VAX 8200/8300
Owner's Manual

digital™

First Printing, December 1985
Revised, May 1986
Revised, September 1986

Printed in U.S.A.

A postage-paid READER'S COMMENTS form is included on the last page of this document. Your comments will assist us in preparing future documentation.

Contents

Tables

Preface

This manual, for owners of a VAX 8200 or 8300 system, explains how to operate and maintain your system after it has been installed by a DIGITAL Field Service representative. Descriptions of operation include turning on, booting, and restarting the system. Maintenance includes interpreting self-test results and updating the Electrically Erasable Programmable Read-Only Memory (EEPROM). This manual does not describe hardware in detail and does not describe diagnostics. The *VAX 8200/8300 Mini-Reference* provides brief descriptions of registers, address spaces, and interrupts and exceptions. For detailed descriptions of hardware and diagnostics, refer to the following manuals:

- *DWBUA UNIBUS Adapter Technical Manual* (EK–DWBUA–TM)
- *KA820 Processor Technical Manual* (EK–KA820–TM)
- *KDB50 Disk Controller User Guide* (EK–KDB50–UG)
- *MS820 MOS Memory Technical Manual* (EK–MS820–TM)
- *VAX 8200/8300 Installation Guide* (AZ–GN5AC–TE)
- *VAXBI Options Handbook* (EB–27271–46)

The file RELEASE_NOTES.DOC, contained on the VAX 8200/8300 Rel Notes diskette, lists any changes to the system that have occurred since the writing of the manuals.

Chapter Overviews

- **Chapter 1, The VAX 8200/8300 System,** outlines the basic components of the VAX 8200/8300 computer.

- **Chapter 2, Operating the VAX 8200/8300,** explains how to turn on the computer, boot the operating system, and run the VAX 8200/8300 self-test.
- **Chapter 3, The EEPROM Utility,** explains how to use the EEPROM Utility to change and examine system parameters.
- **Chapter 4, Identifying System Hardware,** explains how to run a program that lists system devices.
- **Chapter 5, Multiprocessing with the VAX 8300,** provides guidelines for operating multiprocessing.
- **Appendix A, EEPROM Parameters,** lists the system parameters you can change by using the EEPROM Utility.
- **Appendix B, Halt Codes and Error Codes,** lists the codes that can appear on the console terminal.
- **Appendix C, Setting Jumpers for Fast Self-Test,** outlines the steps for setting up the fast self-test.
- **Appendix D, Using the RX50 Dual-Diskette Drive,** explains how to use RX50 diskettes.
- **Appendix E, Deposit and Examine Console Commands,** describes in detail how to use both commands.
- **Appendix F, Control Flags for Booting,** lists the values of General Purpose Register R5 and the corresponding functions in the booting procedure.
- **Appendix G, VAX Disk Formatters,** explains how to reformat corrupted disks with programs contained on the Disk Formatters diskette.
- **Glossary** defines terms used to describe the VAX 8200 and 8300 systems.

Manual Conventions

Several conventions are followed in this manual.

- Console terminal displays are printed in the following typeface:

```
Do you want help?
```

- Your commands to the system or responses to questions are printed in red:

```
Do you want help? YES
```

- Variables are printed in lowercase type, as in the device type code ddnu.
- The symbol CTRL/P represents your terminal's CTRL key and P key pressed simultaneously. The symbol is also used with other letters, such as CTRL/Q and CTRL/S.
- The symbol RET represents your terminal's RETURN key. The symbol is not shown in all command examples. Unless told otherwise, assume that you must press the RETURN key after typing a command.
- The symbols ESC and BREAK, representing your terminal's Escape and Break keys, respectively, also appear in the manual.

The VAX 8200 and 8300 Systems 1

The VAX 8200 is a general-purpose computer system, while the VAX 8300 is designed for compute-intensive applications. VAX 8200 and 8300 systems use the same processor and control panel. Like other VAX systems, both the 8200 and the 8300 can support many users in a time-sharing environment, as well as numerous peripheral devices. Also, both systems:

- Execute the full VAX instruction set
- Use as their system bus the 12-slot or 24-slot VAXBI bus, a high-bandwidth bus designed for multiprocessing
- Allow for memory expansion
- Provide a UNIBUS adapter (optional) to connect UNIBUS peripheral devices
- Have an extensive self-test that runs automatically

In addition to those features, the VAX 8300 uses a second processor. For compute-intensive applications, the VAX 8300 provides up to 1.9 times the computing performance of the VAX 8200.

This chapter describes the VAX 8200/8300 system, shows its basic components, and outlines the contents of the operations kit.

1.1 VAX 8200/8300 COMPONENTS

Once a DIGITAL Field Service representative has installed your VAX 8200/8300 and verified that it runs properly, you are ready to operate the system. Before you turn on the system, examine the following diagrams to become familiar with important components, switches, and lights. Then continue with Chapter 2 for operating procedures.

