

DEC 10000 AXP System VAX 10000 System Service Manual

Order Number EK-1002A-SV.001

This manual tells how to add or replace CPU and memory modules in a DEC 10000 AXP or VAX 10000 system.

First Printing, February 1993

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation.

Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software, if any, described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license. No responsibility is assumed for the use or reliability of software or equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Copyright © 1993 by Digital Equipment Corporation.

All Rights Reserved.
Printed in U.S.A.

The following are trademarks of Digital Equipment Corporation:

Alpha AXP	DECUS	VAXBI
AXP	DWMVA	VAXELN
DEC	OpenVMS	VMScluster
DECchip	ULTRIX	XMI
DEC LANcontroller	UNIBUS	The AXP logo
DECnet	VAX	

OSF/1 is a registered trademark of the Open Software Foundation, Inc.

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 computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such radio frequency interference when operated in a commercial environment. Operation of this equipment in a residential area may cause interference, in which case the user at his own expense may be required to take measures to correct the interference.

Contents

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

Chapter 1 Adding or Replacing CPUs and Memories

1.1	What Is Required	1-2
1.2	LSB Configuration Rules	1-4
1.3	Identifying the Kernel FRUs	1-6
1.4	Removing a Module from the LSB Card Cage	1-8
1.5	Inserting a Module in the LSB Card Cage	1-10
1.6	Verifying the System	1-12

Chapter 2 Servicing the CPU

2.1	System Parameters	2-2
2.2	How to Replace the Only Processor	2-6
2.3	How to Replace the Boot Processor	2-8
2.4	How to Add a New Processor or Replace a Secondary Processor	2-10
2.5	Build EEPROM Command	2-12
2.6	EEPROM Recovery—Hardware Requirements	2-14
2.7	EEPROM Recovery—Software Requirements and Setup ..	2-16
2.8	EEPROM Recovery—Procedure	2-18

Chapter 3 Updating Firmware

3.1	Bootting LFU on a DEC 10000 System	3-2
3.2	Bootting LFU on a VAX 10000 System	3-4
3.3	Show	3-6
3.4	List	3-8
3.5	Update	3-10

3.6	Exit	3- 12
3.7	Display and Verify Commands	3- 14
3.8	How to Update Corrupted Firmware	3- 16
3.9	How to Modify Device Attributes	3- 18

Appendix A Kermit Parameters

Examples

1- 1	Self- Test Display.....	1- 12
2- 1	Replacing a Single Processor	2- 6
2- 2	Replacing the Boot Processor	2- 8
2- 3	Adding or Replacing a Secondary Processor	2- 10
2- 4	Build EEPROM Command	2- 12
2- 5	Setting Up the Source System	2- 16
2- 6	Using Kermit to Downline Load FEPR0M Code	2- 18
3- 1	RRD42 LFU Booting	3- 2
3- 2	Booting LFU	3- 4
3- 3	Show Command.....	3- 6
3- 4	List Command	3- 8
3- 5	Update Command	3- 10
3- 6	Exit Command	3- 12
3- 7	Display and Verify Commands	3- 14
3- 8	Updating an "Unknown" Device	3- 16
3- 9	Modify Command	3- 18

Figures

1- 1	LSB Card Cage	1- 2
1- 2	LSB Configuration Rules	1- 4
1- 3	Sample Configuration—2 Processors, 4 Memories	1- 5
1- 4	Module Barcode	1- 7
1- 5	Removing a Module from the LSB Card Cage	1- 8
1- 6	Inserting a Module in the LSB Card Cage	1- 10
2- 1	Sample Hardware Requirements for Remote FEPR0M Recovery	2- 14

Tables

1	DEC 10000/VAX 10000 Documentation	ix
---	---	----

2	Related Documents	xi
1- 1	Field- Replaceable Units.....	1- 6
2- 1	EEPROM Environment Variables	2- 2

Preface

Intended Audience

This manual is written for Digital customer service engineers and self-maintenance customers servicing DEC 10000 AXP or VAX 10000 systems. This manual is a follow-on to *Basic Troubleshooting* and *Advanced Troubleshooting*.

Document Structure

This manual uses a structured documentation design. Topics are organized into small sections for efficient on-line and printed 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 three chapters and one appendix, as follows:

- **Chapter 1, Adding or Replacing CPUs and Memories**, describes how to add or replace these modules.
- **Chapter 2, Servicing the CPU**, explains how to restore the system environment to its state before the changes were made.
- **Chapter 3, Updating Firmware**, tells how to make updates to firmware on processor or I/O adapter modules.
- **Appendix A, Kermit Parameters**, shows what parameters must be set for using the Kermit program, which is used in down-line loading flash ROM code.

Conventions Used in This Document

Terminology. Unless specified otherwise, the use of "system" refers to either a DEC 10000 AXP or VAX 10000 system. The DEC 10000 AXP systems use the Alpha AXP architecture. References in text use DEC 10000 to refer to DEC 10000 AXP systems.

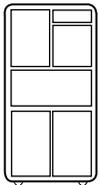
When a discussion applies to only one system, an icon is used to highlight that system. Otherwise, the discussion applies to both systems. Thus, the abstract for a module that applies only to DEC 10000 systems would look like this:



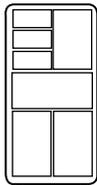
This section shows a sample boot of OpenVMS Alpha AXP from the RRD42 CD drive for DEC 10000 systems. The first step is issuing the show device command to determine the location of the RRD42.

Book titles. In text, if a book is cited without a product name, that book is part of the hardware documentation. It is listed in Table 1 along with its order number.

Icons. Icons such as those 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.



Front



Rear

Documentation Titles

Table 1 lists the books in the DEC 10000 and VAX 10000 documentation set. Table 2 lists other documents that you may find useful.

Table 1 DEC 10000/VAX 10000 Documentation

Title	Order Number
Installation Kit	EK-1000B-DK
<i>Site Preparation Guide</i>	EK-1000B-SP
<i>Installation Guide</i>	EK-100EB-IN
Hardware User Information Kit	EK-1001B-DK
<i>Operations Manual</i>	EK-1000B-OP
<i>Basic Troubleshooting</i>	EK-1000B-TS
Service Information Kit—VAX 10000	EK-1002A-DK
<i>Platform Service Manual</i>	EK-1000A-SV
<i>System Service Manual</i>	EK-1002A-SV
<i>Pocket Service Guide</i>	EK-1000A-PG
<i>Advanced Troubleshooting</i>	EK-1001A-TS
Service Information Kit—DEC 10000	EK-1002B-DK
<i>Platform Service Manual</i>	EK-1000A-SV
<i>System Service Manual</i>	EK-1002A-SV
<i>Pocket Service Guide</i>	EK-1100A-PG
<i>Advanced Troubleshooting</i>	EK-1101A-TS

Table 1 DEC 10000/VAX 10000 Documentation (Continued)

Title	Order Number
Reference Manuals	
<i>Console Reference Manual</i>	EK-70C0B-TM
<i>KA7AA CPU Technical Manual</i>	EK-KA7AA-TM
<i>KN7AA CPU Technical Manual</i>	EK-KN7AA-TM
<i>MS7AA Memory Technical Manual</i>	EK-MS7AA-TM
<i>I/O System Technical Manual</i>	EK-70I0A-TM
<i>Platform Technical Manual</i>	EK-7000A-TM
Upgrade Manuals	
<i>KA7AA CPU Installation Card</i>	EK-KA7AA-IN
<i>KN7AA CPU Installation Guide</i>	EK-KN7AA-IN
<i>MS7AA Memory Installation Card</i>	EK-MS7AA-IN
<i>KZMSA Adapter Installation Guide</i>	EK-KXMSX-IN
<i>DWLMA XMI PIU Installation Guide</i>	EK-DWLMA-IN
<i>DWMBB VAXBI PIU Installation Guide</i>	EK-DWMBB-IN
<i>H7237 Battery PIU Installation Guide</i>	EK-H7237-IN
<i>H7263 Power Regulator Installation Guide</i>	EK-H7263-IN
<i>BA654 DSSI Disk PIU Installation Guide</i>	EK-BA654-IN
<i>BA655 SCSI Disk and Tape PIU Installation Guide</i>	EK-BA655-IN
<i>Removable Media Installation Guide</i>	EK-TFRRD-IN

Table 2 Related Documents

Title	Order Number
General Site Preparation	
<i>Site Environmental Preparation Guide</i>	EK-CSEPG-MA
System I/O Options	
<i>BA350 Modular Storage Shelf Subsystem Configuration Guide</i>	EK-BA350-CG
<i>BA350 Modular Storage Shelf Subsystem User's Guide</i>	EK-BA350-UG
<i>BA350-LA Modular Storage Shelf User's Guide</i>	EK-350LA-UG
<i>CIXCD Interface User Guide</i>	EK-CIXCD-UG
<i>DEC FDDIcontroller 400 Installation / Problem Solving</i>	EK-DEMFA-IP
<i>DEC LANcontroller 400 Installation Guide</i>	EK-DEMNA-IN
<i>DEC LANcontroller 400 Technical Manual</i>	EK-DEMNA-TM
<i>DSSI VAXcluster Installation and Troubleshooting Manual</i>	EK-410AA-MG
<i>InfoServer 150 Installation and Owner's Guide</i>	EK-INFSV-OM
<i>KDM70 Controller User Guide</i>	EK-KDM70-UG
<i>KFMSA Module Installation and User Manual</i>	EK-KFMSA-IM
<i>KFMSA Module Service Guide</i>	EK-KFMSA-SV
<i>RRD42 Disc Drive Owner's Manual</i>	EK-RRD42-OM
<i>RF Series Integrated Storage Element User Guide</i>	EK-RF72D-UG
<i>TF85 Cartridge Tape Subsystem Owner's Manual</i>	EK-OTF85-OM
<i>TLZ06 Cassette Tape Drive Owner's Manual</i>	EK-TLZ06-OM

