0, hose4...
pke.7.0.9.4      pke      term on   Bus ID 7 5.57
polling for units on isp4, slot 10, bus 0, hose4...
pkf.7.0.10.4     pkf      term on   Bus ID 7 5.57
P00>>>
```


- ❶ In Example 5-9, the operator enters a **show device** command to show all disks and tapes supported by the system.
- ❷ The first adapter polled is the floppy drive with the mnemonic floppy0. It is located in slot 0, the bus number is 0, and the hose number is 4.
- ❸ Device information is displayed. Device mnemonics are listed in the first column. Device mnemonics are used with the **boot** command.
- ❹ The name of each disk as presented by the SCSI controller is displayed in the second column.
- ❺ The third column lists the device type.
- ❻ The fourth column lists the device firmware revision, if applicable. Not all devices report this.

5.13 Test Command

You can use the test command to test the entire system, an I/O subsystem, a module, a group of devices, or a specific device. Enter a show configuration command to see a list of the subsystems and devices that you may want to test. Examples of the test command are shown in the following sections.

Example 5-10 Sample Test Command

1. P00>>> test -q # Runs a system test. Since a test
run time was not specified, the
entire system will be tested
provided that testing does not
exceed 10 minutes. Status messages
will not be displayed.
2. P00>>> test ms7cc* # Tests all memory modules in the
system.
3. P00>>> test dua80.0.0.2.1 # Tests the disk unit 80 on
controller A. The disk adapter
located in slot 2.

Two helpful **test** command options are outlined in Table 5-3. Environment variables commonly used with **test** are listed in Table 5-4.

Table 5-3 Test Command Options

Option	Meaning
-t <value>	Specifies the test run time in seconds. For a system test the default run time is 600 seconds (10 minutes). The run time for a device test is dependent upon the number and type of devices selected for testing. The -t option takes any value between 30 and 99999999
-q	Disables the status messages displayed by default as exerciser processes are started and stopped during testing. -q sets d_verbose to zero.

Table 5-4 Test Command Environment Variables

Environment Variable	Meaning
d_report	Specifies the type of error report displayed. Values are summary and full . Full is the default value.
d_harderr	Specifies the action taken when a hard error occurs. Values are halt or continue . Halt is the default value.
d_softerr	Specifies the action taken when a soft error occurs. Values are halt or continue . Continue is the default value.

5.13.1 Testing the System

To test the entire system (except for memory), enter the test command.

Example 5-11 Sample Test Command, System Test

```
P00>>>test ❶
Console is in diagnostic mode
Complete Test Suite for runtime of 1200 seconds

Type ^C to stop testing ❷

Configuring system...
:
:
Memory Tests not run. Must run separately using TEST MS7CC* ❸
Starting network exerciser on ewa0.0.0.12.0 (id #28f) in internal loopback mode
Starting network exerciser on ewb0.0.0.11.0 (id #2a1) in internal loopback mode
Starting network exerciser on ewc0.0.0.12.4 (id #2b3) in internal loopback mode
Starting network exerciser on ewd0.0.0.11.4 (id #2c5) in internal loopback mode
Starting device exerciser on dka0.0.0.4.0 (id #36f) in READ-ONLY mode
Stopping device exerciser on dka0.0.0.4.0 (id #36f)
Starting device exerciser on dka100.1.0.4.0 (id #5df) in READ-ONLY mode
Stopping device exerciser on dka100.1.0.4.0 (id #5df)
Starting device exerciser on dka200.2.0.4.0 (id #858) in READ-ONLY mode
Stopping device exerciser on dka200.2.0.4.0 (id #858)
Starting device exerciser on dka300.3.0.4.0 (id #acc) in READ-ONLY mode
Stopping device exerciser on dka300.3.0.4.0 (id #acc)
Starting device exerciser on dka400.4.0.4.0 (id #d37) in READ-ONLY mode
Stopping device exerciser on dka400.4.0.4.0 (id #d37)

Stopping all testing... please wait ❹

Stopping network exerciser on ewd0.0.0.11.4 (id #2c5) ❺
Stopping network exerciser on ewc0.0.0.12.4 (id #2b3)
Stopping network exerciser on ewb0.0.0.11.0 (id #2a1)
Stopping network exerciser on ewa0.0.0.12.0 (id #28f)

-----Testing done -----
```

Example 5-12 Sample Test Command, System Test (Continued)

```
Shutting down drivers...
Shutting down units on tulip2, slot 12, bus 0, hose 4...
Shutting down units on floppy1, slot 0, bus 1, hose 4...
Shutting down units on isp4, slot 6, bus 0, hose 4...
Shutting down units on isp5, slot 7, bus 0, hose 4...
Shutting down units on isp6, slot 8, bus 0, hose 4...
Shutting down units on isp7, slot 9, bus 0, hose 4...
Shutting down units on isp8, slot 10, bus 0, hose 4...
Shutting down units on tulip3, slot 11, bus 0, hose 4...
Shutting down units on tulip0, slot 12, bus 0, hose 0...
Shutting down units on floppy0, slot 0, bus 1, hose 0...
Shutting down units on isp0, slot 4, bus 0, hose 0...
Shutting down units on isp1, slot 6, bus 0, hose 0...
Shutting down units on isp2, slot 7, bus 0, hose 0...
Shutting down units on isp3, slot 8, bus 0, hose 0...
Shutting down units on tulip1, slot 11, bus 0, hose 0...
:
:
P00>>> ⑥
```

- ① In Example 5-11, the operator enters the **test** command. The complete test suite runs for 1200 seconds.
- ② To stop execution of the test command before normal completion, use Ctrl/C (^C). Termination using ^C may take a number of seconds depending upon the particular configuration being tested.
- ③ Memory testing is done separately. Status messages indicate the start of the console-based exercisers.
- ④ Testing is complete..
- ⑤ All exercisers are stopped, as indicated by the status messages.
- ⑥ The console prompt returns.

5.13.2 Testing a Module or Devices

To test a processor, memory module, or an I/O adapter and its associated devices, enter the test command and the correct mnemonic. Mnemonics are displayed when you enter a show configuration or a show device command.

Example 5-12 Sample Test Command, Memory Test

```
P00>>> set d_report full
P00>>> test ms* ①
Console is in diagnostic mode
Memory subsystem test selected for runtime of 1200 seconds
Type Ctrl/C to abort...

*****
*
* ALLOW AT LEAST 2 MINUTES OF TESTING TIME FOR EACH GIGABYTE *
*                               OF MAIN MEMORY                *
*
* SINGLE-BIT ERROR REPORTING IS ENABLED                       *
*
*****

Starting Cache Coherency Tests
Starting Marching 1's and 0's Tests
Memory size is 8192 MB
More than 2 GB memory present ... memory size is 1FFE
Starting Victimize Tests
>2 GB memory testing beginning ...
  Starting test 4 at addresses 7F400000 and 10F800000
  Starting test 2 at addresses 13F900000 and 16FA00000
Starting test 2 at addresses AF500000 and 19FB00000
Still testing Memory...
Still testing Memory...
Still testing Memory...
  :
  :
Still testing Memory...
Still testing Memory...

Stopping all testing... please wait

-----Testing done -----
```

Example 5-12 Sample Test Command, Memory Test (Continued)

```
Shutting down drivers...
Shutting down units on tulip2, slot 12, bus 0, hose 4...
Shutting down units on floppy1, slot 0, bus 1, hose 4...
Shutting down units on isp4, slot 6, bus 0, hose 4...
Shutting down units on isp5, slot 7, bus 0, hose 4...
Shutting down units on isp6, slot 8, bus 0, hose 4...
Shutting down units on isp7, slot 9, bus 0, hose 4...
Shutting down units on isp8, slot 10, bus 0, hose 4...
Shutting down units on tulip3, slot 11, bus 0, hose 4...
Shutting down units on tulip0, slot 12, bus 0, hose 0...
Shutting down units on floppy0, slot 0, bus 1, hose 0...
Shutting down units on isp0, slot 4, bus 0, hose 0...
      :
      :
P00>>>
```

In Example 5-12:

- ❶ Enter **test ms***.
- ❷ All ms7cc memory modules are tested by the memory exerciser, a series of tests executed from the processor module.

NOTE: To test a single memory module on your system, type:
test ms7ccn, where **n** is the module number.

5.14 Error Reports

In the event of an error, either a summary or a full error report is displayed at the console. Error reports are specified by setting the `d_report` environment variable. See Example 5-13.

Example 5-13 Sample Summary Error Report

```
P00>>> set d_report summary # Command to set the type of error
                                # report to summary (default).
P00>>> test isp0             # Command to test the isp0
Console is in diagnostic mode
Device adapter test selected for runtime of 1200 seconds

Type ^C to stop testing

Configuring isp0
polling for units on isp0, slot 4, bus 0, hose0...
pka.7.0.4.0      pka          term on      Bus ID 7  5.57
dka0.0.0.4.0     DKA0           RZ1CB-CS 0844
dka100.1.0.4.0   DKA100        RZ1CB-CS 0844
dka200.2.0.4.0   DKA200        RZ1CB-CS 0844
dka300.3.0.4.0   DKA300        RZ1CB-CS 0844
dka400.4.0.4.0   DKA400        RZ1CB-CS 0844

Starting device exerciser on dka0.0.0.4.0 (id #155) in READ-ONLY mode ❶
Stopping device exerciser on dka0.0.0.4.0 (id #155)
Starting device exerciser on dka100.1.0.4.0 (id #1b5) in READ-ONLY mode
:
:
Starting device exerciser on dka100.1.0.4.0 (id #b8a) in READ-ONLY mode
Stopping device exerciser on dka100.1.0.4.0 (id #b8a)
Starting device exerciser on dka200.2.0.4.0 (id #bea) in READ-ONLY mode
Stopping device exerciser on dka200.2.0.4.0 (id #bea)
failed to send Read to dka200.2.0.4.0

*** Hard Error - Error #5 - ❷

Diagnostic Name      ID      Device  Pass  Test  Hard/Soft  6-AUG 1999
exer_kid ❸          00000bea      0    0    1    0    10:18:03 ❹
Error in read of 0 bytes at location 0004C000 from device dka200.2.0.4.0

*** End of Error ***
```


In Example 5-13:

- ❶ Testing begins.
- ❷ A hard error, error #5, is reported. The three types of errors reported are hard, soft, and fatal. The error number, in this case error #5, corresponds to the location of the actual error report call within the source code for the failing diagnostic. The FRU, or field-replaceable unit, is dka200.2.0.4.0.
- ❸ The program running when the error occurred is `exer_kid`.
- ❹ The time stamp shows when the error occurred.

The information shown in a full and a summary error report is the same, except the full error report shows extended information, such as a list of addresses and the expected and received data values for each address.

Chapter 6

SRM Console Commands

This chapter describes the SRM console program's command language, console special characters, console environment variables, and console commands.

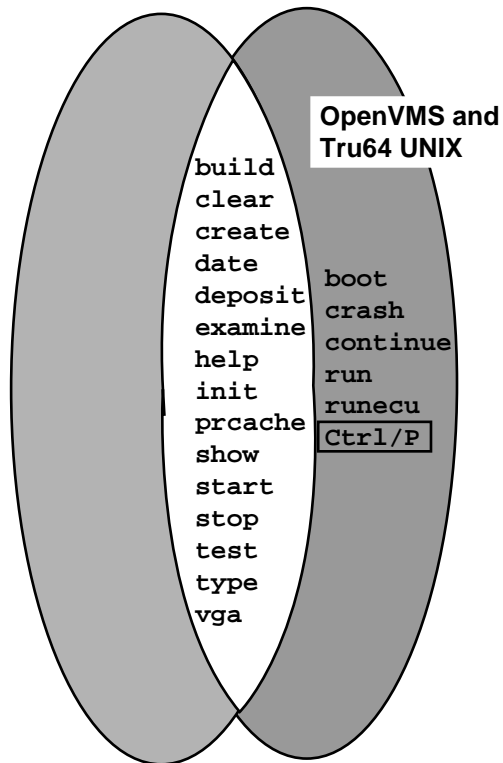
Sections in this chapter include:

- Overview
- SRM Command Syntax
- SRM Console Special Characters
- SRM Console Environment Variables
- SRM Console Commands

6.1 Overview

The SRM console firmware has evolved to provide capabilities for the OpenVMS and Tru64 UNIX operating systems offered with the AlphaServer GS60E system.

Figure 6-1 SRM Console Command and Feature Breakdown



OM61-99

SRM console commands allow you to boot the *OpenVMS* and *Tru64 UNIX* operating systems, display the configuration, and verify the system. The control character Ctrl/P allows you to halt the operating system and return to SRM console mode; the SRM command **continue** will then restart the operating system where it left off.

When the system is in SRM console mode, the system is halted and the SRM console firmware is executing. The operator communicates with the firmware through the console terminal, which displays the following prompt:

```
Pnn>>>
```

where *nn* is 00 to 05, depending on which TLSB slot the processor module is in and which processor on the module is the primary processor.

6.2 SRM Command Syntax

The SRM console command language has syntax rules for forming commands. Commands can contain up to 80 characters on a single line, can be abbreviated, and accept options. Tabs and spaces are compressed.

Table 6-1 SRM Console Command Language Syntax

Command Parameter	Attribute or Action
Length	256 characters maximum, including the terminating carriage return.
Case	Upper- or lowercase characters are accepted.
Abbreviation	Varies with the command; usually the shortest unique combination of letters.
Options	Can appear after the command keyword or after any symbol or number in the command. Begin with a hyphen (-) and must be preceded by at least one space.
Numbers	Most numbers are decimal. Addresses and numbers used with the deposit command are hexadecimal.
No characters	Null command; no action taken
Multiple adjacent spaces and tabs	Compressed to a single space.

Length: The SRM console program accepts commands of up to 255 characters. This does not include the terminating carriage return or any characters deleted as the command is entered. A command longer than 80 characters, without the backslash character causes the display of an error message.

Case: Upper- or lowercase characters can be used for input. Characters are displayed in the case they are entered.

Abbreviation: Commands and options can be abbreviated by dropping characters from the end of words. You must enter the minimum number of characters to identify the keyword unambiguously. All characters specified must match a keyword to be accepted. For example, although **E** uniquely identifies the **examine** command, **Exmn** is not a valid abbreviation. In the command reference sections that follow, characters that can be omitted appear in square brackets ([]). Abbreviation of environment variables (see Section 6.4) is allowed with the **show** command.

Options: You can use command options, to define or modify the environment, after the command keyword or after any symbol or number in the command. See individual keyword descriptions for examples.

Numbers: Most numbers in console commands are in decimal notation except for addresses and those used in the **deposit** command which are in hexadecimal. The default radix can be overridden by preceding decimal numbers with %d, binary with %b, hexadecimal with %x, and octal with %o. Refer to the individual command descriptions. Register names (R0, R1, and so on) are not considered numbers and use decimal notation.

No Characters: A command line with no characters is a null command. The console program takes no action and does not issue an error message. The console prompt returns. The console supports command line recall and editing.

Spaces: Multiple adjacent spaces and tabs are compressed and treated as a single space. The console program ignores leading and trailing spaces.

6.3 SRM Console Special Characters

The console program supports control characters, entered by holding down the Control (Ctrl) key and pressing the desired key, and other special characters.

Table 6–2 SRM Console Special Characters

Character	Function
Return	Carriage return; ends a command line.
Backslash	Line continuation.
Help	By itself, displays first-level help. When pressed after part of a command, displays options available.
Ctrl/A, F14	Toggles between insertion/overstrike mode.
Ctrl/B ^ (up-arrow)	Recall previous command(s).
Ctrl/C	Terminate running process.
< (left-arrow)	Move cursor left one position.
Ctrl/E	Move cursor to end of line.
Ctrl/F, > (right-arrow)	Move cursor right one position.
Ctrl/H	Move cursor to beginning of line.
Ctrl/J	Deletes word to the left of the cursor.
Ctrl/O	Stop output to console terminal for current command. Toggles between enable/disable.
Ctrl/P	In SRM console mode, acts like Ctrl/C. In program mode, on the <i>OpenVMS</i> or <i>Tru64 UNIX</i> operating systems, causes the boot processor to halt and begin running the SRM console program.
Ctrl/Q	Resume output to console terminal.
Ctrl/R	Redisplay the current line.
Ctrl/S	Stop output to console terminal.
Ctrl/U	Delete entire line.

Table 6–2 SRM Console Special Characters (Continued)

Character	Function
*	Wildcarding for certain commands.
“ “	Quotes for set environment variable name.
#	Comment specifier.

Return terminates command line input. No action is taken on a command line until it is terminated by a carriage return. If no characters are entered and the Return key is pressed, it is treated as a null command. No action is taken, and the console prompts for input. Carriage return is echoed as carriage return, line feed.

Backslash (\) allows continuation across lines from the terminal; must be the last character on the line to be continued.

Backspace deletes previously typed character.

Help provides additional information on console commands.

Ctrl/A toggles between insertion mode and overstrike mode for command line editing. The default mode is overstrike.

Ctrl/B or **up-arrow/down-arrow** recall the previous command(s). The last 16 commands are stored in the recall buffer.

Ctrl/C terminates the current command. Echoed as ^C, Ctrl/C clears Ctrl/S and also resumes output that was suspended using Ctrl/O. When Ctrl/C is entered as part of a command line, the line is deleted as if you entered Ctrl/U. Ctrl/C has no effect as part of a binary data stream.

Left-arrow moves the cursor one position to the left.

Ctrl/E moves the cursor to the end of the line.

Ctrl/F or **right-arrow** moves the cursor right one position.

Ctrl/H moves the cursor to the beginning of the line.

Ctrl/J deletes previously typed word.

Ctrl/O stops output to the console terminal until Ctrl/O is entered again. Ctrl/O is echoed as ^O followed by a carriage return and is not echoed when output is reenabled. Output is also reenabled when the console prompts for a

command, issues an error message, enters program mode, or when Ctrl/P is entered. It is not reenabled by displaying a **repeat** command.

Ctrl/P works like Ctrl/C and is echoed as ^C, if the console terminal is in SRM console mode. Under the *OpenVMS* and *Tru64 UNIX* operating systems, if the console terminal is in program mode and is secured, Ctrl/P is not echoed, but is passed to the operating system for processing. If the console terminal is in program mode and is not secured, Ctrl/P halts the processor and begins the SRM console program. See the **continue** command for additional information.

Ctrl/Q resumes console output to the console terminal that was suspended with Ctrl/S. Additional Ctrl/Q strokes are ignored. Ctrl/Q is not echoed.

Ctrl/R is echoed as ^R, followed by a carriage return, line feed, and printing the current command line. Deleted characters are omitted. This command is useful for hardcopy terminals.

Ctrl/S suspends output to the console terminal until Ctrl/Q is entered. Ctrl/S is not echoed.

Ctrl/U discards all characters that you entered on the current line. It is echoed as ^U, followed by a carriage return, line feed, and a new prompt.

* allows wildcarding with device names and environment variables. With the **build** command, wildcarding is permitted for a class of device; for example, KFTHA*. Wildcarding is allowed with the following commands:

- **build**
- **clear**
- **initialize**
- **set -d**
- **show**
- **show configuration**
- **show device**
- **show <envar>**
- **show network**
- **test**

Double quotes (" ") allow you to denote a string for environment variable assignment.

allows you to enter a comment. All characters following the # are recognized as a comment only. Exceptions include the above control characters.

6.4 SRM Environment Variables

Console environment variables allow the user to modify the way the console commands operate.

An environment variable is a name and value association maintained by the console program. The value associated with an environment variable is an ASCII string (up to 127 characters in length) or an integer. Some environment variables can be set to tailor the recovery behavior of the system on power-up and, for *OpenVMS* and *Tru64 UNIX* systems, after system failures.

Volatile environment variables are initialized to their default by a system reset. Nonvolatile environment variables stay set across system failures.

Environment variables can be created, modified, displayed, and deleted using the SRM commands **create**, **set**, **show**, and **clear**. A default value is associated with any variable that is stored in the EEPROM area.

Table 6-3 lists the predefined console environment variables, their attributes, and their functions.

Table 6-3 Environment Variables

Variable	Attribute	Function
arc_enable	Nonvolatile	Enables you to issue to the run command for <i>OpenVMS</i> and <i>Tru64 UNIX</i> systems. Default value is off .
auto_action	Nonvolatile	Specifies the action the console will take following an error halt or power-up. Values are: restart - For <i>OpenVMS</i> and <i>Tru64 UNIX</i> systems, automatically restart. If restart fails, boot the operating system. boot - Automatically boot the operating system. <i>OpenVMS</i> and <i>Tru64 UNIX</i> systems will use as the default device that defined by manufacturing (for factory-installed software), or a default boot device selected by setting the bootdef_dev environment variable. halt (default) - Enter SRM console mode.
bootdef_dev	Nonvolatile	For <i>OpenVMS</i> or <i>Tru64 UNIX</i> systems, the default device or device list from which booting is attempted when no device name is specified by the boot command.
boot_file	Nonvolatile	For <i>OpenVMS</i> or <i>Tru64 UNIX</i> systems, the default file name used for the primary bootstrap when no file name is specified by the boot command, if appropriate.
boot_osflags	Nonvolatile	For <i>OpenVMS</i> or <i>Tru64 UNIX</i> systems, additional parameters to be passed to the system software during booting if none are specified by the boot command with the -flags specifier.

Table 6-3 Environment Variables (Continued)

Variable	Attribute	Function
boot_reset	Nonvolatile	For OpenVMS and Tru64 UNIX systems, the default file name used for the primary bootstrap when no file name is specified in the boot command.
console	Nonvolatile	Defines the type of console device. serial A serial console terminal graphics A graphics console device both Both a serial and graphics console device
cpu	Nonvolatile	Selects the current boot processor.
cpu_enabled	Nonvolatile	A bitmask indicating which processors are enabled to run (leave console mode). Default is 0xffff .
cpu_primary	Nonvolatile	A bitmask indicating which processors are enabled to become the next boot processor, following the next reset. Default is 0xffff .
d_harderr	Volatile	Determines action taken following a hard error. Values are halt (default) and continue . Applies only when using test .
d_report	Volatile	Determines level of information provided by the diagnostic reports. Values are summary and full (default). Applies only when using test .
d_softerr	Volatile	Determines action taken following a soft error. Values are continue (default) and halt . Applies only when using test .
d_trace	Nonvolatile	Specifies whether or not to display test trace messages. off (default) - Disables trace messages on - Enables trace messages
dump_dev	Nonvolatile	Device to which dump file is written if the system crashes, if supported by the operating system.

(Continued on next page)

Table 6-3 Environment Variables (Continued)

Variable	Attribute	Function
enable_audit	Nonvolatile	If set to on (default), enables the generation of audit trail messages. If set to off , audit trail messages are suppressed, Console initialization sets this to on .
ew*0_loop_count	Nonvolatile	Specifies the number of times each message is looped for a test command exercising a PCI network adapter.
ew*0_loop_inc	Nonvolatile	Specifies the amount the message size is increased from message to message.
ew*0_loop_patt	Nonvolatile	Specifies the type of data pattern to be used when doing loopback. Current patterns are accessed by: 0xffffffff All the patterns 1 All zeroes 2 All ones 3 All A's 4 Incrementing 5 Decrementing
ew*0_loop_size	Nonvolatile	Specifies the size of the loop data to be used.
ew*0_lp_msg_node	Nonvolatile	Specifies the number of messages originally sent to each node.
ew*0_mode	Nonvolatile	Specifies the value for the Ethernet port node when it is started. Allowed values are: Auto-Sensing AUI Twisted-Pair Full Duplex, Twisted Pair BNC FastFD (Full Duplex) Auto-Negotiate

Table 6-3 Environment Variables (Continued)

Variable	Attribute	Function
interleave	Nonvolatile	Specifies interleave arrangement for memory modules. Defined values are: DEFAULT The SRM console decides how to interleave memories. NONE Memory modules are not interleaved.
os_type	Nonvolatile	Used to store operating system type that user will be operating under. Values are vms , openvms , osf , and unix .
simm_callout	See "Function"	Specifies that if any memory errors are detected during system self-test, the SRM console should identify the failing SIMM. Defined values are: OFF (default) ON Identify failing SIMM. The "ON" value is preserved across one system initialization only, then reset to OFF.
tta0_baud	Nonvolatile	Sets the serial console terminal port baud rate. Allowable values are 300, 600, 1200, 2400, 4800, and 9600. The default value is 9600.

6.5 SRM Console Commands

Console commands provide the capabilities to examine and modify system state. Additionally, they allow tests to be directed to functional components of the system.

6.5.1 Boot

The boot command boots the Tru64 UNIX and OpenVMS operating systems and the Loadable Firmware Update (LFU) utility.

Example 6-1 Boot Command