Figures 1–1 and 1–2 show the basic 12-slot and 24-slot systems, which typically consist of the following:

- The main cabinet, which contains the control panel switches, the dual-diskette drive, and the VAXBI bus
- A cabinet that contains a tape drive and a disk drive
- Two switch keys for the control panel
- A hard-copy terminal (console terminal)
- An optional cabinet that contains the UNIBUS
- The VAX 8200/8300 operations kit, which includes six diskettes and four manuals

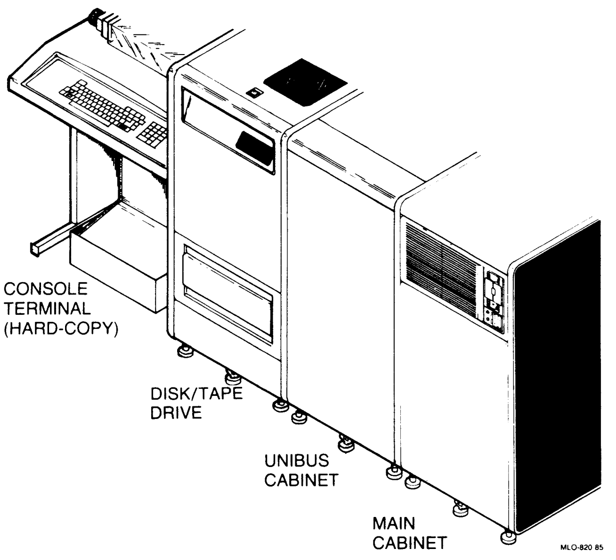


Figure 1–1: Basic VAX 8200/8300 System with 12-Slot Main Cabinet

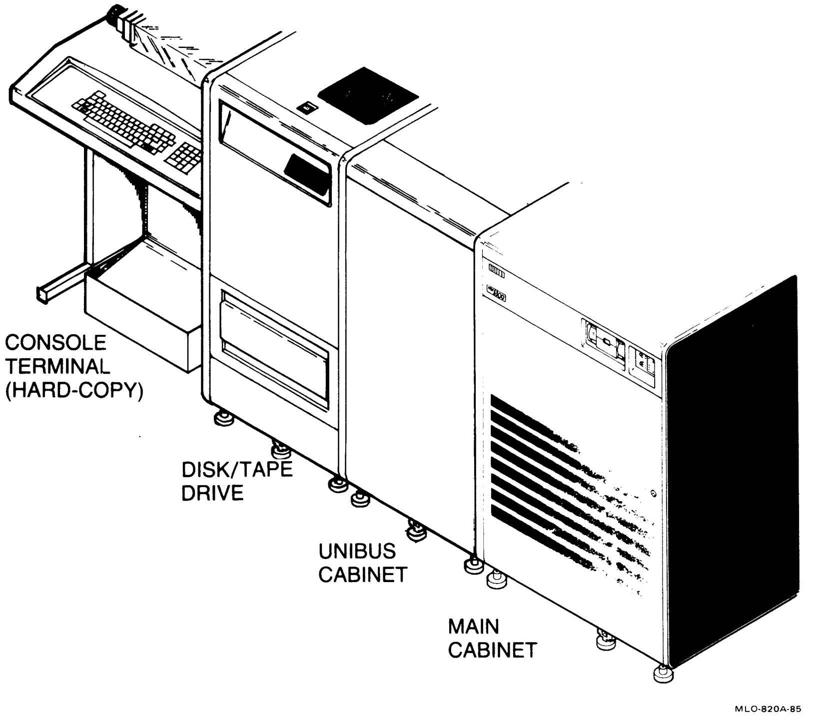


Figure 1–2: Basic VAX 8200/8300 System with 24-Slot Main Cabinet

This manual discusses the main cabinet and the operations kit. For information on terminals, disk drives, tape drives, and other optional devices, see their user guides.

1.2 MAIN CABINET

The 12-slot and 24-slot main cabinets, shown in Figures 1–3 and 1–4, include the following:

- Control panel
- Dual-diskette drive
- Input/output controllers
- Primary processor (and attached processor) and memory

- Adapters for other buses
- Power supply and cooling blower
- Optional battery backup unit