Table 2 Related Documents (Continued)

Title	Order Number
Operating System Manuals	
<i>Alpha Architecture Reference Manual</i>	EY-L520E-DP
<i>DEC OSF/1 Guide to System Administration</i>	AA-PJU7A-TE
<i>DECnet for OpenVMS Network Management Utilities</i>	AA-PQYAA-TK
<i>Guide to Installing DEC OSF/1</i>	AA-PS2DA-TE
<i>OpenVMS Alpha Version 1.0 Upgrade and Installation Manual</i>	AA-PQYSA-TE
<i>VMS Upgrade and Installation Supplement: VAX 7000-600 and VAX 10000-600 Series</i>	AA-PRAHA-TE
<i>VMS Network Control Program Manual</i>	AA-LA50A-TE
VMScusters and Networking	
<i>HSC Installation Manual</i>	EK-HSCMN-IN
<i>SC008 Star Coupler User's Guide</i>	EK-SC008-UG
<i>VAX Volume Shadowing Manual</i>	AA-PBTVA-TE
Peripherals	
<i>Installing and Using the VT420 Video Terminal</i>	EK-VT420-UG
<i>LA75 Companion Printer Installation and User Guide</i>	EK-LA75X-UG

Chapter 1

Adding or Replacing CPUs and Memories

This chapter provides information on how to remove and install processor and memory modules in DEC 10000 and VAX 10000 systems. Sections include:

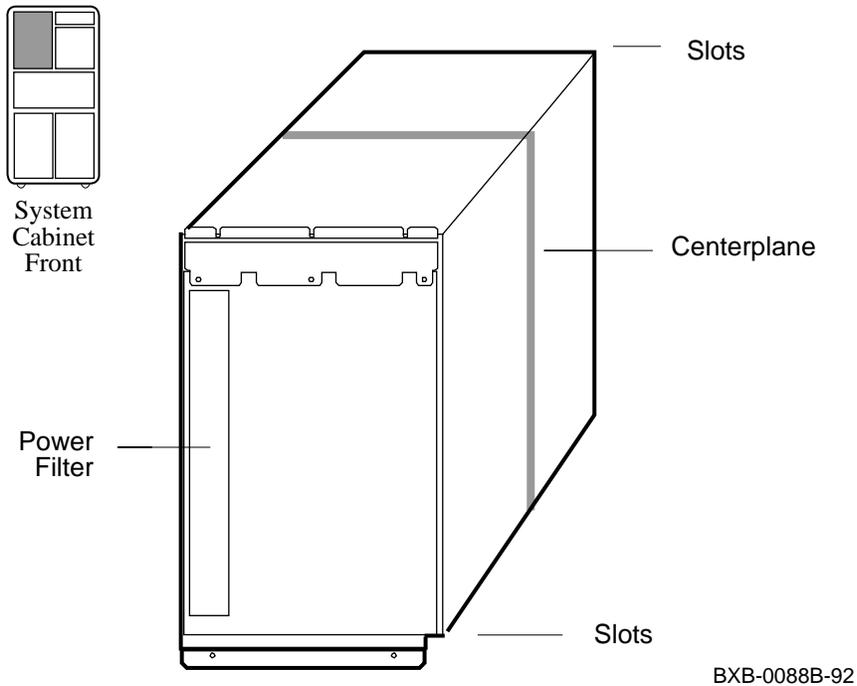
- What Is Required
- LSB Configuration Rules
- Identifying the Kernel FRUs
- Removing a Module from the LSB Card Cage
- Inserting a Module in the LSB Card Cage
- Verifying the System

For information on servicing I/O modules and batteries, see the *Platform Service Manual*; it also describes removal and replacement of the LSB centerplane and card cage (Part No. 70- 28574- 01), the IOP module (Part No. E2044- AA), and all power system FRUs.

1.1 What Is Required

Adding or replacing processor or memory modules is a simple operation. Afterward you must verify that the new modules are recognized in the system. You may need to set system parameters.

Figure 1-1 LSB Card Cage



Processor and memory modules reside in the LSB card cage, a centerplane card cage in the system cabinet with nine slots for modules. The LSB card cage always contains an IOP module, a clock module, and at least one processor and one memory module (see Figure 1- 1).

To add or replace modules, you will follow the steps in Sections 1.2 through 1.6. Then you will:

- Set system parameters to the original operating environment (Chapter 2).
- Upgrade firmware if required (Chapter 3).

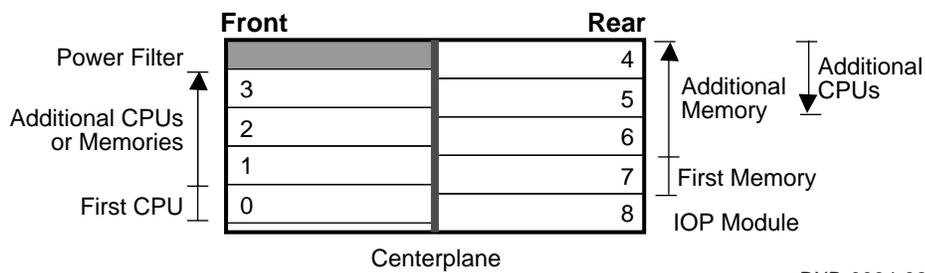
For more information:

Basic Troubleshooting
Advanced Troubleshooting

1.2 LSB Configuration Rules

The first CPU module is node 0, and the first memory module is at node 7. The LSB bus requires that an IOP module be at node 8. See Figure 1- 2.

Figure 1-2 LSB Configuration Rules



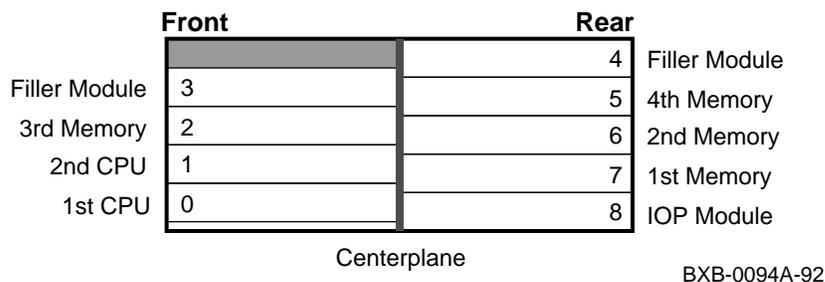
BXB-0094-92

The LSB card cage (see Figure 1-2) has nine slots. Slot numbers are equivalent to node numbers. Four slots are at the front of the cabinet (nodes 0 through 3, right to left), and five slots are at the rear (nodes 4 through 8, right to left).

A system can have up to six processors and up to seven memory modules, as space allows. The maximum memory configuration is bounded by the operating system support and the physical slots.

- The first CPU module is installed in node 0 (in the front at the far right).
- Additional CPU modules are installed in slots 1 through 5.
- The IOP module is in node 8.
- The first memory module is in node 7.
- Additional memory modules are installed next to filled slots. Modules are installed contiguously to the centerplane of the card cage. Install additional memories alternating between installation from the front and the back of the cabinet (see the example below of a 2 processor, 4 memory system).
- Install filler modules in all empty slots to direct the airflow through the card cage.

Figure 1-3 Sample Configuration—2 Processors, 4 Memories



For more information:
Platform Service Manual
KN7AA CPU Technical Manual
KA7AA CPU Technical Manual
MS7AA Memory Technical Manual

1.3 Identifying the Kernel FRUs

Table 1-1 lists the field-replaceable units (FRUs) for DEC 10000 and VAX 10000 systems that are discussed in this book.

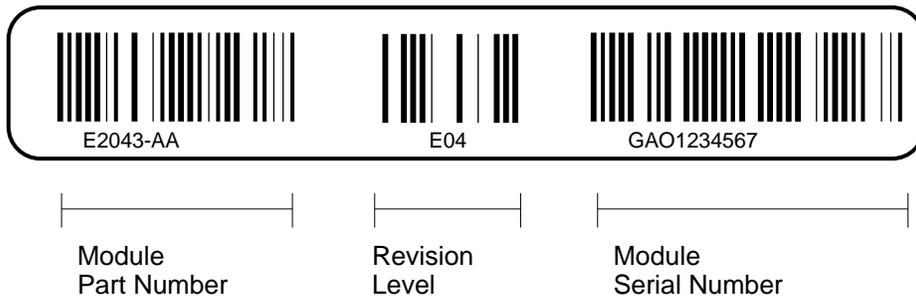
Table 1-1 Field-Replaceable Units

Option No.	Part No.	Description
KN7AA- AB	E2040- AB	DEC 10000 CPU module
KA7AA- AA	E2045- AA	VAX 10000 CPU module
MS7AA- AA	E2043- AA	64- Mbyte memory module
MS7AA- BA	E2043- BA	128- Mbyte memory module
MS7AA- CA	E2043- CA	256- Mbyte memory module
MS7AA- DA	E2046- AA	512- Mbyte memory module

Removal and replacement of the LSB centerplane and card cage (Part No. 70- 28574- 01), the IOP module (Part No. E2044- AA), and all power system FRUs are described in the *Platform Service Manual*.