```
P00>>> show boot*
boot_dev          dkc300.3.0.7.1
boot_file
boot_osflags      a
boot_reset        OFF
bootdef_dev       dkc300.3.0.7.1
booted_dev
booted_file
booted_osflags
P00>>> boot
  (boot dkc300.3.0.7.1 -flags A)
SRM boot identifier: scsi 1 7 0 3 300 ef00 81011
boot adapter: kzpsa0 rev 0 in bus slot 7 off of kftha0 in TLSB slot 8
block 0 of dkc300.3.0.7.1 is a valid boot block
reading 16 blocks from dkc300.3.0.7.1
bootstrap code read in
Building FRU table.....
FRU table size = 0x2632
.
.
.
[initialization and configuration information]
.
.
.
Tru64 UNIX Version V4.0 (thenut.Eng.PKO.DEC.Com) console
login:
```

The boot command syntax is:

b[oot] [<device_name>] [<flags>]

Where <device_name> specifies a particular disk, and **flags** applicable to *OpenVMS* and *Tru64 UNIX* are described in detail in Appendix A.

6.5.2 Building the EEPROM

The `build -e` command is used to initialize a module's EEPROM during installation or to restore a corrupted serial EEPROM image.

Example 6-2 Building the EEPROM

```
P00>>> build -e                # Initialize the EEPROM
                                # on kn7cg.
Build EEPROM on kn7cg-ab0? [Y/N]> Y
      EEPROM built on kn7cg-ab0
P00>>>
```

The **`build -e`** command syntax is:

`bu[ild] -e <device>`

where **`<device>`** is a processor module. If you are restoring a corrupted EEPROM, you will be prompted to supply the system serial number and module serial, part, and firmware revision numbers. Since some environment variables are volatile, before upgrading you should refer to Table 6-3 and use the **`show <envar>`** command (see Section 6.5.28) to display present environment variable values. After rebuilding, use the **`set <envar>`** command (see Section 6.5.20) to set the environment variables to their desired values.

6.5.3 Building the Nonvolatile RAM

The `build -n` command is used to initialize the CPU's nonvolatile RAM (NVR).

Example 6-3 Building the Nonvolatile RAM

```
P00>>> build -n
Build NVR on kn7cg-ab0? [Y/N]Y
NVR built on kn7cg-ab0
P00>>>
```

The **`build -n`** command syntax is:

`bu[ild] -n <device>`

where **`<device>`** is a processor module. The **`build -n`** command sets the NVR to its default values. The **`build -e`** command may be required during a console firmware upgrade.

6.5.4 Building the SEEPROM

The `build -s` command is used to restore a module's corrupted serial EEPROM.

Example 6-4 Building the SEEPROM

```
P00>>> build -s ms7cc0          # Initialize the serial
                                # EEPROM on ms7cc0.
Build serial EEPROM on ms7cc0? [Y/N]> Y
This program will take at most several minutes
    Serial EEPROM built on ms7cc0
P00>>>
```

The **`build -s`** command syntax is:

`bu[ild] -s <device>`

where **<device>** is MS7CC, KFTHA, or DWLPB. Wildcarding is allowed within a class of devices (for example, **`build -s ms7cc*`**). The serial EEPROM in a device is used to log diagnostic symptom and test-directed data (DIAG_SDD, DIAG_TDD) and holds the module's serial number. It is also used by the operating system to log operating system symptom data. During manufacturing, all module serial EEPROMs are initialized (using **`build -s *`**). In the case of a module upgrade, the serial EEPROM should be initialized also.

6.5.5 Clear EEPROM

The `clear eeprom` command allows you to clear the selected EEPROM option.

Example 6-5 Clear EEPROM Command

```
P00>>> clear eeprom log      # Clears all failure
                               # information logged in
                               # EEPROM.
```

The **`clear eeprom`** command syntax is:

`cl[ear] ee[prom] <option>`

The **`clear eeprom`** command can be used to clear **`diag_sdd`**, **`diag_tdd`**, **`halt`**, **`symptom`**, or **`log`**.

6.5.6 Clear <envar>

Clear <envar> is used to remove an environment variable.

Example 6-6 Clear <envar>

```
P00>>> create fred      # Create fred with null value
fred set to
P00>>> set fred "this is a string in an environment variable"
P00>>> show fred
fred      this is a string in an environment variable
P00>>> clear fred
P00>>> show fred
Environment variable not found
P00>>>
```

The **clear <envar>** removes an environment variable. However, some environment variables, such as **tta0_baud**, are permanent and cannot be removed.

The **clear <envar>** command syntax is:

cl[ear] <envar>

where **<envar>** is the name of an environment variable, for example, a boot specification to be cleared (see Table 6-3).

6.5.7 Clear Screen

The clear screen commands allows you to clear the terminal screen.

Example 6-7 Clear Screen Command

```
P00>>> clear screen          # Refresh the terminal
                                # screen.
```

The **clear screen** command syntax is:

cl[ear] sc[reen]

There are no parameters or options.

6.5.8 Continue (for OpenVMS or Tru64 UNIX)

For OpenVMS or Tru64 UNIX systems, the `continue` command resumes processing at the point where it was interrupted by a Ctrl/P. Programs continue executing at the address currently in the program counter of the processor.

Example 6-8 Continue Command

```
$ ^P                # Stop processing on boot processor;
                   # processor enters console mode.
halted CPU 0
CPU 1 is not halted
halt code = 1
operator initiated halt # System responds with message;
PC = ffffffff80008c04  # system has halted with 80008c04
                       # in the program counter (PC).
P00>>>              # Console session begins
                       #
                       #
                       #
P00>>> continue     # Processor resumes at the address
                       # where processing was stopped by
                       # Ctrl/P. Here processing continues
                       # at address 80008c04.
```

The **continue** command syntax is:

c[ontinue]

Continue causes the primary processor to resume program mode, executing at the address currently in the program counter (PC). This address is the address that was in the PC when the primary processor received a Ctrl/P command. The system displays the hexadecimal PC value.

When the boot processor receives a **continue** command, it does not perform processor initialization as it would for a boot procedure. The boot processor just returns to the program it was processing.

NOTE: *Following execution of the **continue** command, the console terminal enters program mode, and any ASCII characters entered on the console terminal are passed on to the operating system. In program mode, the console terminal acts like any other terminal on the system until a Ctrl/P is issued to return it to console mode.*

*^P followed by **continue** should be used selectively since some console commands (for example, **deposit**, **set host**, **show device**, **show network**, and **test**) can corrupt the machine state so that the execution of the current program cannot resume successfully.*

6.5.9 Crash (for OpenVMS or Tru64 UNIX)

For Open VMS or Tru64 UNIX systems, the crash command causes the operating system to be restarted and generates a memory dump.

Example 6-9 Crash Command

```
P01>>> crash
      [operating system output appears]
```

The **crash** command causes the operating system to be restarted. This allows the user to ^P a hung system and generate a memory dump.

The **crash** command syntax is:

crash

There are no parameters or options.

6.5.10 Create

The create command allows you to create an environment variable.

Example 6-10 Create Command

```
1. P00>>> create fred          # Create a new environment
    fred set to                # variable fred with a value
P00>>> show fred              # equal to null.
    fred

2. P00>>> create stuff 356     # Create a new environment
                                # variable stuff with a
                                # value equal to 356.
```

The **create** command syntax is:

cre[ate] <envar> [<value>]

where the **<envar>** is the environment variable name and **<value>** is the optional variable value. Created environment variables are volatile by default. **value** can be a quoted string for specifying boot specifications (see **boot** command description).

6.5.11 Date

The date command is used to display or set the system date and time.

Example 6-11 Date Command

```
P02>>> date
15:30:27 June 3, 1999      # System displays time
                           # and date.
```

The **date** command syntax is:

da[te] [<yyyymmddhhmm.ss>]

where you can set the date and time by using **yyyy** for the year, **mm** for the month, **dd** for the day, **hh** for the hour, **mm** for minutes, and **ss** for seconds.

6.5.12 Deposit

The deposit command stores data in a specified location.

Example 6-12 Deposit Command

```
1. P00>>> dep -b -n 1FF pmem:0 0 # Clear first 512 bytes
# of physical memory.
2. P00>>> d -l -n 3 vmem:1234 5 # Deposit 5 into four
# longwords starting at
# virtual memory address
# 1234.
3. P00>>> d -n 8 R0 FFFFFFFF # Load GPRs R0 through R8
# with -1.
4. P00>>> d -l -n 10 -s 200 pmem:0 8 # Deposit 8 in the
# first longword of the first
# 17 pages in physical
# memory.
5. P00>>> d -l pmem:0 0 # Deposit 0 to physical
# memory address 0.
6. P00>>> d + FF # Deposit FF to physical
# memory address 4.
7. P00>>> d scbb 820000 # Deposit SCBB
# with 820000.
```

When using **deposit**, if no options are given in subsequent commands, the system uses the options from the preceding commands as the defaults for address or location referenced, data type (**-b**, **-l**, **-w**, and so forth), data size for increment (**-s**), and address space (**gpr**, **ipr**, **pmem**, and so forth).

The **deposit** command syntax is:

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

where the options are values from Table 6-4, and <*data*> is the value to be stored. If the specified value is too large to fit in the data size to be deposited, the console ignores the command and issues an error response. For data lengths longer than a longword, each longword of data should be separated by a space. If the data is smaller than the data size to be deposited, the higher order bits are filled with zeros.

Table 6-4 Deposit Command Options

Option	Meaning
-b	Defines data size as a byte.
-h	Defines data size as a hexword.
-l	Defines data size as a longword; initial default.
-o	Defines data size as an octaword.
-q	Defines data size as a quadword.
-w	Defines data size as a word.
-n val	Number of consecutive locations to modify.
-s val	Specifies the address increment size. Default is data size.

space: is the optional device name (or address space) of the device to access (see Table 6-5), and *address* specifies the offset within a device to which data is deposited. Valid symbolic address forms include:

- *fpr-name*, a symbol representing a floating-point register.
- *gpr-name*, a symbol representing a general purpose register.
- *ipr-name*, a symbol representing the internal processor register.
- PC, the program counter. The address space is set to GPR.
- *pt-name*, a symbol representing a PAL temp register.
- +, the location immediately following the last location referenced in an examine or deposit 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 one.

- -, the location immediately preceding the last location referenced in an examine or deposit command. For physical and virtual memory, the referenced location is the last location minus 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 minus one.
- *, the last location referenced in an examine or deposit command.
- @, the location addressed by the last location referenced in an examine or deposit command.

NOTE: *Since the console program actually resides in low memory when running, depositing to memory should be done with care.*

Table 6-5 Device Name and Address Space Options

Option	Device Name and Address Space Meaning
<dev_name>	Device name: pci0, kn7cg1, demna0, and so forth.
fpr	Defines the address space as the floating-point register set, F0 through F31.
gpr	Defines the address space as the general register set, R0 through R31.
ipr	Defines the address space as the internal processor registers (IPRs).
pt	Defines the address space as the PAL temp register set, PT0 through PT23.
pmem	Defines the address space as physical memory; initial default.
vmem	Defines the address space as virtual memory. All access and protection checking occur.

For more information, see the *Alpha Architecture Reference Manual*.

6.5.13 Examine

The examine command displays the contents of a memory location, a register, or a device. The options are similar to the deposit command options.

Example 6-13 Examine Command

1. P00>>> examine pc # Examine the
PC psr: 0 (PC) 0000000000001170 # program counter.
2. P00>>> examine sp # Examine the
gpr: F0 (R30) 0000000000072A60 # stack pointer.
3. P00>>> e -n 6 r4 # Examine register R4
and the next six
registers.

gpr: 20 (R4) 0000000000005000
gpr: 28 (R5) 00000000FFFE000
gpr: 30 (R6) 00000003F8000C00
gpr: 38 (R7) 0000000053F761AE
gpr: 40 (R8) 0000010000000000
gpr: 48 (R9) 00000003F7800100
gpr: 50 (R10) 00000000000C7FFC
4. P00>>> examine pmem:400EC # Examine physical
pmem: 400EC A49D0078A47D0070 # memory.
5. P00>>> examine demna0:0 # Examine demna0's
demna0: 0 0000000108020C03 # Device Register.

The **examine** command syntax is:

e[xamine] [-{b,w,l,q,o,h,d}] [-{n val, s val}] [space:] <address>

where the options are values from Table 6-6, **space:** is the optional device name (or address space) of the device to access, and **address** is a longword that specifies the first location to be examined.

The display line consists of the device name, the hexadecimal address or offset within the device, and the examined data also in hexadecimal.

Table 6-6 Examine Command Options

Option	Meaning
-b	Defines data size as a byte.
-d	Disassembles instruction at current address.
-h	Defines data size as a hexword.
-l	Defines data size as a longword; initial default.
-o	Defines data size as an octaword.
-q	Defines data size as a quadword.
-w	Defines data size as a word.
-n val	Number of consecutive locations to examine.
-s val	Specifies the address increment size. Default is data size.

Examine uses most of the same options as **deposit**. Additionally, the **examine** command supports the **-d** option (instruction decode, which will disassemble the instructions at the current address). When using **examine**, if no options are given in subsequent commands, the system uses the options from the preceding commands as the defaults for address or location referenced, data type, including **-d**, (**-b**, **-l**, **-w**, and so forth), data size for increment (**-s**), and address space (**gpr**, **i****pr**, **pmem**, and so forth).

After initialization, the default address space is physical memory, the default data size is a longword, the default address is zero, and the default address increment size is the data size. If conflicting address space or data sizes are specified, the console ignores the command and issues an error response.

Table 6-7 Device Name and Address Space Options

Option	Device Name and Address Space Meaning
<dev_name>	Device name: pci0, kn7cg1, demna0, and so forth.
fpr	Defines the address space as the floating-point register set, F0 through F31.
gpr	Defines the address space as the general register set, R0 through R31. The data size is always a quadword.
i pr	Defines the address space as the internal processor registers (IPRs). The data size is always a longword.
pt	Defines the address space as the PAL temp register set, PT0 through PT23.
pmem	Defines the address space as physical memory.
v mem	Defines the address space as virtual memory. All access and protection checking occur.

6.5.14 Halt (OpenVMS or Tru64 UNIX)

The halt command halts the specified processor or device. Equivalent to the stop command.

Example 6-14 Halt Command

```
P00>>> halt
P00>>>
```

The **halt** command syntax is:

halt [-drivers[<device_prefix>]][<processor_num>]

where **-drivers[device_prefix]** specifies the name of the device or device class to stop. If no device prefix is specified, then all drivers are stopped. The **<processor_num>** argument specifies the number of the processor to stop.

6.5.15 Help or Man

The help (or man) command provides basic information on the console commands.

Example 6-15 Help Command

```
1. P00>>> help create      # Display basic create command
   NAME                    # information. Minimum
   create                  # command input is highlighted.
   FUNCTION
   create environment variable
   SYNOPSIS
   create <envar> <value>
       [-nv] [integer] [-string]

2. P00>>> help h           # Display help information on
                           # all commands beginning with h.
   NAME
   halt
   FUNCTION
   Halt the specified processor or device.
   SYNOPSIS
   halt [-drivers [<device_prefix>]] [processor_num]
   NAME
   help
   FUNCTION
   Display information about console commands.
   SYNOPSIS
   :
   :
```

Example 6-13 Help Command (Continued)

```
3. P02>>> help runecu
NAME
    runecu
FUNCTION
    Run the EISA Configuration Utility from floppy.
SYNOPSIS
    runecu
P02>>> help run
NAME
    run
FUNCTION
    Run an ARC utility program.
SYNOPSIS
run <program> [-d <device>][-p <n>][-s <parameter string>]
  where:
<device> is the console device containing the program.
          Default is dva0.
<n> is the unit number of PCI to configure.(ie,0 for pci0)
<parameter string> Any optional parameters which must be
                   passed to the utility, must be enclosed
                   in quotes.
```

The **help** command syntax is:

he[lp] [<command>]

where **<command>** is one of the console commands. The **<helpkey>** can also be used after a partial command has been typed. For example, **set <helpkey>** will display the options supported by the **set** command.

6.5.16 Init

The `init` command performs a reset.

Example 6-16 Init Command

```
P00>>> init
```

The **`init`** command syntax is:

`i[nit]`

The **`init`** command is used to reset, or initialize, the entire system. The **`init`** command resets the machine and runs systemwide self-test. Self-test results are displayed after a system reset.

6.5.17 Prcache

The prcache command manipulates an optional NVRAM disk cache.

Example 6-17 Prcache Command

```
P00>>> prcache -f
PCI NVRAM Disk Cache: passed
Size: 2MB
Base Address: 040000000
System ID: 21000035
State: valid
Battery Status: good
Battery Disconnect Circuit Status: disabled
```

The **prcache** command syntax is:

prcache [-z] [-f] [-u] [nvram_bus]

where **-z** zeros the NVRAM disk cache

-f displays the NVRAM status

-u disables the NVRAM battery disconnect circuit

nvram_bus defines the action to be performed on the NVRAM card installed on the bus.

6.5.18 Run (for OpenVMS or Tru64 UNIX)

For OpenVMS or Tru64 UNIX systems, the run command is used to run an ARC utility program. Run has four options: rcu (the RAID Configuration Utility), swxcfrw, eeromcfg, and util_cli. The arc_enable environment variable must be set to on before this command can be used. See Appendix C for more information.

Example 6-18 Run Command

```
P00>>> init                                # Initialize after setting
                                           # arc_enable on.
F  E  D  C  B  A  9  8  7  6  5  4  3  2  1  0  NODE #
                                           A  A  M  M  .  .  .  P  P  TYP
                                           o  o  +  +  .  .  .  ++ ++ ST1
                                           .  .  .  .  .  .  .  EE EB BPD
                                           o  o  +  +  .  .  .  ++ ++ ST2
                                           .  .  .  .  .  .  .  EE EB BPD
                                           +  +  +  +  .  .  .  ++ ++ ST3
                                           .  .  .  .  .  .  .  EE EB BPD
                                           +  +  .  +  +  +  +  +  .  .  .  +  C0 PCI +
                                           .  .  .  .  .  .  .  .  .  .  .  .  EISA +
                                           .  .  .  .  .  .  .  .  .  .  .  .  C1
                                           .  .  .  .  .  .  .  .  .  .  .  .  C2
                                           .  .  .  .  .  .  .  .  .  .  .  .  C3

                                           .  .  A1 A0 .  .  .  .  ILV
                                           .  .  4GB 4GB .  .  .  .  8GB
Compaq AlphaServer GS60E 4-6/525/4, Console V5.5-11 7-JUL-1999 13:15:44
SR0M V2.3, OpenVMS PALcode V1.61-1, Tru64 UNIX PALcode V1.54-1
System Serial = qv, OS = OpenVMS, 3:11:57 August 2, 1999
P00>>> show dev isp31                       # Use show dev to find CD drive.
polling for units on isp1, slot 6, bus 0, hose0...
pkb.7.0.5.0          pkb                    Bus ID 7
dkb500.5.0.5.0       DKB500                 RRD47 0436
P00>>> run eeromcfg -d dkb500

Loading image...
Transferring control...
# The screen is erased, followed by:
```

```
-----
ISP1020/1040 EEROM Configuration Utility V2.1-102
-----
```

```

+===== Select ISP1020 =====+
| Bus      Virtual Slot    Address |
| [  0          0          p0 ] |
|   0          1          p1 |
|   0          4          p4 |
|   0          5          p5 |
+-----+
| < (O)K >      < E(x)it > |
+=====+

```

The **run** command has four options:

1. **rcu** - Run the RAID Configuration Utility
2. **swxcrfw** - Update firmware on RAID controller.
3. **eeromcfg** - Run the ISP1020/1040 EEROM Configuration Utility.
4. **util_cli** - Run the KZPSA Configuration Utility

Options 1 and 2 (**rcu** and **swxcrfw**) are on floppy and require a floppy drive.
Options 3 and 4 (**eeromcfg** and **util_cli**) are on CD.

The **run** command syntax is:

run <program> [-d <device>][-p <n>][-s <parameter string>]

where **<program>** is the command option selected, **<device>** is the console device containing the program (default is dva0), **<n>** is the unit number of the PCI to configure, and **<parameter string>** are any optional parameters that must be passed to the utility, which must be enclosed in quotes. The utility documentation specifies any required parameters.

To use this command, the **arc_enable** environment variable must be set to **on** and the system initialized. An example would be:

```
P08>>> run swxcrfw -p 1 n -s "swxcrfw.215 -v 215"
```


6.5.19 Set EEPROM

The `set eeprom` command allows you to set the selected EEPROM option.

Example 6-20 Set EEPROM Command

```
1. P00>>> set eeprom field
   LARS #> 09494820          # Enter labor activity
   Message> EEPROM update   # reporting system (LARS)
   P00>>>                   # number (8 digits) and
                               # message (up to 68
                               # characters).
2. P00>>> set eeprom man    # Enter module serial number
                               # and unified part number.

   Module Serial Number> SG226LFH01
   Module Unified 2-7-2.4 Part Number> -E2040-AA. M06
   P00>>>
```

The **`set eeprom`** command syntax is:

`se[t] ee[prom] <option>`

where **option** is **field**, **halt**, **manufacturing**, **serial**, or **symptom**. You must use the **`set cpu`** command (see second example in Example 6-21) for the CPU you would like to act on.

6.5.20 Set <envar>

Set <envar> allows you to modify environment variables.

Example 6-21 Set <envar>

```
1. P00>>> set auto_action restart # On an error halt,
                                     # system will automatically
                                     # restart. If restart fails,
                                     # boot the operating system.

2. P00>>> set cpu 1                  # Designate CPU in slot
   cpu set to 1                       # 1 as the primary, or
   P01>>>                               # boot, processor.

3. P00>>> set d_harderr halt         # System will halt on hard
                                     # error.

4. P00>>> se class                   # Set the value of
                                     # environment variable
                                     # class to null.

5. P00>>> show enable*               # Display the status
   enable_audit OFF                   # of the enable_audit
   P00>>> set enable_audit on        # environment variable. Set
                                     # enable_audit on to enable
                                     # generation of audit trail
                                     # messages.
```

The **set <envar>** syntax is:

se[t] <envar> [value]

where **envar** (environment variable) and **value** are from Table 6-3, which also indicates which environment variables are volatile. Certain environment variables, such as boot specifications, must be defined using the **create** command. For additional information, see Section 6.5.10. Unambiguous abbreviations can be used for an environment variable name when using the **set** command. **Set -d envar** resets the value of **envar** to its default value. Wildcarding is also allowed with the **set** command. For example, **set -d *** resets all environment variables to their default values.

6.5.21 Set SEEPROM

The set seeprom command allows you to set the selected SEEPROM option.

Example 6-22 Set SEEPROM Command

```
1. P00>>> set seeprom field
   LARS #> 09494820          # Enter labor activity
   Message> SEEPROM update  # reporting system (LARS)
   P00>>>                   # number (8 digits) and
                              # message (up to 68
                              # characters).

2. P00>>> set seeprom man kftha0 # Enter module serial
                                     # number, part number, and
                                     # device type.
   Module Serial Number> SG226LFH01
   Module Unified 2-7-2.4 Part Number> -E2040-AA. M06
   Module Device Type> 2000
   P00>>>
```

The **set seeprom** command syntax is:

se[t] see[prom] <option> <device>

where **option** is **field**, **manufacturing**, or **serial** and **device** is the device mnemonic; for example, **kftha1**.

6.5.22 Set Time

The set time command allows you to set the CPU time.

Example 6-23 Set Time Command

```
P00>>> set time 199908031115
```

The **set time** command syntax is:

Se[t] tim[e] yyyyymmddhhmm[.ss]

where **yyyy** is the year, **mm** is the month, **dd** is the day, **hh** is the hour, **mm** is minutes, and the optional **.ss** is seconds.

6.5.23 Show Configuration

The show configuration command displays the last configuration seen at system initialization.

Example 6-24 Show Configuration Command

```
P00>>> show configuration
```

Name	Type	Rev	Mnemonic
TLSB			
0++ KN7CG-AB	8014	0000	kn7cg-ab0
6+ MS7CC	5000	0000	ms7cc0
7+ KFTHA	2020	0000	kftha0
8+ KFTHA	2000	0000	kftha1
C0 PCI connected to kftha0			
0+ SIO	4828086	0003	sio0
7+ KZPSA	8101	0000	kzpsa0
8+ ISP1020	8101	0000	kzpsa1
A+ DAC960	11069	0000	dac0
Controllers on SIO			
0+ DECchip 21040-AA	21011	0000	tulip0
1+ FLOPPY	2	0000	floppy0
2+ KBD	3	0000	kbd0
3+ MOUSE	4	0000	mouse0

```
P00>>>
```

The **show configuration** command syntax is:

sh[ow] c[onfiguration] [-v]

where **-v** indicates vector displayed.

The screen displays the system configuration, including the hardware device type, revision level, and mnemonic for each TLSB node.

6.5.24 Show CPU

The show CPU command displays information on CPUs in the system.

Example 6-25 Show CPU Command

```
P08>>> show cpu
Primary CPU:      00
Active CPUs:     00 01 02
Configured CPUs: 00 01 02 03
P08>>>
```

The **show cpu** command syntax is:

sh[ow] cpu

The console displays CPU information. **Primary CPU** is the current primary processor and 00 is its node number. **Active CPUs** are those CPUs actively running the console, and **configured CPUs** are those present in the system but not necessarily running the console (in Example 6-25, CPU 03 failed self-test).

6.5.25 Show Device

Displays device information for any disk/tape adapter or group of adapters.

Example 6-26 Show Device Command