Each memory or processor board is enclosed in the module case, protecting the module electronics from static discharge. A barcode label gives information about the module, including the module part number, revision level, and the module serial number (see Figure 1-4).

Figure 1-4 Module Barcode



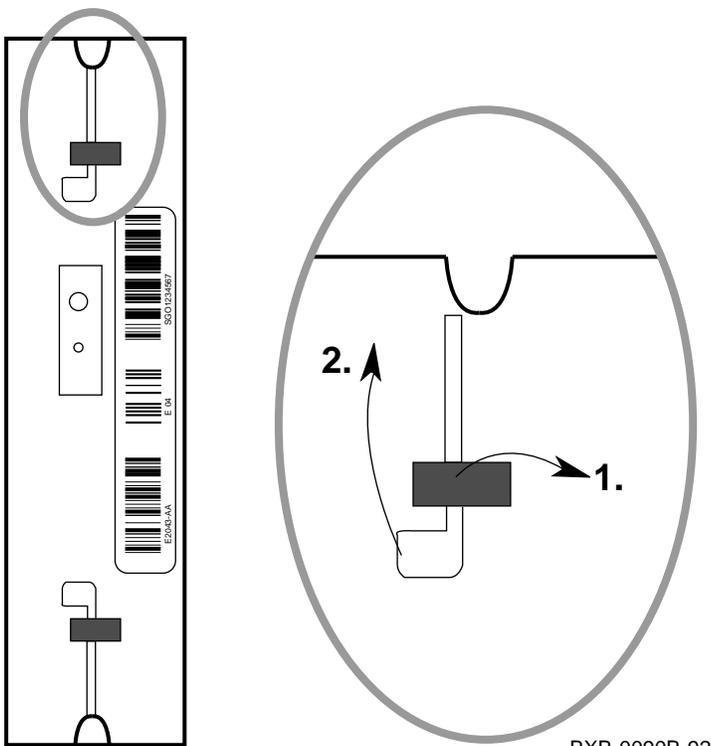
BXB-0089-92

For more information:
Platform Service Manual

1.4 Removing a Module from the LSB Card Cage

Use the following procedure to remove a module from the LSB card cage for replacement or reconfiguration.

Figure 1-5 Removing a Module from the LSB Card Cage



BXB-0090B-92

1. Perform an orderly shutdown of the system.
2. Turn the keyswitch on the front control panel to the Disable position and wait for the control panel yellow Fault LED to stop flashing. When the Fault LED stops flashing, power has been removed from the LSB backplane and you may safely proceed.
3. Open the cabinet door by holding the recessed handhold and pulling the door out toward you.
4. Put on the antistatic wrist strap.
CAUTION: You must wear a wrist strap when you handle any modules.
5. Release the plate covering the LSB card cage by loosening the two thumbscrews on the end of the plate. The plate is connected to the card cage by a cable; let the plate swing to one side.
6. On the module you are removing, use your thumbs to pull the two black restraining clips out and to the right (see Figure 1- 5, step 1). The clips snap when they open.
7. Pull both levers out at the same time until they are perpendicular to the front of the module (see Figure 1- 5, step 2). This frees the module from the backplane.
8. Holding the levers, pull on the module until it is out far enough to hold it underneath as you remove it.
9. When the module becomes free of the card cage, hold it with both hands, and place the module on an ESD pad in a safe area. If the module is being replaced, pack the module in the box from the new module.

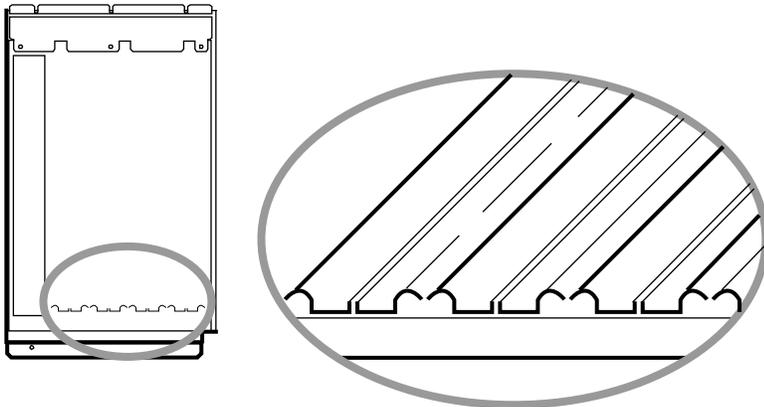
For more information:

Platform Service Manual

1.5 Inserting a Module in the LSB Card Cage

Use the following procedure when replacing or adding a module in the system card cage during maintenance or upgrade.

Figure 1-6 Inserting a Module in the LSB Card Cage



BXB-0091A-92

Follow Steps 1 through 6 in Section 1.4 and then:

1. If you are adding a module, remove the filler module from the slot where you will install the new module. Hold the filler module firmly on the vertical piece closest to you and gently pull it out toward you. Place it aside for return.
2. On the module to be inserted, pull out the two black restraining clips to the right and pull the two levers out until they are perpendicular to the front edge of the module. The clips snap open.
3. Pick up the module to be inserted, holding it securely with both hands.
4. Align the bottom tracks of the module with the tracks in the card cage slot (see Figure 1- 6). Align the top ridge of the module with the track at the top of the card cage slot.
5. Holding the module level, gently guide it into the card cage. If you encounter any resistance, check the alignment of the tracks and reinsert. As the module slides in, release your hand from the corner and guide the module with two hands as you insert it.

When the module is fully inserted, the front of the module will be flush with the card cage. Note that the module does not click when it is fully inserted.

6. Holding the two metal tabs, push both of them toward the edges of the module simultaneously. Check that the ends of the metal tabs are fitting into the guides of the card cage slot (see Figure 1- 6). Push the levers toward the module case.
7. Snap the black restraining clips across the levers to secure the module.
8. Check that filler modules are placed in all unused slots.
9. Replace the plate covering the card cage by tightening the two thumb-screws.
10. Close the cabinet doors.

1.6 Verifying the System

Power up the system and check that all processor and memory modules appear in the self-test display.

Example 1-1 Self-Test Display

```

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

      . . + . . . + . + . +      C0 XMI + ③
      + . . + . . + . . + . - . +      C1 XMI +
      . . . . . . . . . . . .      C2
      . . . . . . . . . . . .      C3

      . A1 A0 . . . . . ILV      ④
      .128 128 . . . . . 256Mb

```

```

Firmware Rev = v1.0-1625 SROM Rev = v1.0-0 SYS SN = GA01234567
P00>>>

```

BXB-0087-92

Power up the system by turning the keyswitch from Disable to either the Enable or Restart position. Power sequencing begins and the system runs self- test. Check the self- test display to make sure that the system recognizes the newly installed modules.

Example 1- 1 shows the self- test display of a system in which one processor and one memory module were added. The newly installed modules are at nodes 1 and 6, respectively.

- ❶ On the TYP line the **P** indicates that processors are at nodes 0 and 1. The **M** indicates that memory modules are at nodes 6 and 7.
- ❷ The plus signs on the ST lines indicate that the modules passed their self- tests.
- ❸ Two I/O channels are part of the system, both XMI buses. All adapters connected on the first channel's XMI passed self- test (see C0 XMI line). One adapter on the second channel's XMI failed self- test as indicated by the minus sign (see C1 XMI line).
- ❹ These two lines show the memory interleave set and size.
- ❺ The identification line shows the system firmware revision level and serial number.

If the primary processor is new to the system, the system serial number will not be shown. You must enter this information, along with any other system parameters in effect before the processor was changed. Chapter 2 explains how to set the parameters.

If any processors or memory fail self- test, refer to the appropriate *Advanced Troubleshooting* manual.

For more information:

Basic Troubleshooting
Advanced Troubleshooting

Chapter 2

Servicing the CPU

This chapter describes how to service a CPU in a DEC 10000 or a VAX 10000 system should it break or should new CPUs be added to a system. Some CPU firmware problems are covered in this chapter; others are covered in Chapter 3. Sections in this chapter include:

- System Parameters
- How to Replace the Only Processor
- How to Replace the Boot Processor
- How to Add a New Processor or Replace a Secondary Processor
- Build EEPROM Command
- FEPRM Recovery
 - Hardware Requirements
 - Software Requirements and Setup
 - Procedure

2.1 System Parameters

Several system parameters must be set during repair or when adding CPUs. Other system parameters may require setting depending upon how the customer wants the system configured.

Table 2- 1 EEPROM Environment Variables

Environment Variable	Default Value	Variable Description
auto_action	Halt	Specifies the action the system will take following an error halt. Values are: restart - Automatically restart. If restart fails, boot the operating system. boot - Automatically boot the operating system. halt - Enter console mode.
baud	9600	Sets the console terminal baud rate. Allowable values are 300, 600, 1200, 2400, 4800, and 9600.
boot_file	Null	The default file name used for the primary bootstrap when no file name is specified by the boot command.
boot_osflags	Null	Operating system flags used for booting the OS in specific ways, if none are specified by the boot command with the - flags qualifier.
boot_reset	Off or On	Resets system and displays self- test results during booting. Default is off for VAX 10000 and on for DEC 10000.
bootdef_dev		The default device or device list from which booting is attempted when no device name is specified by the boot command.
cpu	n	Node ID of the primary (n = 0, 1, 2, 3).