```
P00>>> show device
polling for units on isp0, slot 0, bus 0, hose0...
dka200.2.0.0.0 DKA200 RZ26L 440C
dka200.4.0.0.0 DKA400 RZ26L 440C
polling for units on isp1, slot 1, bus 0, hose0...
d kb200.3.0.1.0 DKB300 RZ26L 440C
d kb200.5.0.1.0 DKB500 RZ26L 440C
d kb200.6.0.1.0 DKB600 RZ26L 440C
polling for units on isp2, slot 4, bus 0, hose0...
d kc200.1.0.4.0 DKC100 RZ26L 440C
d kc200.1.0.4.0 DKC200 RZ26L 440C
d kc200.1.0.4.0 DKC300 RZ26L 440C
polling for units on isp3, slot 5, bus 0, hose0...
d kd400.4.0.5.0 DKD400 RRD47 0064
d kd500.5.0.5.0 DKD500 RZ26L 440C
```

The **show device** command syntax is:

sh[ow] dev[ice] [<dev_name>]

Show device with no **<dev_name>** gives all devices in the system.

<dev_name> can be any adapter name (wild-carding is allowed). For example, **show device ms*** will display information on all MS7CC memory modules in the system.

6.5.26 Show EEPROM

The show EEPROM command allows you to display selected EEPROM information.

Example 6-27 Show EEPROM Command

1. P00>>> show eeprom serial # Display system serial
number.
System Serial Number = GAO1234567
2. P00>>> show eeprom manufacturing # Display manufactur-
ing information.
Module Serial Number = SG226LFH01
Module Unified 2-7-2.4 Part Number = -E2040-AA. M06

The **show eeprom** command syntax is:

sh[ow] ee[prom] <option>

where **option** is **diag_sdd**, **diag_tdd**, **field**, **halt**, **manufacturing**, or **symptom**.

6.5.27 Show <envar>

Show <envar> displays the current state of the specified environment variable.

Example 6-28 Show <envar> Command

```
1. P00>>> show auto_action
   auto_action          restart
   P00>>>
2. P00>>> show tta0_baud
   tta0_baud           9600
3. P00>>> show d_harderr
   d_harderr           halt
4. P00>>> show enable*      # Displays status
   enable_audit        OFF    # of enable_audit
5. P00>>> show interleave
   interleave          none
```

The **show envar** command syntax is:

sh[ow] <envar> or sh[ow] *

where **envar** is an environment variable name (see Table 6-3). Unambiguous abbreviations can be used for an environment variable name when using the **show <envar>** command. See the **set <envar>** command for related information.

6.5.28 Show Memory

The show memory command displays memory module information.

Example 6-29 Show Memory Command

```
P00>>> show memory
```

Set	Node	Size	Base Address	Intlv	Position
---	---	---	-----	-----	-----
A	7	4 Gb	00000000 00000000	2-Way	0

The **show memory** command syntax is:

sh[ow] m[emory]

In the above example, the memory module at node 7 is in an on-board two-way interleave indicated by the interleave set A. The total memory size is 4 Gbytes.

6.5.29 Show Network

The `show network` command displays the names and physical addresses of all known network devices in the system.

Example 6-30 Show Network Command

```
P00>>> show network
polling for units on tulip0, slot 12, bus 0, hose0...
ewa0.0.0.12.0  00-00-F8-25-90-A6  Twisted-Pair
polling for units on tulip1, slot 11, bus 0, hose0...
ewb0.0.0.11.0  08-00-2B-C3-9E-9B  Twisted-Pair
```

The **`show network`** command syntax is:

`sh[ow] ne[twork]`

There are no options or qualifiers.

6.5.30 Show SEEPROM

The show SEEPROM command allows you to display selected SEEPROM information.

Example 6-31 Show SEEPROM Command

1. P00>>> show seeprom field kftha0 # Displays field
LARS # = 0949820 # entered Labor Activity
Message = EEPROM update # Number and message.
2. P00>>> show seeprom manu kftha0 # Displays
manufacturing
information.
Module Serial Number = SG226LFH01
Module Unified 2-7-2.4 Part Number = -E2040-AA. M06
Module Device Type = 2000

The **show seeprom** command syntax is:

sh[ow] see[prom] <option> <device>

where **option** is **diag_sdd**, **diag_tdd**, **symptom**, **field**, **manufacturing**, or **serial** and **device** is **kfthaz**.

6.5.31 Show Time

The `show time` command allows you to display the CPU and TOY times.

Example 6-32 Show Time Command

```
P000>>> sho time
CPU 0/1 11:15:03 August 27, 1999
CPU 2/3 11:15:03 August 27, 1999
TOY 1 11:15:03 August 27, 1999
TOY 0 11:15:03 August 27, 1999
```

The **`show time`** command syntax is:

`sh[ow] tim[e]`

6.5.32 Start (OpenVMS or Tru64 UNIX)

The start command begins execution of an instruction at the address specified in the command string. The start command does not initialize the system.

Example 6-33 Start Command

```
P00>>> start 40000000 # Start processor at
                    # address 40000000.
```

The **start** command syntax is:

s[*start*] address

where **address** is the address the program counter is set to start execution. The **start** command is equivalent to **continue**, except you can specify the address at which to begin executing.

NOTE: *The **start** command should be used selectively since some console commands (for example, **deposit**, **set host**, **show device**, **show network**, and **test**) may corrupt the machine state so that execution of the current program may not resume successfully.*

6.5.33 Stop

The stop command halts a specified processor or device. Equivalent to the halt command.

Example 6-34 Stop Command

```
P00>>> stop 1          # Stop CPU 1.
```

The **stop** command syntax is:

stop [-drivers[<device_prefix>]][<processor_num>]

where **-drivers[device_prefix]** specifies the name of the device or device class to stop. If no device prefix is specified, then all drivers are stopped. The **<processor_num>** argument specifies the number of the processor to stop.

6.5.34 Test

The test command allows you to test the entire system (focusing on memory and I/O), a portion of the system (subsystem), or a specific device. By default, the entire system is tested.

Example 6-35 Test Command

1. P00>>> test
Console is in diagnostic mode
Complete Test Suite for runtime of 1200 seconds
Type ^C to stop testing

Configuring system...
polling for units on tulip0, slot 12, bus 0, hose0...
ewa0.0.0.12.0 00-00-F8-25-7F-2A Twisted-Pair
polling for units on tulip1, slot 11, bus 0, hose0...
:
:
-----Testing done -----
Shutting down drivers...
Shutting down units on tulip2, slot 12, bus 0, hose 4
:
:
P00>>>
2. P>>> test -q # Runs a system test. Since a test
 # run time was not specified, the
 # entire system will be tested
 # for approximately 10 minutes.
 # Status messages
 # will not be displayed.
3. P00>>> test ms7cc* # Tests all memory modules in the
 # system.
4. P00>>> test pci0 # Tests the pci0 I/O subsystem.

The **test** command syntax is:

t[est][-write][-nowrite "list"][-t time][-q][dev_arg]

where **<dev_arg>** specifies the target device, group of devices, or subsystem to test. A list of available devices and subsystem mnemonics can be obtained by issuing a **show configuration**, **show device**, or **show network** command. You would then issue the **test dev_arg** command to test the desired device. Table 6-8 lists the command options.

If no parameter is specified, the entire system, except for memory, is tested. Note that system testing performed by the test command is very different from that performed during power-on or reset. To execute systemwide self-test, use the **inititalize** command. See Chapter 5 for additional examples of the **test** command.

Table 6-8 Test Command Options

Option	Meaning
write	Selects writes to media as well as reads (read only is the default). Only applicable to disk testing (ignored otherwise).
-nowrite "list"	Used with -write to prevent selected devices or groups of devices from being written to.
-t time	Run time in seconds, following system sizing and configuration; default for system test is 600 seconds (10 minutes).
-q	Disables the status messages displayed by default as exerciser processes are started and stopped during testing. -q sets d_verbose to zero.

6.5.35 Type

The type command displays the contents of a specified file, one screen at a time. One common use of type is to display the event log, as shown in Example 6-36. Type a space to see the next screen, or Enter to see the next line.

Example 6-36 Type Command

```
P00>>> type el
01:14.63
01:14.63 Starting console on CPU 0
01:14.63 initialized idle PCB
01:14.63 initializing semaphores
01:14.63 initializing heap
01:14.63 initial heap 100000
01:14.63 Twilight zone at 7e000, length 82000(532480)
01:14.63 Heap size effc0(982976)
01:14.63 memory low limit = 1f0000
01:14.63 heap = 100000, effc0
01:14.63 initializing driver structures
01:14.63 initializing overlays
01:14.63 initializing idle process PID
01:14.63 initializing flash index
01:14.63 hf_init
01:14.63 hf_init_platform: console_dlb_flag = 0
01:14.63 hf_init_platform: from_adr = 13d694, index_count 150
01:14.63 hf_setup_platform
01:14.63 load XDELTA
01:14.63 Transferring to xdelta
01:14.63 initializing file system
01:14.63 DDB Startup, phase = 0
[--More-- (SPACE - next page, ENTER - next line)
```

6.5.36 Vga

The vga command tests the graphics console monitor.

Example 6-37 Vga Command

```
P00>>> vga test 0
      .
      .
      .
[displays character set on graphics monitor]
```

The **vga** command syntax is:

vga [reinit] [test <test_nbr> [<n>]]

where **reinit** reinitializes the graphics monitor, and **test_nbr** can have three values:

- test_nbr = 0** Display the entire character set on the graphics monitor <n> times.
- 1** Display color bars on the graphics monitor. You can then adjust the color values, if your monitor offers such adjustment.
- 2** Display the character E repeatedly on the screen. This allows you to determine whether the alignment of the screen is as it should be. You can then adjust the alignment, if your monitor offers such adjustment.

6.5.37 Comment (#)

A comment can be introduced using the # symbol. The entire comment is ignored.

Example 6-38 Comment (#) Command

1. P00>>> # This example illustrates the comment command.
P00>>>
2. P00>>> exam pmem:0400EC # Examine physical memory.
pmem: 000400EC D0FFFFFFD
P00>>>

Appendix A

OpenVMS and Tru64 UNIX Boot Options

Table A-1 lists the *Tru64 UNIX* options used with the **boot** command. Table A-2 lists the Alpha primary boot (APB) options used with the **boot** command for *OpenVMS*. These options allow you to control various phases of booting.

Table A-1 Tru64 UNIX Boot Options

Option	Function
a	Boots the system disk to multiuser mode.
d	Do full dumps.
i	Boot to interactive mode plus options (prompt for system image to boot and boot options).
s	Boot to single-user mode.

Table A-2 OpenVMS Boot Options

Hexadecimal Value	Function
1	Allows a conversational boot.
2	Maps XDELTA to a running system.
4	Stops the boot procedure at the initial system breakpoint.
8	Performs a diagnostic boot.
10	Stops the boot processor at the bootstrap breakpoints.
20	Omits the header from the secondary bootstrap image.
40	Inhibits memory testing.
80	Prompts for the name of the secondary bootstrap file.
100	Halts the system before the secondary bootstrap.
2000	Marks corrected read data error pages as bad.
10000	Enables debug messages in the APB.EXE, SYSBOOT.EXE, and EXEC_INIT.EXE files.
8200	Enables user messages in the APB.EXE, BOOT.EXE, and EXEC_INIT.EXE files.

Appendix B

Updating Firmware

Use the Loadable Firmware Update (LFU) utility to update system firmware. LFU runs without any operating system and can update the firmware on any system module. LFU handles modules on the TLSB bus (for example, the CPU) as well as modules on the I/O buses. You are not required to specify any hardware path information, and the update process is highly automated.

Both the LFU program and the firmware microcode images it writes are supplied on a CD-ROM. From the SRM console, you start LFU with the **boot** command.

A typical update procedure is:

1. Verify the **console** environment variable setting (*must be **serial***).
2. Boot the LFU CD-ROM. (Use the **show config** command to find the device name of the CD-ROM device.)
3. Use the LFU **list** command to show the revisions of modules that LFU can update and the revisions of update firmware.
4. Use the LFU **update** command to write the new firmware.
5. Exit.

Sections in this appendix are:

- Booting LFU
- List
- Update
- Exit
- Display and Verify Commands
- How to Update Corrupted Firmware
- Create

B.1 Booting LFU

LFU is supplied on the Alpha CD-ROM (Part Number AG-RCFB*-BE, where * is the letter that denotes the disk revision). Make sure this CD-ROM is mounted in the in-cabinet CD drive. Boot LFU from the CD-ROM.

Example B-1 Booting LFU from CD-ROM

```
P00>>> sho dev ❶
polling for units on isp0, slot 0, bus0, hose0...
dka400.4.0.0.0      DKA400      RZ26L  440C
polling for units on isp1, slot 1, bus0, hose0...
polling for units on isp2, slot 4, bus0, hose0...
polling for units on isp3, slot 5, bus0, hose0...
dkd400.4.0.5.0     DKD400      RRD47  0000
dkd500.5.0.5.0     DKD500      RZ26L  440C
P00>>> boot dkd400 ❷
Building FRU table.....
(boot dkd400.4.0.5.0 -flags 0,a0)
SRM boot identifier: scsi 4 0 5 0 400 ef00 81011
boot adapter: isp3 rev 2 in bus slot 5 off of kftia0 in
TLSB slot 8
block 0 of dkd400.4.0.5.0 is a valid boot block
reading 1150 blocks from dkd400.4.0.5.0
bootstrap code read in
Building FRU table.....
base = 200000, image_start = 0, image_bytes = 8fc00
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
```

The default bootfile for this platform is

```
[gs140]gs140_v55_10.exe
```

Hit <RETURN> at the prompt to use the default bootfile.

Bootfile: ❸

```
Starting Firmware Update Utility
Unpacking firmware files
```

```
***** 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.
Lfu	Restarts LFU.
Readme	Lists important release information.
Create	Make a custom Console Grom Image.
Update	Replaces current firmware with loadable data image.
Verify	Compares loadable and hardware images.
? or Help	Scrolls this function table.

WARNING

Before upgrading the "ARC" (AlphaBIOS) section of the console, make sure that the HAL.DLL on WNT boot disk is compatible with the "ARC" section of the console.

See release notes for details.

UPD> ④

- ① Use the **show device** command to find the name of the RRDCD drive.
- ② Enter the **boot** command to boot LFU from the RRDCD drive. This drive has the device name dkd400.
- ③ Press Enter for the default bootfile, or enter the directory and file name of the utility.
LFU starts, displays a summary of its commands, and issues its prompt (UPD>).
- ④ UPD> is the LFU prompt for command entry.

B.2 List

The list command displays the inventory of update firmware on the CD-ROM. Only the devices listed at your terminal are supported for firmware updates.

Example B-2 List Command

```
UPD> list
```

Device	Current Revision	Filename	Update Revision
cipca0	A315	cipca_fw	A420
kn7cg-ab0_arc	V5.68-0	kn7xx_arc	V5.68-0
kn7cg-ab0	G5.5-11	kn7xx_fw	V5.5-12
kn7cg-ab1_arc	V5.68-0	kn7xx_arc	V5.68-0
kn7cg-ab1	G5.5-11	kn7xx_fw	V5.5-12
		ccmab_fw	22
		cixcd_fw	7
		demfa_fw	2.1
		demna_fw	9.4
		dfxaa_fw	3.10
		kdm70_fw	4.4
		kfmsb_fw	2.4
		kzmsa_fw	5.6
		kzpsa_fw	A12

```
UPD>
```

The **list** command shows three pieces of information for each device:

- Current revision — The revision of the device's current firmware
- Filename — The name of the file that is recommended for updating that firmware
- Update revision — The revision of the firmware update

B.3 Update

The update command writes new firmware from the CD-ROM to the module. Then LFU automatically verifies the update by reading the new firmware image from the module into memory and comparing it with the CD-ROM image.

Example B-3 Update Command

```
UPD> update kn7cg-ab0 1
WARNING: updates may take several minutes to complete for each device.

Confirm update on: kn7cg-ab0_arc [Y/(N)] y 2
DO NOT ABORT!
kn7cg-ab0_arc Updating to V5.5-12... .Verifying V5.5-12... Passed

Confirm update on: kn7cg-ab0 [Y/(N)] y 2
DO NOT ABORT!
kn7cg-ab0 Updating to V4.9-20... Verifying V4.9-20... Passed. 3

UPD> update kzpsa0 4
WARNING: updates may take several minutes to complete for each device.

Confirm update on: kzpsa0 [Y/(N)] y
DO NOT ABORT!
kzpsa0 Updating to A10... FAILED. 5

UPD> exit

Errors occurred during update with the following devices:
kzpsa0

Do you want to continue to exit?
Continue [Y/(N)]y
Initializing...
[self-test display appears]
```

- ❶ This command requests a firmware update for a specific module. If you want to update more than one device, you may use a wildcard but not a list. For example, **update k*** updates all devices with names beginning with k, and **update *** updates all devices.
- ❷ LFU requires you to confirm the update. For processors, the first update to confirm is the AlphaBIOS firmware; the second is the SRM console firmware. In either case, the default is no.
- ❸ Status message reports update and verification progress.
- ❹ This is a second example.
- ❺ The update failed. This could indicate a bad device.

Continued on next page

CAUTION: *Never abort an update operation. Aborting corrupts the firmware on the module.*

Example B-3 Update Command (Continued)

```
UPD> update ⑥  
  
confirm update on: ⑦  
kzpsa0  
kzpsa1  
pfi0  
  
[Y/(N)]n  
  
UPD> update kzpsa0 -path cipca_fw ⑧  
WARNING: updates may take several minutes to complete for  
each device.  
  
Confirm update on:  kzpsa0  [Y/(N)]y  
  
                          DO NOT ABORT!  
  
Kzpsa0                  firmware filename 'kdm70_fw' is bad  
  
UPD>
```

- ⑥ When you do not specify a device name, LFU tries to update all devices.
- ⑦ LFU lists the selected devices to update and prompts before devices are updated.
- ⑧ In this next example, the **-path** option is used to update a device with different firmware from the LFU default. A network location for the firmware file can be specified with the **-path** option. In this example, the firmware filename is not a valid file for the device specified.

CAUTION: *Never abort an update operation. Aborting corrupts the firmware on the module.*

B.4 Exit

The exit command terminates the LFU program, causes system initialization and self-test, and returns the system to console mode.

Example B-4 Exit Command

```
UPD> exit ❶
Initializing...
    [self-test display appears]
P00>>> ❷

UPD> update kzpsa0
WARNING: updates may take several minutes to complete for each device.
Confirm update on:  kzpsa0  [Y/(N)]y

                DO NOT ABORT!

kzpsa0          Updating to A10...  FAILED.

UPD> exit

Errors occurred during update with the following devices: ❸
kzpsa0

Do you want to continue to exit? ❹
Continue [Y/(N)]y ❺
Initializing...
    [self-test display appears]
P00>>>
```


- ❶ At the UPD> prompt, **exit** causes the system to be initialized.
- ❷ The console prompt appears.
- ❸ Errors occurred during an update.
- ❹ Because of the errors, confirmation of the exit is required.
- ❺ Typing **y** causes the system to be initialized and the console prompt to appear.

B.5 Display and Verify Commands

Display and verify commands are used in special situations. Display shows the physical configuration. Verify repeats the verification process performed by the update command.

Example B-5 Display and Verify Commands

```
UPD> display ❶
```

	Name	Type	Rev	Mnemonic
TLSB				
0++	KN7CG-AB	8014	0000	kn7cg-ab0
2+	MS7CC	5000	0000	ms7cc0
5+	MS7CC	5000	0000	ms7cc1
8+	KFTHA	2020	0000	kftha0
C0 C0 PCI connected to kftha0				pci1
6+	DECchip 21040-AA	21011	0023	tulip2
A+	KZPSA	81011	0000	kzpsa0

```
UPD> verify kzpsa0 ❷
```

```
kzpsa0          Verifying A10... PASSED.
```

```
UPD>
```

- ❶ **Display** shows the system physical configuration. **Display** is equivalent to issuing the console command **show configuration**. Because it shows the slot for each module, **display** can help you identify the location of a device.
- ❷ **Verify** reads the firmware from the module into memory and compares it with the update firmware on the CD-ROM. If a module already verified successfully when you updated it, but later failed self-test, you can use **verify** to tell whether the firmware has become corrupted.

B.6 Create

The create command allows you to make a custom console image.

Example B-6 Create Command

```
UPD> create ❶

Console ARC image:
  File = obj\alpha\tl6ab Version = V5.68-0 Creation time =
26-NOV-1998 05:56:28
  Image size = 70000(458752)

Console GROM image:
  File = tl6 Version = V5.5-12 Creation time = 16-JUL-1999
11:50:35
  Overlays = 163 Image size = 13b5f4(1291764)

Flash free bytes 49ec(18924)

Select form of new Console Grom image
[Auto/Modify/Full/(A)] m ❷
Do you wish to include debug capability [Y/(N)]

Included overlays:
tl6          advcmd      advshell    arc          arccmd
ashshell     basiccmd    bitmap      boot         cipca
cpu_mem      cpu_tst     diag_tio    diagcmd     diagsupport
eecmd        eeprom      eisa        environ      ether
examine      fat         flash       floppy       fptest
fru          galaxy      hpc_diag    info         iso9660
isp1020      isp1020fw  kbd         kzpaa        lfu
lfu_drivers  memtest    mp_ex       mscp         net
netttest     nport      ods2        optional     pci
pci_diag     phase3     powerup     prcache     scsi
set          show       show_power  test         tiop_diag
toast        tulip      vga         x86          x86a

Flash free bytes 13fefc(1310460)

Do you wish to add, remove or list overlays? [a,r,l,n] - 1 ❸
```

Example B-6 Create Command (Continued)

Available overlays:

cixcd	dac960	debug	defpa	demfa
demna	dup	i82558	kdm70	kfesa
kfmsb	kfpsa	kgpsa	kzmsa	kzpsa
lamb_diag	mc_diag	simport	tga	xct
xdelta	xmi			

Included overlays:

tl6	advcmd	advshell	arc	arccmd
ashshell	basiccmd	bitmap	boot	cipca
cpu_mem	cpu_tst	diag_tio	diagcmd	diagsupport
eecmd	eeeprom	eisa	environ	ether
examine	fat	flash	floppy	fptest
fru	galaxy	hpc_diag	info	iso9660
isp1020	isp1020fw	kbd	kzpa	lfu
lfu_drivers	mentest	mp_ex	mscp	net
nettest	nport	ods2	optional	pci
pci_diag	phase3	powerup	prcache	scsi
set	show	show_power	test	tiop_diag
toast	tulip	vga	x86	x86a

Flash free bytes 13fefc(1310460)

Do you wish to add, remove or list overlays? [a,r,l,n] -

- ❶ When you select create, LFU first displays the ARC and Grom console parameters.
- ❷ LFU asks if you want to modify any parameter values. The default response is **no**.
- ❸ Enter **l** to list the available overlays; or select another function.

Appendix C

Running Configuration Utilities from the SRM Console

The **run** command has four options, as follows:

- **rcu** - Run the RAID Configuration Utility
- **swxcfw** - Update the firmware on the RAID Controller
- **eeromcfg** - Run the ISP1020 EEPROM Configuration Utility
- **util_cli** - Run the KZPSA Configuration Utility

This appendix describes some of the utilities that are available when using the **run** command. **Rcu** and **swxcfw** are on a floppy and require a floppy drive. **Eeromcfg** and **util_cli** are on a CD. Topics discussed here include running the RAID Configuration Utility (to automatically or interactively configure a RAID array) and the ISP1020 Configuration Utility. For further information on these utilities, and for information on the other two, refer to the user documentation that ships with each utility.

C.1 Configuring a RAID Storage Array

Once you have selected the **run** command and the **rcu** option, the utility displays the main menu:

```
+-----+
|                Main Menu                |
+-----+
|[01. View/Update Configuration]          |
| 02. Automatic Configuration              |
| 03. New Configuration                    |
| 04. Initialize Logical Drive             |
| 05. Parity Check                         |
| 06. Rebuild                             |
| 07. Tools                               |
| 08. Select SWXCR                        |
| 09. Controller Setup                    |
| 10. Diagnostics                         |
+-----+
| Choose this option to view, modify, and |
| view physical drive groups, logical     |
| drives, and hot spare drive            |
| assignments.                            |
+-----+
Use Cursor keys for selection, press <ENTER> to select,
<ESC> to Quit
```

Using the Standalone Utility

Use the arrow keys to move through the menu and highlight your selection, or you can type the highlighted letter (usually the first letter of the option). Press the Enter key to select the option. You can use the Enter key or the Return key interchangeably.

As you move the cursor from one option to the next, the message window changes to display a message for the highlighted option.

Exiting the Utilities

Press the Esc key twice from the main menu to exit the standalone utility. The utility prompts you to confirm that you want to exit. Select *Yes* and press the Enter key to exit.