(Table 2-1 continued on page 2-4)

Table 2- 1 shows the permanent environment variables stored in EEPROM. Some of these variables must be set when either adding a CPU or replacing a broken one. You may view these variables by typing **show *** at the console prompt.

Volatile environment variables are initialized by a system reset; others are nonvolatile across system failures. Environment variables can be created and modified using the **create** and **set** commands, respectively.

The form of the **set** command is **set <envar> <value>** and causes the variable named to be changed in memory and in primary and secondary EEPROMs. The change takes place immediately.

Table 2- 1 EEPROM Environment Variables (Continued)

Environment Variable	Default Value	Variable Description
cpu_enabled	0xff	A bitmask determining which CPUs are enabled to run (leave console mode). If not defined, all available processors are considered enabled
cpu_primary	0xff	A bitmask indicating which CPUs are eligible to become the primary processor, following the next system reset. If not defined, all available processors are considered enabled.
d_harderr	Halt	Determines action taken following a hard error. Values are halt and continue . Applies only when using the test command.
d_report	Summary	Determines level of information provided by the diagnostic reports. Values are summary and full . Applies only when using the test command.
d_softerr	Continue	Determines action taken following a soft error. Values are halt and continue . Applies only when using the test command.
dump_dev	Null	Device to which dump file is written if system crashes (DEC 10000 only).
enable_audit	On	If set to on , enables the generation of audit trail messages used to track boot to determine location of boot failure.
interleave	Default	The memory interleave specification. Value must be default (memory configuration algorithm that attempts to maximize memory interleaving is used), none , or an explicit interleave list.
language	English	Determines whether system displays message numbers or message text. Default value is 36 (English).

Another important environment variable, not shown in Table 2- 1, because it is not a default variable but may be defined by the customer, is a nickname. Should the customer want to have a default boot path for a cluster and a different local one, a nickname variable may be used for that purpose. Nicknames are set by a console command of the form **create - nv old_disk** **dua0.0.0.4.0**. The **- nv** option indicates this nonvolatile environment variable will be stored in EEPROM. Once **old_disk** is so defined, one can type **boot old_disk** at the console prompt. Once the environment variable **old_disk** has been defined, it can be modified with the **set** command.

For more information:

Console Reference Manual

2.2 How to Replace the Only Processor

When replacing the only processor in a system, you must store the system ID and customized boot paths. If the customer changed console environment variables from the default values, you will want to set them as the customer wishes.

Example 2- 1 Replacing a Single Processor

```
>>> show device                # Shows device sizes in the
                                # system and the path to the
                                # devices.
polling for units on kzmsa0, slot 3, xmi0...
dka300.3.0.3.0                DKA300                RZ73
dka400.4.0.3.0                DKA400                RZ73
dkb400.4.1.3.0                DKB300                RRD42

>>> set eeprom serial
System Serial Number> GAO1234567 # Enter system serial no.
>>> set bootdef_dev dka400.4.0.3.0 # Sets the default
                                    # boot device.

>>> set boot_osflags 8001,2,0# Set custom boot flags;
                                    # operating system dependent.
>>> show *                       # Shows the environment
                                    # variables.

>>> create -nv old_disk ducl.0.0.11.2
                                    # Setting a possible boot
                                    # path to "old" disks. See
                                    # the Operations Manual for
                                    # details.

>>> set eeprom field             #
LARS> 01234567                 # Enter LARS number.
Message>                       # Enter message.
>>> boot
```

After you have removed and replaced the defective module, following the instructions in Chapter 1, take the following steps:

1. Power up the system. Self- test is run and you need to decide whether the new CPU module is functioning properly. If it is not, try reseating the new CPU and/or refer to the *Advanced Troubleshooting* manual; otherwise continue.
2. Retrieve the console printout of environment variables, system serial number, custom boot paths, and any other system- specific configurations saved in the *Site Management Guide*.

If this information has not been saved, you can find the system serial number on the barcode at the back of the system cabinet on the surface just below the AC input box. Details on boot paths and changed console environment variables you will need to get from the system manager.

3. Once you have collected the data you need, enter the console commands shown in Example 2- 1 that you need. For example, you may or may not need to use the **show device** command, which sizes the system and can be used to see what devices are available. The fields after the device name indicate the path to the device and are necessary when specifying the **bootdef_dev** environment variable.
 - a. You will definitely need to set the system serial number.
 - b. You will definitely need to set the **bootdef_dev**.
 - c. For example, you may need to set the **boot_osflags** depending upon the operating system used and how the system manager normally boots the system.
 - d. If you have a list of environment variables from the *Site Management Guide*, you will want to compare it against the output given by the **show *** command. If there are differences, you will want to change the variables using commands similar to those that follow.
 - e. You may need to set one or more special boot paths depending upon how the system manager configures the system.
4. Once the console environment is set up, verify the repair by booting the system. If there are alternative boot paths, you will want to make sure that all boot paths function properly.
5. Use the **set eeprom field** command to enter the 8- digit LARS number and a short message (up to 68 characters), stating the date and reason for service, into the EEPROM.
6. Boot the system and return control to the customer.

2.3 How to Replace the Boot Processor

In cases where the boot processor in a multiprocessing system is the CPU that is in need of repair, you need to manipulate which CPU receives data from the console terminal.

Example 2-2 Replacing the Boot Processor

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

      . . . . + . + . + . . . + . C0 XMI +
      . . . . + . + . + . . . + . C1 XMI +
      . . . . . . . . . . . . . C2
      . . . . . . . . . . . . . C3

      . A0 . . . . . ILV
      .256 . . . . . 256Mb

Firmware Rev = V2.0-1234 SROM Rev = V1.0-0 SYS SN = GAO1234567
KN7AA1: Firmware Revision Mismatch V1.0-1625 ②
P00>>>
P00>>> set cpu kn7aa1 # Set console to kn7aa1
P01>>> update -e kn7aa0 # Copy customized EEPROM envars
                          # from kn7aa1 to kn7aa0.
Updating kn7aa0's EEPROM done
P01>>> set cpu kn7aa0
P00>>> update -f kn7aa1 # Copy flash ROM from kn7aa0
                          # to kn7aa1
Update kn7aa1's FLASH ROMs [Y/(N)]? y
Updating kn7aa1's FLASH ROMs ...done
P00>>> set eeprom field #
LARS> 01234567 # Enter LARS number.
Message> # Enter message.
P00>>> boot
```

There are at least two factors to consider when replacing a primary CPU:

1. The desire to retain the system environment.
2. The possibility that the new CPU is at a higher or lower firmware revision than other CPUs in the system. **2** shows the mismatch message should the firmware differ between CPUs. Note that in this case the newer primary has a higher firmware revision than the secondary.

The procedure described here takes these into consideration.

After you have removed and replaced the defective primary CPU following the instructions in Chapter 1, take the following steps:

1. Power up the system. If self-test fails, refer to the appropriate *Advanced Troubleshooting* manual; otherwise continue.
2. Note the firmware revision of the new CPU.
3. Use the **set cpu** command to connect the console terminal to another CPU in the system.
4. Use the **update - e** command to copy the EEPROM environment variables from the secondary processor to the new primary processor. (The **update - e** command copies the system serial number and other parameters that can be set, as well as any additional information stored in the EEPROM to the target CPU.)
5. If there is a firmware revision mismatch, you should update the EEPROMs on the older CPUs. To do this, run the console on the newer CPU. Use the **set cpu** command to get to the newer version and then use the **update - f** command. Note that if there are several CPUs that need updating, you can use a wildcard to target them all, as in **update - f kn7aa***.
6. Use the **set eeprom field** command to enter the 8-digit LARS number and a short message (up to 68 characters), stating the date and reason for service, into the EEPROM.
7. Boot the system and return control to the customer.

*NOTE: If the customer used the **set cpu_primary** command to change the default which allows all CPUs in the system to become the primary, the **set cpu** command may not work.*

To allow all CPUs in the system to become the primary processor, use the **set cpu_primary ff** command. Modify the **cpu_primary** environment variable to the customer setting after performing the updates.

2.4 How to Add a New Processor or Replace a Secondary Processor

Add a new secondary in the slot to the left of the boot processor or other secondary processors.

Example 2-3 Adding or Replacing a Secondary Processor

```

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

      .  .  .  .  +  .  +  .  +  .  .  .  +  .  C0 XMI +
      .  .  .  .  +  .  +  .  +  .  .  .  +  .  C1 XMI +
      .  .  .  .  .  .  .  .  .  .  .  .  .  .  C2
      .  .  .  .  .  .  .  .  .  .  .  .  .  .  C3