Understanding the Configuration Methods

You can configure your array in either of the following ways:

1. **Automatically.** You can configure automatically only if:
 - a) You want to configure one RAID 5 logical RAID drive (requires between three and eight drives of same capacity)
 - b) You want to configure up to 8 JBODs of any capacity
2. **Interactively.** Configure interactively if you want to:
 - a) Create more than one drive group
 - b) Create more than one logical RAID drive
 - c) Specify RAID levels
 - d) Specify caching policy
 - e) Define a hot spare

Your RAID subsystem stores the configuration information on flash EEPROM/NVRAM on the RAID controller. If your RAID controller fails, you must restore your configuration from a backup copy on floppy disk, or create a new configuration to access the data on your subsystem.

Configuring Automatically

To configure automatically, you must do the following:

1. Select the Automatic Configuration option
2. Initialize the logical drives
3. Save your configuration to diskette

If you want to define a spare drive, you must configure interactively.

Selecting the Automatic Configuration Option

To configure automatically, follow these steps:

1. Select the *Automatic Configuration* option from the main menu and either one of the following happens depending upon whether a configuration currently exists. If a configuration currently exists, the utility displays a warning message saying that a valid configuration exists and if you proceed, you will destroy it. A confirmation window also appears. See Step 2. If no configuration exists, then the *Automatic Configuration* menu appears. Go to Step 3.
2. Select Yes to delete the current configuration (select No to quit without changing the configuration and to return to the main menu). The *Automatic Configuration* menu appears.
3. Use the arrow keys to highlight RAID 5 or JBOD and press the Enter key to select that configuration. The utility displays a window with this message: Do you want to have Write Cache enabled?
4. Press the Enter key to select No. This sets the cache policy to Write Through (the recommended policy). The utility displays a window with *Automatic Configuration* at the top of the screen and the message, Saving configuration, please wait ..., at the bottom of the screen. The utility then displays a screen with your system configuration information.
5. Press any key to return to the main menu.

```
-----  
                          Automatic Configuration  
Number of Logical drives = 1  
Raid Level                = 5  
Write Cache               = Enabled  
Number of Physical drives = 6  
Available Capacity       = 18245 MB  
                          Automatic configuration successfully done.  
Make certain to INITIALIZE Logical drive 0 before exiting  
this utility.  
                          Press any key to return to Main Menu  
-----
```

See the *StorageWorks RAID 200 Subsystems Controller Installation and Standalone Configuration Utility User's Guide* for information on how to initialize a RAID array.

Configuring Interactively

To configure interactively, do the following:

1. Create one or more drive groups
2. Create one or more logical RAID drives
3. Create a logical RAID drive of RAID level 0, 0 + 1, or 1
4. Define a spare drive
5. Specify caching policy

To configure interactively, you must do the following:

1. Create one or more drive groups

A drive group defines the drives that you want to work together as the available space for the logical RAID drives that you will create.

2. Create a logical RAID drive

A logical RAID drive allows your system to see and respond to a drive group as a single drive and defines how your system will store data in that space, based on the RAID level selected.

3. Add a hot spare (optional)

A hot spare drive is a drive available in your subsystem for the controller to automatically begin to use, in the event of a disk failure from a redundant logical RAID drive, to store the data of the failed drive.

4. Initialize the logical RAID drives

Initialize a logical RAID drive before you use it to ensure consistent RAID parity information.

5. Save your configuration to diskette

Your RAID subsystem stores the configuration information on flash EEPROM/NVRAM on the RAID controller. In the event that your RAID controller fails, you must restore your configuration from a backup copy on floppy disk, or create a new configuration to access the data on your subsystem.

Creating a Drive Group

A drive group is from one to eight drives that operate as a single drive. Determine how many drives to use for each drive group based on the following:

1. The RAID levels of the logical RAID drives you will create

See Table C-1 to see the minimum and maximum number of drives allowed for each RAID level. (You can create multiple logical RAID drives of varying RAID levels on a drive group.)

2. The amount of disk space you need.

Table C-1 shows the amount of storage available for unique data for each of the RAID levels. Also, consider the capacities of each of the individual disks. The capacities of each of the individual disks are affected by the other disks in a group. Disk drives of varying capacities are limited to the capacity of the lowest capacity drive in the drive group. Table C- 2 provides examples of how the capacity of each individual drive affects the capacities of the other drives in the group. To maximize your disk space resources, use drives of the same capacity in a drive group.

If you plan to use more than one drive group, remember:

1. Each disk can belong to only one drive group at a time.
2. You can create a maximum of eight drive groups.
3. Also, after you create drive groups you must arrange them in the order in which you want to use them. (You must use all the available space on a drive group before you can use another one.) So, consider what data you want to put on which drive group before arranging your drives, because you can only delete drive groups in the reverse order in which they were arranged. For example if you create drive groups a, b, and c, and arrange them in that order, then you must delete drive groups c and b before you can delete drive group a.

Creating a drive group requires the following procedures:

1. Defining a drive group
2. Arranging a drive group

First determine how you want to allocate your drive resources and then configure the drives in your array.

Table C-1 Number of Drives You Can Use in a Drive Group for Each RAID Level

RAID Level	Number of Drives in Drive Group	Amount of Storage Available for Unique Data (percent)	Data Redundancy
0	2 to 8	100	No
1	2	50	Yes
0 + 1	3 to 8	50	Yes
5	3 to 8	66 to 87	Yes
JBOD	1	100	No

*NOTE: The following equation determines the amount of storage available for unique information in a RAID 5 logical RAID drive:
Usable Space = Total Space * ((N-1)/N where N is the number of drives in the group.*

Table C-2 How the Capacity of Each Drive Affects the Capacity of the Drive Group

Drive Group	Drives	Drive's Capacity	Drive's Usable Capacity in This Group	Drive Group's Total Capacity
A	0	1 gigabyte	1 gigabyte	3 gigabytes (due to drive 0's capacity being 1 gigabyte)
	1	2 gigabytes	1 gigabyte	
	2	3 gigabytes	1 gigabyte	
B	0	2 gigabytes	2 gigabytes	6 gigabytes
	1	2 gigabytes	2 gigabytes	
	2	2 gigabytes	2 gigabytes	

Defining a Drive Group

Use this option to bind drives together into drive groups. To maximize your disk space resources, use drives of the same capacity in a drive group. To define a drive group, follow these steps:

1. Select the *New Configuration* option from the main menu and either one of the following things happens depending upon whether a configuration currently exists. If a configuration currently exists, the utility displays a warning message saying that a valid configuration exists and if you proceed, you will destroy it. A confirmation window also appears. Go to Step 2. If no configuration currently exists, then the utility displays the *New Configuration* menu. The *New Configuration* menu appears with the *Define Drive Group* option highlighted. Go to Step 3.
2. Select Yes to delete the current configuration (select No to quit without changing the configuration and to return to the main menu). The utility displays the *New Configuration* menu. The *New Configuration* menu appears with the *Define Drive Group* option highlighted.
3. Press the Enter key to select the Define Drive Group option. The utility displays a drive matrix that shows the drives connected to the adapter and the status of each. The Create Group function appears highlighted. If you are uncertain about the size of a particular drive, select the Device Information option on the Define Drive Group screen to check the size before you create a drive group.
4. Press the Enter key to select the *Create Group* option. The cursor moves to the drive matrix.
5. Position the cursor on each drive with a status of RDY (ready) that you want to add to the drive group, one at a time, and press the Enter key. The utility adds that drive to the drive group by changing its status from RDY to OPT (optimal); and assigning a group letter and a sequence number to the drive. To maximize the I/O performance of your multichannel RAID subsystem, locate each member of a drive group on a separate SCSI channel. This allows the RAID controller concurrent access to the disk drives.
6. Press the Esc key after you add all the drives that you want for this drive group.

7. Repeat Steps 3 through 6 of this procedure to create additional drive groups. If you plan to define a hot spare, leave at least one drive with RDY status. This drive must be of equal or greater capacity to the drives you assigned to drive groups. If you assign all the physical drives to drive groups, the system automatically highlights the *Arrange Group* option. Otherwise, press the Esc key and the cursor returns to the *Group Definition* menu.

NOTE: If you select a drive for the drive group by mistake or you want to redefine a drive group, highlight the Cancel Group option and press the Enter key. The cursor appears on the first drive in a group on the matrix. Move the cursor to the drive group you want to cancel and press the Enter key. You can only delete drive groups in the reverse order in which you arranged them.

For more information on arranging a drive group, creating a logical RAID drive, defining a spare drive, and specifying the caching policy, refer to the *StorageWorks RAID Array 200 Subsystems Controller Installation and Standalone Configuration Utility User's Guide*.

C.2 ISP1020/1040 Configuration Utility

When the EEROMCFG configuration utility is selected, after the "transferring control" message, a screen such as that shown in Example C-1 is displayed. This shows the ISP1020/1040 devices on the PCI device unit number you selected (if you are running the EEROMCFG utility from the SRM console). You use the arrow keys to move the brackets [] up and down to select a particular ISP1020/1040 device.

Example C-1 ISP1020/1040 Configuration Utility Opening Screen

```
-----
ISP1020/1040 EEROM Configuration Utility V2.1-101
-----
+=====Select ISP1020=====+
| Bus   Virtual Slot   Address |
| [0    0              p0]   |
| 0     1              p1   |
| 0     3              p3   |
| 0     4              p4   |
| 0     6              p6   |
| 0     7              p7   |
| 0     9              p9   |
| 0    10             p10   |
|-----|
| < (O)K >          <E(x)it > |
+=====+
```

Once you have chosen a specific ISP1020/1040, type "O" for OK and a screen like or similar to the one shown in Example C-2 is shown. Choose the option you wish. The H option allows you to modify host adapter parameters.

Example C-2 ISP1020/1040 Configuration Utility Second Screen

```
-----
ISP1020/1040 EEROM Configuration Utility V2.1-101
-----
+===== Configure EEROM Parameters =====+
| [(L)oad Default EEROM Parameters          ] |
| Edit EEROM (H)ost Adapter Parameters     |
| Edit EEROM (D)evice Adapter Parameters   |
| Enable (U)ltrafast [FAST-20] on all devices |
| Enable (F)AST-10 on all devices          |
| E(x)it                                   |
+=====+
```


Example C-3 shows an example of host adapter parameters for a KZPBA adapter. Table C-3 describes the host adapter parameters. (You can also refer to your adapter Installation Guide for further information.) You use the arrow keys to select a parameter to modify from the default and type in the modification. When parameters are as you wish, type “O” for OK.

Example C-3 Edit Host Adapter Parameters Example

```

+=====+Edit EEROM Host Adapter Parameters =====+
|
|      Parameters      Default  New
|  Fifo Threshold      3      [3_ ]
|  Host Adapter Enable  1      [1  ]
|  Initiator SCSI ID   7      [7  ]
|  Bus Reset Delay     1      [1  ]
|  Retry Count         0      [0  ]
|  Retry Delay         1      [1  ]
|  Asynchronous Data Setup Time 9 [9  ]
|  REQ/ACK Active Negation 1 [1  ]
|  Data Line Active Negation 1 [1  ]
|  Data DMA Burst Enable 1 [1  ]
|  Command DMA Burst Enable 1 [1  ]
|  Tag Aging           8      [8  ]
|  Low Termination Enable 1 [1  ]
|  High Termination Enable 1 [1  ]
|  Selection Timeout   250   [250 ]
|  Maximum Queue Depth 256   [256 ]
|  Single Ended or Differential 0 [0  ]
|
|-----|
|      |      < (O)K>      < (C)ANCEL >
|
+=====+

```

Table C-3 Host Adapter Parameters

Parameter	Default	Description	Reason to Change
FIFO Threshold	2 (KFTHA) 3 (KZPBA)	Sets the FIFO threshold point at which burst transfers are requested on the ISP1020/1040 adapter.	Optimize system performance. ¹
Host Adapter Enable	1	Determines whether the BIOS recognizes the ISP1020/1040 host adapter.	Disable the host adapter without physically removing it from the system.
Initiator SCSI ID	7	Sets the SCSI ID for the ISP1020/1040.	If a SCSI bus has multiple SCSI adapters.
Bus Reset Delay	1	Sets the delay (after resetting the SCSI bus) before the firmware initiates any SCSI activity.	Optimize system performance. ¹
Retry Count	0	Determines the number of times the firmware attempts to retry a selection timeout or a busy status.	Optimize system performance. ¹
Retry Delay	1	Sets the time (in 100 microsecond increments) that the firmware waits before reattempting an operation.	Optimize system performance. ¹
Asynchronous Data Setup Time	9	Sets the number of clock periods the ISP1020/1040 host adapter card waits after driving the SCSI data signals before asserting a SCSI bus acknowledge signal.	Optimize system performance. ¹

Table C-3 Host Adapter Parameters (Continued)

Parameter	Default	Description	Reason to Change
REQ/ACK Active Negation	1	Provide active pullup assist in single-ended mode. (the REQ and ACK signals are pulled up.) By enabling active negation, the ISP1020/1040 host adapter is less sensitive to an imperfect SCSI bus.	
Data Line Active Negation	1	Provide active pullup assist in single-ended mode. The SD17-0 and SDP1-0 signals are pulled up.) By enable active negation, the ISP1020 host adapter is less sensitive to an imperfect SCSI bus.	
Data DMA Burst Enable	1	When set to 1, performs burst transfers on the data DMA channel. When set to 0, data is transferred in nonburst mode with each cycle initiated by a new address phase.	Optimize system performance. ¹
Command DMA Bust Enable	1	When set to 1, performs burst transfers on the data DMA channel. When set to 0, data is transferred in nonburst mode with each cycle initiated by a new address phase.	Optimize system performance. ¹
Tag Aging	8	Ensures tagged commands are not lost in the target device. Tag aging is a backup to the timeout mechanism.	Optimize system performance. ¹

Table C-3 Host Adapter Parameters (Continued)

Parameter	Default	Description	Reason to Change
Low Term Enable	1		If termination not required on the board.
High Term Enable	1		If termination not required on high-order 8 bits, or not required on the board
Selection Timeout	250	Sets the selection phase timeout value (in microseconds)	Optimize system performance. ¹
Maximum Queue Depth	256	Specifies the maximum number of outstanding commands issued to each SCSI target. When the number is reached, new commands are returned with Queue Full Status.	Optimize system performance. ¹
Single-Ended or Differential	0		Not applicable for the ISP1020.