      .  A0  .  .  .  .  .  .  .  ILV
      .256  .  .  .  .  .  .  .  256Mb

Firmware Rev = V1.0-1625 SROM Rev = V1.0-0 SYS SN = GAO1234567
KN7AA1:  Firmware Revision Mismatch V2.00-1234      ②
P00>>>
P00>>> update -e kn7aa1          # Copy customized EEPROM envars
                                   # from kn7aa0 to kn7aa1.
Updating kn7aa1's EEPROM done
P00>>> set cpu kn7aa1            # Set console to kn7aa1.
P01>>> update -f kn7aa0          # Copy flash ROM from kn7aa1 to
                                   # kn7aa0.
Update kn7aa0's FLASH ROMs [Y/(N)]? y
Updating kn7aa0's FLASH ROMs ...done
P01>>> set eeprom field
LARS> 01234567                  # Enter LARS number.
Message>                         # Enter message.
P01>>> boot

```

There are at least two factors to consider when adding or replacing a CPU:

1. The desire to retain the system environment.
2. The possibility that the new CPU is at a higher firmware revision than other CPUs in the system. **2** shows the mismatch message should the firmware differ between CPUs. In this case the newer secondary has a higher firmware revision than the older primary.

The procedure described here takes these into consideration.

After you have removed and replaced the defective CPU following the instructions in Chapter 1, take the following steps:

1. Power up the system. If self-test fails, refer to the appropriate *Advanced Troubleshooting* manual; otherwise continue.
2. Note the firmware revision of the primary CPU.
3. Use the **update - e** command to copy the EEPROM environment variables from the primary processor to the new secondary. (The **update - e** command copies the system serial number and other parameters that can be set, as well as any additional information stored in the EEPROM to the target CPU.)
4. Use the **set cpu** command to connect the console terminal to the new CPU.
5. If there is a firmware revision mismatch, you will want to update the EEPROMs on the older CPUs. Since you are already running the console from the newest CPU, use the **update - f** command. Note that if there are several CPUs that need updating, you can use a wildcard to target them all, as in **update - f kn7aa***.
6. Use the **set eeprom field** command to enter the 8-digit LARS number and a short message (up to 68 characters), stating the date and reason for service, into the EEPROM.
7. Boot the system and return control to the customer.

*NOTE: If the customer used the **set cpu_primary** command to change the default which allows all CPUs in the system to become the primary, the **set cpu** command may not work.*

To allow all CPUs in the system to become the primary processor, use the **set cpu_primary ff** command. Modify the **cpu_primary** environment variable to the customer setting after performing the updates.

2.5 Build EEPROM Command

Should the EEPROM become corrupted, you can use the build eeprom command to recover. The build eeprom command is the proper response to the console error messages shown in Example 2- 4. If the build eeprom command fails, return the module for repair.

Example 2- 4 Build EEPROM Command

```
EEPROM image failed to verify           # Checksum bad.
EEPROM environment parameters not set up # Ev area corrupt.
Fail to update EEPROM envar on CPU x     # Cannot write
                                           # EEPROM.

P00>>> build eeprom
Creating new EEPROM image
System Serial Number>    # Enter system serial number.
Module Serial Number>    # Enter module serial number.
Module Unified 2-5-2.4 Part Number> # Enter part number.
#
P00>>> set cpu kn7aa1    # Move to the next CPU should
# more than one be corrupt.

P01>>> build eeprom
Creating new EEPROM image
System Serial Number>    # Enter system serial number.
Module Serial Number>    # Enter module serial number.
Module Unified 2-5-2.4 Part Number> # Enter part number.
#
P01>>> initialize
```

Should an EEPROM become corrupted, the error message, EEPROM image failed to verify, is printed. Should this occur, use the **build eeprom** command to rebuild the EEPROM. When the EEPROM is rebuilt, all settings will revert to default settings. Follow Section 2.2 to customize environment variables.

The **build eeprom** command prompts you for several pieces of information. You can find this information in the following locations:

- The system serial number is on the barcode at the back of the system cabinet on the surface just below the AC input box.
- The 10-digit module serial number is the last number on the barcode found on the module itself. See Figure 1- 4.
- Most of the module unified 2- 5- 2.4 part number can also be found on the barcode. The first number is 80, the second for the DEC 10000 is E2040 and for the VAX 10000 is E2045, the third is a two-letter module variant, like AA, and the fourth is the hardware revision. A valid 2- 5- 2.4 part number is:

80- E2045- AA.AX01

The dashes should be entered when responding to the prompt.

If the EEPROM image fails to rebuild, send the module back for repair.

Once the image has been rebuilt, use the **initialize** command to set the environment variables to the default values. Customize the system, if necessary, using the **set** and **create** commands.

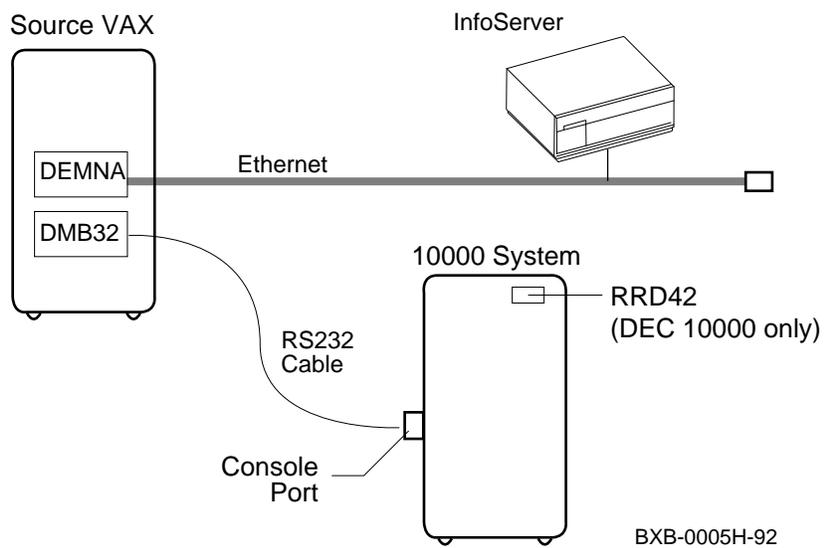
For more information:

Console Reference Manual

2.6 FEPRM Recovery—Hardware Requirements

When FEPRMs are corrupt and you do not have a CPU to use the update - f command, you may be able to recover the console and diagnostic code through the console terminal line. A serial line receive program in the serial ROM forces a prompt, AXP- or VAX-7000/10000- FRRC>, on the console terminal.

Figure 2- 1 Sample Hardware Requirements for Remote FEPRM Recovery



There are three methods for recovering console and diagnostic code. The first is to use the **update - f** command; the second is to use the Loadable Firmware Update (LFU) Utility; and the third is to downline load the console/diagnostic firmware into the damaged system and copy it into the FEPROMs. The use of the **update - f** command can only be done in a multiprocessing system and is documented in Sections 2.3 and 2.4. LFU can be used when the console is completely functioning and is described in Chapter 3. The third method, the least desirable, is described here.

Other pertinent information follows:

- The console terminal must be set at 9600 baud for the AXP- /VAX-7000/10000- FRRC> prompt to appear.
- On site or remote access to the system is required.
- The procedure may take 20 to 30 minutes to complete if there are no line problems.

What you need is:

- An independent source system that can logically connect to the damaged system through the console line. The system can be on site or remote. You do not need a DMB32 as shown in Figure 2- 1, but you need some hardware mechanism to connect to the target's console port.
- The source system must have access to an RRD42 or an InfoServer.
- The software program Kermit must reside on the source system.
- The AXP/VAX 10000 Console CD- ROM, with the console/diagnostic code on it. The file name for the console/diagnostic code is AXP7000_10000_CONSOLE_IMAGE.GROM for the DEC 10000 system and VAX7000_10000_CONSOLE_IMAGE.GROM for the VAX 10000 system.

The serial line receive program in the serial ROM of the CPU is called FRRC (FEPROM recovery code). Its function is to receive and store data sent down the console line.

2.7 FEPRM Recovery—Software Requirements and Setup

On the source system you need to "bind" the RRD42 or InfoServer to a virtual disk container, mount it, and set the terminal speed to that of the target console.

Example 2-5 Setting Up the Source System

```
$ set term/speed=9600/perm txa3: # Set DMB32 port to
                                # 9600 baud.
$ mcr ess$ladcp                  # Run LADCP.
LADCP> BIND VAX7000_V01           # This is the volume label
VAX7000_V01 is bound to DAD104    # from the CD.
LADCP> exit
$ mount/ov=id dad104             # Mount the CD-ROM.
$ dir dad104:[sys0.sysexex]      # Get a directory of the
.                                 # CD.
.                                 # Desired file name will
VAX7000_10000_CONSOLE_IMAGE.GROM # be AXP7000_10000_...
                                # or VAX7000_10000_...
.
$ Kermit                         # Enter Kermit from DCL.
Kermit-32> set file type binary
Kermit-32> set retry packet 5    # Set max retries/packet.
Kermit-32> set send time 5      # Set send timeout.
Kermit-32> show all             # Shows all Kermit params.
Kermit-32> set xxx              # Set other parameters.
```

Example 2- 5 illustrates the steps needed to prepare Kermit with Open-VMS VAX. What you do is:

1. First make sure that you have the hardware necessary to perform the task.
2. Make sure you have the correct CD- ROM for the damaged system.
3. Set the terminal speed on the source system to 9600. FRRRC only works at 9600 baud.
4. Run LADCP at the source system to "bind" the CD- ROM volume name to a virtual disk container pointed to by a logical name created by LADCP.
5. Next, set the correct parameters for Kermit. Example 2- 5 shows how to run Kermit from a DCL prompt and gives examples of setting the parameters. The **show all** command produces a list of Kermit parameters that should be compared to the parameters shown in Appendix A.
6. Set all parameters in accordance with Appendix A.

2.8 FEPRM Recovery—Procedure

After Kermit has been set up and you are ready to downline load the file, AXP or VAX7000_10000_CONSOLE_IMAGE.GROM, connect to the target system, prepare it to receive the file and then load it. The final steps are to copy the file into the FEPRMs and boot the system. Note that all commands are entered on the source system.

Example 2- 6 Using Kermit to Downline Load FEPRM Code

```
Kermit-32> connect txa5:          # Line = com path e.g., TXA5
    [should now get a FRRC> prompt on remote terminal]
VAX-7000/10000-FRRC> r          # Type FRRC receive command.
    Ctrl/] C                    # Escape sequence to return
                                # to Kermit.
Kermit-32> send
dad104:[SYS0.SYSEXE]VAX7000_10000_CONSOLE_IMAGE.GROM
    [Type Ctrl/A to get a brief status update]
    [Kermit responds OK when finished]
Kermit-32> status                # To get a status report.
Kermit-32> connect              # To reconnect to FRRC on the
                                # target system.
VAX-7000/10000-FRRC> c          # To verify checksum of image.
VAX-7000/10000-FRRC> p          # To copy program image into
                                # FEPRMs.
VAX-7000/10000-FRRC> i          # To reset node.
    [Selt-test display appears]
>>> Ctrl/] C                    # Escape sequence to return
                                # to Kermit.
Kermit-32> exit                 # To return to DCL.
$
```

NOTE: Updating using a laptop is exactly the same except the local MS-Kermit program must be configured appropriately. The procedure as far as FRRC is concerned is identical.

Assuming you have the correct CD-ROM in an InfoServer, you are now ready to connect to the damaged target system and downline load the code. Example 2- 6 illustrates a VAX 10000 recovery. Follow the same steps for the DEC 10000, using a **AXP7000_10000_console_image.grom** file.

1. At the Kermit prompt, connect to the target system. Here are two examples of connections:
 - connect txa5:** Logically connect to the target console line
 - connect lta1004:** Logically connect to the target console line
2. Once connected to the console, make the target system ready to receive the code by typing **r** (for receive) at the VAX- 7000/10000- FRRC> prompt on the source system.
3. Return to Kermit on the source system by typing **Ctrl/] C**.
4. Downline load the console and diagnostic code by using the Kermit **send** command. Here are several examples to help you with the location and name of the file:
 - send dadx:[sys0.sysexe]vax7000_10000_console_image.grom**
 - send -**
 - dua0:[directory_name]vax7000_10000_console_image.grom**If the file was copied to a disk earlier.
5. Once the code has been downline loaded, reconnect to the target.
6. Issue the following three VAX- 7000/10000- FRRC> commands:
 - c** causes the checksum to be verified for the code just loaded.
 - p** clears and then writes the code into the FEPROMs.
 - i** resets the system. The console prompt on the target system should appear and the flash ROM repair should be complete.
7. Return to Kermit on the source system by typing **Ctrl/] C**.
8. Exit Kermit.

Chapter 3

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 LSB bus (for example, the CPU) as well as modules on the I/O buses (for example, a CI controller on the XMI bus). 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. You start LFU on DEC 10000 systems by booting the RRD42. On VAX 10000 systems you start LFU by booting the Info-Server on your Ethernet.

A typical update procedure is:

1. Boot the LFU CD-ROM.
2. Use the LFU **show** command to indicate modules whose firmware needs to be updated.
3. Use the LFU **list** command if you want to check the firmware version numbers on the CD-ROM.
4. Use the LFU **update** command to write the new firmware.
5. Exit.

Sections in this chapter are:

- Booting LFU on a DEC 10000 System
- Booting LFU on a VAX 10000 System
- Show
- List
- Update
- Exit
- Display and Verify Commands
- How to Update Corrupted Firmware
- How to Modify Device Attributes

3.1 Booting LFU on a DEC 10000 System



LFU is supplied on the DEC 7000/10000 AXP Console CD-ROM (Part Number AG-PQW3*-RE, where * is the letter that denotes the disk revision). Make sure this CD-ROM is mounted in the RRD42 in-cabinet CD drive. Boot LFU from the CD-ROM.

Example 3-1 RRD42 LFU Booting

```
>>> show device ❶
polling for units on kzmsa, slot 1, xmi0...
dka100.1.0.1.0      dka100      RRD42
polling for units on kdm70, slot 6, xmi0...
dub1.1.0.6.0      R2TDYCS$DIA1  RF73
dub2.2.0.6.0      R2TDYCS$DIA2  RF73

>>> boot dka100 ❷
Booting...

                Copyright Digital Equipment Corporation
                    1992
                All Rights Reserved.
Loadable Environment Rev: V1.0-1625   Jul 12 1992  10:50:56

***** Loadable Firmware Update Utility *****
Version 2.1                               16-jun-1992

-----
Function      Description
-----
Display      Displays the system's configuration table.
Exit         Return to loadable offline operating environment.
List         Lists the device types and firmware revisions
             supported by this revision of LFU.
Modify       Modifies port parameters and device attributes.
Show         Displays device mnemonic, hardware and firmware
             revisions.
Update       Replaces current firmware with loadable data
             image.
Verify       Compares loadable and device images.
? or Help    Scrolls the function table.
-----
```

Function? ❸

- ❶ Use the **show device** command to find the name of the RRD42 CD drive.
- ❷ Enter the **boot** command to boot from the RRD42. The RRD42 has a device name of dka100.
- ❸ LFU starts, displays a summary of its commands, and issues its prompt (Function?).

3.2 Booting LFU on a VAX 10000 System



LFU is supplied on the VAX 7000/10000 Console CD-ROM (Part Number AG-PQW1*-RE, where * is the letter that denotes the disk revision). Make sure this CD-ROM is mounted in one of the system's InfoServers. Boot the Initial System Load (ISL) program, and select the service corresponding to the console CD-ROM.

Example 3-2 Booting LFU

```
>>> boot exa0 -flags 0,0,0 -file ISL_LVAX_V01 ❶
Resulting file is mopdl:ISL_LVAX_V01/exa0
..... Load complete!
      [boot information]

Network Initial System Load Function
Version 1.1
FUNCTION  FUNCTION
ID
1 -      Display Menu
2 -      Help
3 -      Choose Service
4 -      Select Options
5 -      Stop
Enter a function ID value: 3 ❷
OPTION          OPTION
ID
1 -      Find Services
2 -      Enter known Service Name
Enter an Option ID value: 1 ❸

Working
Servers found:: 3
Service Name Format:
Service Number
Service Name
Server Name
Ethernet ID
#1
INFO4$RZ57
INFO4
08-00-2B-26-A6-98

#2
6000_DIAG_H
INFO3
08-00-2B-16-04-D4
#3
VAX7000_V01
OPUS_ESS
08-00-2B-18-A9-75
Enter a Service Number or <CR> for more: 3 ❹
```

Copyright Digital Equipment Corporation
1992
All Rights Reserved.
Loadable Environment Rev: V1.0-1625 Jul 12 1992 10:50:56

***** Loadable Firmware Update Utility *****
Version 2.1 16-jun-1992

Function	Description
Display	Displays the system's configuration table.
Exit	Return to loadable offline operating environment.
List	Lists the device types and firmware revisions supported by this revision of LFU.
Modify	Modifies port parameters and device attributes.
Show	Displays device mnemonic, hardware and firmware revisions.
Update	Replaces current firmware with loadable data image.
Verify	Compares loadable and device images.
? or Help	Scrolls the function table.

Function?

5

- 1 Enter the **boot** command to boot from the InfoServer. Note that the ISL file name must be typed in upper case. The final two characters are the ISL file version, which you can read from the last two characters of the volume label printed on the CD-ROM.
- 2 Enter 3, to select Choose Service from the Function menu.
- 3 Enter 1 to select Find Services from the Option menu.
- 4 Enter the number of the service named VAX7000_Vnn. This service name is the volume label printed on the CD-ROM. In this example, service number 3 supplies the console CD-ROM.
- 5 LFU starts, displays a summary of its commands, and issues its prompt (Function?).

3.3 Show

The show command shows the current revision of firmware and hardware for every module in the system that contains microcode. In the display, each module that needs to be updated is indicated by a plus sign (+) following the device mnemonic.

Example 3-3 Show Command

```
Function? show 1
Device Mnemonic(s)? ? 2
-----
Valid Device
  Entries          Selected function is:
-----
Device Mnemonic#   performed to a single device.
Device Mnemonic*   performed to all devices of the same type.
* or All           performed to all devices in the system.
Exit               not performed. Program returns to Selection
                  prompt.
-----
Device Mnemonic(s)? exit 3
Function? sho * 4 5

```

	Firmware Revision	Hardware Revision	
kn7aa0 +	1.0	E04	
ms7aa0	--	--	not supported.
iop0	--	--	not supported.
xmi0	--	--	not supported.
kdm700	3.0	Cannot be read	
demna0	6.8	Cannot be read	
demfa0	--	--	not supported.
cixcd0 +	69.0	A01	
kzmsa0	4.2	F01	

'+' indicates the update firmware revision
is greater than the adapter's firmware revision.