¹The default setting provides the best performance optimization for most system configurations. We recommend that you change this default setting only if it does not provide optimal performance for your system configuration.

Index

A

AC input box, 2-4, 2-14
AC power consumption, 2-3
AC power cord, 2-20
AC power indicators, 2-21
Address space, 6-27
Address space options, 6-28, 6-31
Airflow, cabinet, 2-23
Altitude, 2-3
Arc_enable environment variable,
6-10
Architecture, system, 1-6
ARC utility, 6-37
Attribute, environment variable, 6-10
to 6-13
Attributes, setting device, B-18
Auto_action environment variable,
4-4, 6-10

B

BA36R, 2-5
Baud rate, 6-13
Blowers, 2-4, 2-6
Boot command, 4-12, 4-14, B-2
Bootdef_dev environment variable,
6-10
Boot device,
default with factory installed
software, 4-7
setting default, 4-7
Boot_file environment variable, 6-10
Boot options, A-1
Boot_osflags environment variable,
6-10
Boot processor, 1-7, 5-12, 5-13, 5-17

Boot_reset environment variable,
6-11
Booting, 4-1
auto_action environment
variable, 4-4
boot device, 4-6
command, 4-12, 4-14, 5-4, 6-14
console environment variable,
4-3
device name for *OpenVMS* and
Tru64 UNIX, 4-7
OpenVMS, 4-12
os_type environment variable,
4-2
preparation, 4-2
Tru64 UNIX, 4-14
Booting LFU, 6-2
Booting *OpenVMS*, x-x
Booting *Tru64 UNIX*, y-y
Build -e command, 6-15
Build EEPROM command, , 6-15
Build -n command, 6-16
Build nonvolatile ROM command,
6-16
Build -s command, 6-17
Build SEEPROM command, 6-17
Bus fault, system, 2-19

C

Cabinet control logic (CCL) panel,
2-5, 2-10
Cabinet control system,
cabinet control logic panel, 2-6
CD-ROM drive, 2-4, 2-5, 4-15
operator control panel (OCP),
2-16, 2-18
floppy drive, 2-4, 2-5, C-1

- Cabinet,
 - Airflow, 2-22
 - clearances, 2-2, 2-23
 - control logic panel (CCL), 2-6
 - dimensions, 2-2
 - expander, 1-2, 2-5
 - front view, 2-3
 - rear view, 2-5
 - size, 2-2
 - system, 1-2
- Card cage, TLSB, 2-8
- CCL panel, 2-5, 2-10
- CD-ROM drive, 2-4, 4-9
- Centerplane, TLSB, 2-8
- Channels, PCI, 5-18, 5-19
- Circuit breaker, 2-6
- Clearance, cabinet, 2-2, 2-23
- Clear EEPROM command, 6-18
- Clear <envar> command, 6-19
- Clear screen command, 6-20
- Command language syntax, 6-4
- Comment (#) command, 6-59
- Configuration utilities, SCSI, 1-5
- Connector module, 3-11
- Console commands, 6-1
- Console environment variable, 4-3, 6-11
- Console load device (CD-ROM), 2-12
- Console printer, 2-5
- Console special characters, 6-2
- Console, SRM, 1-5
- Continue command, 6-21
- Control panel, operator (OCP), 2-16, 2-18
- Controls and indicators, 2-16
- Cooling system, 2-22
- CPU environment variable, 6-11
- Cpu_enabled environment variable, 6-11
- Cpu_primary environment variable, 6-11
- Crash command, 6-23
- Create command, 6-24
- Create command, LFU, B-14
- Ctrl/P, 4-5, 6-21

- Configuration rules,
 - cabinet, 2-5
 - TLSB card cage, 2-9
- Configuration utilities, B-1
- Console CD-ROM, 6-2
- Console commands, A-1
- CPU chip, 1-6
- CPU module
 - placement rules , 2-8, 2-9

D

- Date command, 6-25
- Device code, boot, 4-15
- Device name options, 6-28
- D_harderr environment variable, 5-31, 6-11
- Dimensions, cabinet, 2-2
- D_report environment variable, 5-31, 6-11
- D_softerr environment variable, 5-31, 6-11
- D_trace environment variable, 6-11
- D_harderr environment variable, 6-11
- DC distribution module, 2-15
- DECchip 21264, 1-7
- Deposit command, 6-25
- Device name format,
 - SRM console, 4-10
- Display command, LFU, B-12
- Dump_dev environment variable, 6-11
- DWLTPB, 1-7, 2-4, 2-6, 3-10, 5-6

E

- EEROMCFG, 1-5, C-1
- Electrical characteristics, 2-3
- Enable_audit environment variable, 6-12
- Environment variable, 6-9
 - arc_enable, 6-10
 - attributes, 6-10
 - auto_action, 4-14, 6-10
 - boot_dev, 4-12
 - boot_file, 4-12, 6-10
 - boot_osflags, 4-12, 6-10

- boot_reset, 4-12, 6-11
- bootdef_dev, 4-10, 4-11, 6-10
- console, 4-3, 6-11
- cpu, 6-11
- cpu_enabled, 6-11
- cpu_primary, 6-11
- d_harderr, 5-31, 6-11
- d_report, 5-31, 6-11
- d_softerr, 5-31, 6-11
- d_trace, 6-11
- dump_dev, 6-11
- enable_audit, 6-12
- ew* (network), 6-12
- interleave, 6-13
- os_type, 4-2, 6-13
- removal, 6-19
- simmm_callout, 6-13
- ttat0_baud, 6-13
- Environment variables
 - affecting test command, 5-31
 - ew* (network), 6-12
- Environmental characteristics, 2-3
- Error reports,
 - full, 5-38
 - Summary, 5-36
- Ethernet, 1-7, 3-7, 4-6, 5-25
- Event log, 6-57
- Examine command, 6-29
- Exit command, B-10
- Expander cabinet, 1-2, 2-5

F

- Factory installed software (FIS), 4-7
- Fault light, OCP, 5-10, 5-11
- Filler modules, 2-9
- Firmware, updating, B-1, B-6
- Flash ROM, 1-4
- Footprint, system, 2-2
- Front view, system, 2-4

G

- General register set, 6-28
- Graphics console subsystem, 1-7
- Graphics module, 3-11
- Graphics console vga command, 6-58

H

- H7506 power supply, 2-5
- Halt command, 6-32
- Hard error, 5-37
- Heat dissipation, 2-3
- Help command, 6-33
- <helpkey>, 6-34
- Hose, 3-3
- Hose error, 5-7
- Hose numbering scheme, 3-5

I

- I/O module, 3-3
- I/O connections, 2-11, 4-15
- I/O subsystem, 3-1
- Initialize command, 6-35
- Interleave environment variable, 6-13
- Interleaving memory, 2-9, 5-21

K

- Keyswitch, system, 2-16
- Keyboard port, 3-11
- KFE72 adapter, 1-7, 3-7, 3-10
- KFTHA module, 1-7, 2-9, 2-13, 3-5
 - description, 3-3
 - configuration rules, 2-9
- KN7CG module, 2-9
- KZPBA, 2-12, 3-7, 3-10
- KZPSA configuration utility, 6-38, C-1

L

- LEDs
 - DWLPB, 5-6
 - H7605 power supply, 2-20
 - Module, 5-10, 5-11, 5-17
 - Operator control panel (OCP), 2-18, 2-19
 - StorageWorks shelf, 5-8
 - TL5B modules, 2-4
- LFU, 1-5
 - booting, B-2
 - create command, B-14
 - display command, B-6

- exit command, B-10
- list command, B-4
- update command, B-6
- verify command, B-6
- List command, LFU, B-4
- Loadable Firmware Update (LFU) Utility, 1-5, B-1
- Logical partitioning, 2-5, 2-13

M

- Man command, 6-33
- Memory
 - interleaving, 1-7, 2-9, 5-21
 - module, 1-7, 2-9
 - size, 1-7, 2-9
 - configuration rules, 2-9
 - options, 1-7
- Module placement rules, 2-9
- Module troubleshooting, 5-3
- MS7CC module, 2-9

N

- Network (show) command, 6-49
- Node, 1-7, 4-15
- NRAM, 6-36
- Null command, 6-5

O

- On/off pushbutton, 2-17
- OpenVMS boot, 4-12
- Operator control panel (OCP), 2-16, 2-18, 5-10
- Options,
 - boot command, A-1, A-2
 - deposit command, 6-27
 - device name, 6-28, 6-31
 - examine command, 6-30
 - test command, 6-55
- Os_type environment variable, 4-2

P

- PCI configuration rules, 3-10
- Power filter, 2-9
- Power subrack, 2-14
- Power supplies, 2-4, 2-6, 2-14

- LEDs, 2-20
 - location, 2-4, 2-6
 - troubleshooting, 5-3
- Power-up troubleshooting, 5-2
- Prcache command, 6-36
- Processor module, 2-9
- Processor number, 6-32
- Program counter (PC), 6-27
- Pushbuttons, OCP, 2-16
- Processor system unit, 1-3

R

- RAID configuration utility (RCU), 6-37, C-1
- RAID controller, 1-5
- Rear view, 2-6
- Relative humidity, 2-3
- Reset pushbutton, 2-17
- Reset, system, 6-35
- RRD47, 4-9
- Run command, 1-5, 6-37

S

- SCSI-related utilities, 1-5
- Secondary processors, 1-7
- Secure pushbutton, 2-17
- Selecting a boot device, 4-6
- Self-test display, 5-12
- Self-test result lines,
 - C0, C1, C2, ..., Cn, 5-18
 - Identification line, 5-22
 - ILV and GB, 5-20
 - NODE # and TYPE, 5-14
 - ST and BPD, 5-16

Set

- auto_action, 4-4
- console command, 4-3
- cpu command, 6-39
- EEPROM command, 6-39
- <envar> command, 6-40
- os_type, 4-2
- SEEPROM command, 6-41
- time command, 6-42

Show

- boot command, 4-11
- configuration command, 3-6, 4-8, 5-24, 6-43
- cpu command, 6-44
- device command, 3-8, 4-14, 5-28, 6-45
- EEPROM command, 6-46
- <envar> command, 6-47
- memory command, 5-21, 6-48
- network command, 6-49
- SEEPROM command, 6-50
- time command, 6-51

Slot configuration, PCI, 3-10

Slot numbering, TLSB, 2-9

SRM console, 1-5

SRM command syntax, 6-4

SRM console special characters, 6-6

SRM environment variables, 6-9

Start command, 6-52

Stop command, 6-53

StorageWorks shelves, 2-5, 5-8

- configuration rules, 2-5

- LEDs, 5-8

System bus fault, 2-19

System components, 2-8

System overview, 1-2

T

Temperature, 2-3

Terminator module, 2-9

Test command, 5-30, 5-32

Test command options, 5-31, 6-55

Test command environment
variables, 5-31

TLSB card cage, 1-3, 2-4

- configuration rules, 2-9

- power filter, 2-9

- terminator module, 2-9

Troubleshooting, 5-1

Tru64 UNIX boot, 4-14

Type command, 6-56

U

Update command, LFU, B-5

Utilities, configuration, C-1

V

VAUX LED, 2-20

Vga command, 6-58

W

Wildcarding, 6-7, 6-8, 6-40