```
Function?
```

- ❶ If you type just the command **show** without a device mnemonic, LFU prompts for the device mnemonic. All the commands that require device mnemonics will prompt.
- ❷ If you enter **?** (or **help**) for the device, a table displays the syntax for specifying devices. All the commands that require device specifications use this syntax. Note the use of wildcards. For example, **show kdm70*** would display all KDM70 controller modules.
- ❸ If you enter an **exit** command at the device prompt, LFU returns to the function prompt for another command.
- ❹ The most useful form of the command is **show *** which displays every module in the system. Note in this example that the CPU and CIXCD modules require updating. (In other words, the + means that the firmware version on the CD-ROM is higher than the version on the module.)
- ❺ VAX 10000 systems do not support kn7aa and kzmsa. The following devices show up in the display instead:

ka7aa0	+	1.1	E04
kfmsa0		5.6	A04

3.4 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 3-4 List Command

Function? l ❶

Loadable Firmware Update Utility Version 2.1 ❷

Name	Mnemonic	Update Firmware Revision	Update Hardware Revision
CIXCD	cixcd*	70.0	A01 - A01
KDM70	kdm70*	3.0	All Revisions
KN7AA	kn7aa*	1.1	All Revisions
KZMSA	kzmsa*	2.0	All Revisions

Function?

- ❶ The **list** command shows the revisions of firmware corresponding to the revisions of hardware for each device. (There may be several hardware revisions for a particular device, but only one firmware revision corresponds to any hardware revision.) Comparing the output of the **list** and **show** commands helps you understand which devices should receive firmware updates.
- ❷ VAX 10000 systems do not support kn7aa and kzmsa. The following devices show up in the display instead:

KA7AA	ka7aa*	1.1	All Revisions
KFMSA	kfmsa*	5.6	A02 - A01
		4.0	All Remaining Revs

3.5 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 back into memory and comparing it with the CD-ROM image.

Example 3-5 Update Command

```
Function? update kn7aa0 cixcd0          ❶
Update kn7aa0? [Y/(N)] y                ❷
WARNING: updates may take several minutes to complete for each device.
DO NOT ABORT!
kn7aa0 Updating to 1.1... Reading Device... Verifying 1.1...PASSED. ❸
Update cixcd0? [Y/(N)] y                ❹
WARNING: updates may take several minutes to complete for each device.
DO NOT ABORT!
cixcd0 Updating to 70.0... Reading Device... Verifying 70.0... PASSED.
Function? update *                        ❺
```

	Name	Type	Rev	Mnemonic	FW Rev	HW Rev
LSB						
0+	KN7AA	(8001)	0000	kn7aa0	1.1	E04
7+	MS7AA	(4000)	0000	ms7aa0	N/A	A01
8+	IOP	(2000)	0001	iop0	N/A	A
C0 XMI				xmi0		
8+	DWLMA	(102A)	A5A6	dwlma0	N/A	A
E+	DEMNA	(0C03)	060B	demna0	6.8	

```
Update ALL devices? [Y/(N)] y          ❻
WARNING: updates may take several minutes to complete for each device.
DO NOT ABORT!
kn7aa0 Updating to 1.1... Reading Device... Verifying 1.1...PASSED
ms7aa0 not supported.
iop0 not supported.
xmi0 not supported.
demna0 firmware rev is greater or equal to update rev.
```

Continue? [Y/(N)] y

7

WARNING: updates may take several minutes to complete for each device.

DO NOT ABORT!

demna0 Updating to 6.6... Reading Device... Verifying 6.6... PASSED.

Function? update demna*

8

Update all demna?

[Y/(N)] n

Function?

- 1** This command specifically requests firmware updates for the CPU and CIXCD modules. Note the syntax of a device list, separated by spaces.
- 2** LFU requires you to confirm each update, if you named the modules specifically.
- 3** Status message reports update and verification progress.
- 4** LFU prompts for each device in turn.
- 5** This is a second example. When you specify the * wildcard, LFU tries to update all modules.
- 6** LFU prints the configuration table and prompts before all devices are updated.
- 7** This message appears because, in this example, the firmware on the DEMNA module is not at a lower revision level than the firmware image on the CD-ROM. You can still request LFU to perform the update. If the module version is equal to the update firmware, you may have previously tried the update (making the module and CD-ROM firmware images the same revision). However, if the verification process had reported an error, you can repeat the update. Also, this feature allows you to update a module with an older revision of firmware.
- 8** This is another example, using a wildcard to request LFU to update all DEMNA adapters in the system. When you use a device mnemonic followed by a wildcard *, LFU prompts once for all devices of the same type. A wildcard alone updates all devices.

CAUTION: Never abort an update operation; you will corrupt the firmware on the module.

3.6 Exit

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

Example 3-6 Exit Command

```
Function? show
Device Mnemonic(s)? exit ❶

Function? exit ❷

Initializing...

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

      . . . . + . + . + . . . + . C0 XMI +
      . . . . + . + . + . . . + . C1 XMI +
      . . . . . . . . . . . . . C2
      . . . . . . . . . . . . . C3

      . A0 . . . . . ILV
      .256 . . . . . 256Mb
Firmware Rev = V1.0-1625 SROM Rev = V1.0-0 SYS SN = GA01234567
P00>>> ❸
```

- ❶ From within the "Device Mnemonic(s)?" prompt, **exit** returns to the Function prompt.
- ❷ At the Function prompt, **exit** causes the system to be initialized.
- ❸ The console prompt appears.

3.7 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 3-7 Display and Verify Commands

```
Function? disp 1
      Name      Type      Rev      Mnemonic  FW Rev  HW Rev
LSB
0+   KN7AA      (8001)   0000     kn7aa0    1.0     E04
5+   MS7AA      (4000)   0000     ms7aa0    N/A     A01
7+   MS7AA      (4000)   0000     ms7aa0    N/A     A01
8+   IOP        (2000)   0001     iop0      N/A     A

C0 XMI
      Name      Type      Rev      Mnemonic  FW Rev  HW Rev
8+   DWLMA      (102A)   A5A6     dwlma0    N/A     A
C+   KDM70      (0C22)   IE11     kdm700    3.0
E+   DEMNA      (0C03)   060B     demna0    6.6

C1 XMI
      Name      Type      Rev      Mnemonic  FW Rev  HW Rev
1+   ??????    (0000)   0000     unknown0
8+   DWLMA      (102A)   A5A6     dwlma0    N/A     A
A+   CIXCD      (0C05)   EB11     cixcd0    69.0    A01

Function? verify kdm700 2

kdm700 Reading Device... Verifying 3.00... FAILED.
At Address 3d830 3 Hardware data 41570020 Update data 20200020
At Address 3d834 Hardware data 4E494E52 Update data 20202020
At Address 3d838 Hardware data 54203A47 Update data 54202020
At Address 3d844 Hardware data 65696620 Update data 20202020
At Address 3d848 Hardware data 7420646C Update data 302E3356
At Address 3d84c Hardware data 20747365 Update data 20202020
At Address 3d988 Hardware data 69662020 Update data 20202020
At Address 3d98c Hardware data 00646C65 Update data 00202020
At Address 3d998 Hardware data 73657420 Update data 20202020
At Address 3d99c Hardware data 65722074 Update data 65722020 4
```

- ❶ **Display** shows the system physical configuration. **Display** is equivalent to issuing the console command **show configuration**. Because it shows the LSB slot for each module, **display** can help you identify unknown devices.
- 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.
- ❸ The address displayed for a failed compare is relative to the beginning of the update firmware image.
- ❹ **Verify** terminates after 10 comparisons fail.

3.8 How to Update Corrupted Firmware

If LFU identifies a device as unknown, the firmware on the module is corrupted. The update command allows you to specify the correct device type so that new firmware can be written to the module.

Example 3-8 Updating an "Unknown" Device

```
Function? sho * ❶
```

		Firmware Revision	Hardware Revision	
kn7aa0	+	1.0	E04	
ms7aa0		--	--	not supported.
iop0		--	--	not supported.
xmi0		--	--	not supported.
kdm700		3.0	Cannot be read	
demna0		--	--	not supported.
unknown0	❷	--	--	Updates only.
cixcd0	+	69.0	A01	

'+' indicates the update firmware revision is greater than the adapter's firmware revision.

```
Function? disp ❸
```

	Name	Type	Rev	Mnemonic	FW Rev	HW Rev
LSB						
0+	KN7AA	(8001)	0000	kn7aa0	1.0	E04
7+	MS7AA	(4000)	0000	ms7aa0	N/A	A01
8+	IOP	(2000)	0001	iop0	N/A	A
C0 XMI						
8+	DWLMA	(102A)	A5A6	dwlma0	N/A	A
C+	KDM70	(0C22)	1E11	kdm700	3.0	
E+	DEMNA	(0C03)	060B	demna0	6.8	
C1 XMI						
1+	?????	(0000)	0000	unknown0	❹	
8+	DWLMA	(102A)	A5A6	dwlma1	N/A	A
A+	CIXCD	(0C05)	EB11	cixcd0	69.0	A01

```
Function? update unknown0 ❺
Enter device name or 'exit' to skip this device.
Device name? cixcd ❻
Hardware revision? A01 ❼
WARNING: updates may take several minutes to complete for each device.
```

DO NOT ABORT!

```
unknown0 Updating to 70.0... Reading Device... Verifying 70.0... PASSED.
```

```
⑧  
Function? exit          ⑨
```

```
Initializing...  
F E D C B A 9 8 7 6 5 4 3 2 1 0 NODE #  
[self-test map appears]  
      . A0 . . . . . ILV  
      .256 . . . . . 256Mb  
Firmware Rev = V1.0-1625 SROM Rev = V1.0-0 SYS SN = GAO1234567
```

```
>>> sho config  
Name      Type      Rev      Mnemonic  
LSB  
0+ KN7AA   (8001)   0000    kn7aa0  
7+ MS7AA   (4000)   0000    ms7aa0  
8+ IOP     (2000)   0001    iop0  
  
C0 XMI  
8+ DWLMA   (102A)  A5A6    dwlma0  
C+ KDM70   (0C22)  IE11    kdm700  
E+ DEMNA   (0C03)  060B    demna0  
  
C1 XMI  
1+ CIXCD   (0C05)  EB11    cixcd0 ⑩  
8+ DWLMA   (102A)  A5A6    dwlma1  
A+ CIXCD   (0C05)  EB11    cixcd1
```

- ① Issue the **show** command.
- ② The display indicates an unknown device — LFU is unable to recognize device type.
- ③ Issue the **display** command. Then you can identify the unknown device by looking at the physical configuration.
- ④ **Display** shows that the unknown device is in slot 1 of the second XMI bus. You inspect the module and identify it as a second CIXCD with hardware revision A01.
- ⑤ Issue the command **update unknown0**.
- ⑥ LFU prompts you for only the device name (not mnemonic). The mnemonic is the device name plus a unique number which is assigned by the system. The device name in this example is cixcd.
- ⑦ If the device has several hardware versions supported by different firmware images, LFU prompts you for the hardware version so that it will be updated with the correct firmware image.
- ⑧ Status message indicates that the update succeeded.
- ⑨ To make the device known, initialize the system by exiting LFU.
- ⑩ Initialization has made the device known to the system. The previously unknown device is now assigned device mnemonic cixcd0 by the system. The previous cixcd0 is now cixcd1.

3.9 How to Modify Device Attributes

The modify command can change parameters stored in EEPROM on the following devices: KZMSA (DEC 10000 system), KFMSA (VAX 10000), DEC LANcontroller 400 (DEMNA), and CIXCD. The attributes are specific to each device.

Example 3-9 Modify Command

```
Function? modify cixcd0 1
cixcd0
Current Hardware rev is E03
Modify Module Hardware Rev? [Y/(N)]y
Enter new Hardware Rev:e04
Are you sure? [Y/(N)]y

Updating Hardware rev

Function? modify kfmsa0 2

XPC Interrupt Vector:          Port 1      Port 2
XPC Control Register Mask:    0400      0400
XPC Interrupt Priority Level:  0000      0000
Real-time Clock Timeout Period: 06         06
DSSI Initiator Timeout Period: 9D         9D
DSSI Target Timeout Period:   2C         2C
DSSI Selection Timeout Period: 27         27
DSSI Retry Initial Seed:      15         15
Number of Immediate DSSI Retries: 0008      0008
Number of Delayed DSSI Retries: 0100      0100
Maximum number of Coin-flips:  000A      000A
Idle Counter for Immediate Retries: 0000      0000
DSSI Retry Initial Seed:      A7524B79  A7524B79
XMI Transaction Timeout Value: 0003D810  0003D810
XMI Lockout Assertion Value:   000003E8  000003E8
XMI Lockout Deassertion Value: 00007A12  00007A12
CP Bus Timeout Value:          0000E66D  0000E66D
Secondary Lock Retries:        0020      0020

Do you wish to modify any of these parameters? [Y/(N)]y
Which Port (1=port 1, 2=port 2, 3=both) ? [1-3(1)]1 3
Modify CP Bus parameters? [Y/(N)]y 4
XPC Interrupt Vector? [(0400)]
Enter new value (HEX) or <CR> to keep present value:
XPC Control Register Mask? [(0000)]
Enter new value (HEX) or <CR> to keep present value:
XPC Interrupt Priority Level? [(06)]
Enter new value (HEX) or <CR> to keep present value:
Real-time Clock Timeout Period? [(9D)]
Enter new value (HEX) or <CR> to keep present value:
CP Bus Timeout Value? [(0000E66D)]
```

Enter new value (HEX) or <CR> to keep present value:
Secondary Lock Retries? [(0020)]

Enter new value (HEX) or <CR> to keep present value:

Modify DSSI Timeouts? [Y/(N)]n **4**

Modify DSSI Retries? [Y/(N)]n **4**

Modify XMI Timeouts? [Y/(N)]n **4**

Finished display/modify parameters? [(Y)/N]y

Function? m demna0 **5**

demna0

Remote Boot:	ENABLED
Remote Console:	ENABLED
Local Console:	ENABLED
Monitor Facility:	ENABLED
Promiscuous Mode:	ENABLED
Log Selftest Errors:	ENABLED
Log NI RBD Errors:	ENABLED
Log XMI RBD Errors:	ENABLED
Log XNA RBD Errors:	ENABLED
Diagnostic Error Logging:	DISABLED
Error Frame Overflow:	ENABLED

Console Password: XNABOARD

Module Serial Number: *SG909T1455*

Do you wish to modify any of these parameters?[y/(n)]y

Remote Boot:	ENABLED	Change? [y/(n)]y
Remote Console:	ENABLED	Change? [y/(n)]
Local Console:	ENABLED	Change? [y/(n)]
Monitor Facility:	ENABLED	Change? [y/(n)]
Promiscuous Mode:	ENABLED	Change? [y/(n)]
Log Selftest Error:	ENABLED	Change? [y/(n)]
Log NI RBD Errors:	ENABLED	Change? [y/(n)]
Log XMI RBD Errors:	ENABLED	Change? [y/(n)]
Log XNA RBD Errors:	ENABLED	Change? [y/(n)]
Diagnostic Error Logging:	DISABLED	Change? [y/(n)]
Error Frame Overflow:	ENABLED	Change? [y/(n)]

Modify demna0 with these parameter values?[y/(n)]y

Remote Boot:	DISABLED	6
Remote Console:	ENABLED	
Local Console:	ENABLED	
Monitor Facility:	ENABLED	
Promiscuous Mode:	ENABLED	
Log Selftest Errors:	ENABLED	
Log NI RBD Errors:	ENABLED	
Log XMI RBD Errors:	ENABLED	
Log XNA RBD Errors:	ENABLED	
Diagnostic Error Logging:	DISABLED	
Error Frame Overflow:	ENABLED	

Console Password: XNABOARD

Module Serial Number: *SG909T1455*

- ❶ The CIXCD has only one parameter: the hardware revision. You would need to modify the value only if the EEPROM had become corrupted.
- ❷ When you modify the KFMSA, LFU first displays all the parameters for both ports.
- ❸ You select which port to modify.
- ❹ LFU prompts for parameters by category.
- ❺ LFU displays the DEMNA parameters.
- ❻ This example modifies one parameter, disabling remote booting.

For more information:

KFMSA Module Installation and User Manual
CIXCD Interface User Guide
DEC LANcontroller 400 Technical Manual

Appendix A

Kermit Parameters

To transmit a file using Kermit, the following parameters must be set:

```
Kermit-32> show all
```

```
VMS Kermit-32 version 3.3.111
```

Block check type	One character checksum
Debugging	OFF
Delay	5 (sec)
Server sends NAKs every 75 seconds while waiting for a command	
Escape character	035 (octal)
File type	BINARY
File naming	Normal form
Handshaking character	None
Incomplete file disposition	Discard
Line used	(Optional)
Local echo	OFF
Parity type	None
Retry maximums	
Initial connection	5 (dec)
Sending a packet	5 (dec)
Send parameters	
Packet length	80 (dec)
Padding length	0 (dec)
Padding character	000 (octal)
Time out	5 (sec)
End of line character	015 (octal)
Quoting character	043 (octal)
Start of packet	001 (octal)
Receive parameters	
Packet length	80 (dec)
Padding length	0 (dec)
Padding character	000 (octal)
Time out	5 (sec)

End of line character	015 (octal)
Quoting character	043 (octal)
8-bit quoting character	046 (octal)
Start of packet	001 (octal)
Transmit parameters	
Delay	0.0 (sec)
Echo	OFF
Repeat quoting character	176 (octal)

Index

A

Attributes, setting device, 3- 18,
3- 19

B

Booting

DSSI VAXcluster, 3- 2

LFU, 3- 2, 3- 4

Build eeprom command, 2- 12

C

Configuration rules, LSB, 1- 4

Console CD- ROM

part number AG- PQW1*- RE, 3- 4

part number AG- PQW3*- RE, 3- 2

Console commands

build eeprom, 2- 12

create, 2- 6

set eeprom serial, 2- 6

update - e, 2- 8, 2- 10

update - f, 2- 8, 2- 10

D

Device mnemonic, LFU, 3- 6

Display command, LFU, 3- 14

E

Environment variables, 2- 2

Exit command, LFU, 3- 12

F

FEPROM recovery

hardware requirements, 2- 14

procedure, 2- 18

software requirements, 2- 16

FEPROM recovery code, 2- 15

Field- replaceable units, 1- 6

Firmware

corrupted, 3- 16

updating, 3- 1

Firmware revision of CPU, 2- 9

FRU part numbers, 1- 6

I

InfoServer 2- 15, 3- 1, 3- 4

Initialize command, 2- 13

Initial system load (ISL) file, 3- 5

K

Kermit program

example, 2- 18

parameters needed, A- 1

L

LFU, 3- 1

List command, LFU, 3- 8

LSB card cage, 1- 2

LSB configuration rules, 1- 4

M

Modify command, LFU, 3- 18

Module barcode label, 1- 7

Module installation procedure, 1- 10

Module removal procedure, 1- 8

N

Nickname, setting, 2- 5

R

Replace

- boot processor, 2- 8
- only processor, 2- 6
- secondary processor, 2- 10

RRD42, 2- 15, 3- 1, 3- 3

S

Self- test display, 1- 12

Set CPU command, 2- 9

Set eeprom serial command, 2- 6

Show command, LFU, 3- 6

Show device command, 2- 7

System parameters, how to set, 2- 2

U

Update command, 2- 9

Update command, LFU, 3- 10

V

Verification, 1- 12, 2- 2

Verify command, LFU, 3- 14