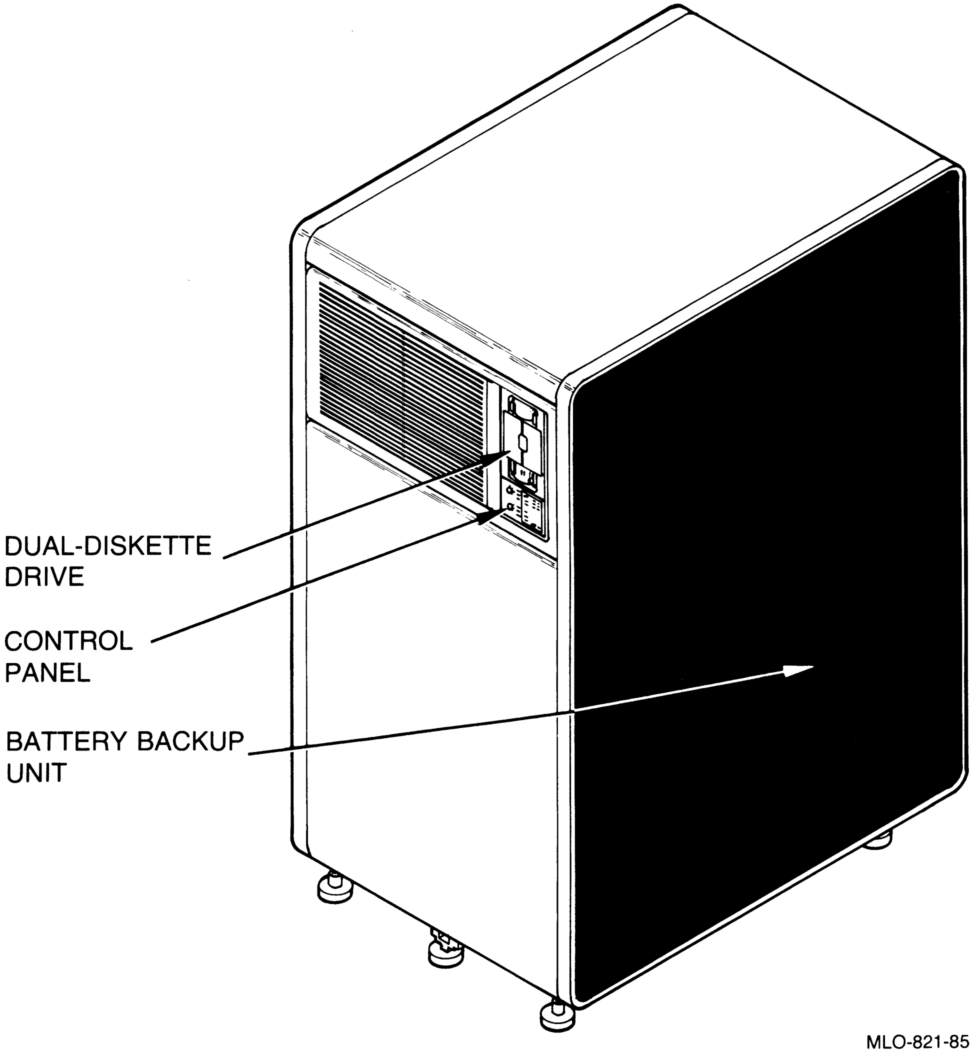


Figure 1–3: 12-Slot Main Cabinet

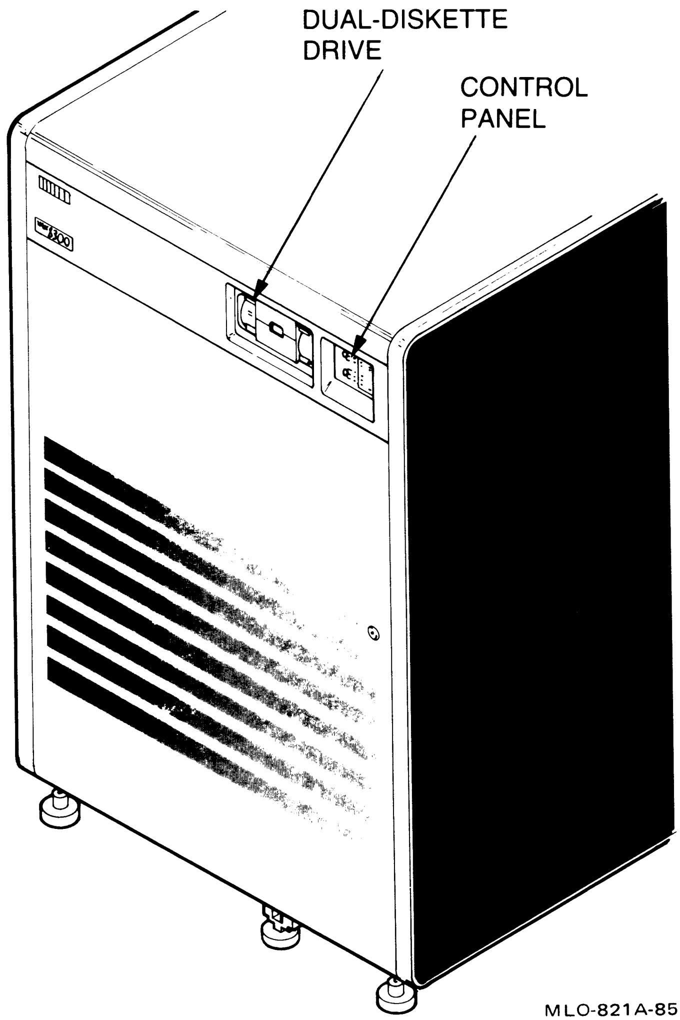


Figure 1–4: 24-Slot Main Cabinet

1.3 CONTROL PANEL

The control panel, shown in Figure 1–5, contains the following switches and indicators:

- Upper key switch: Off, Standby, Enable, Secure
- Lower key switch: Update, Halt, Auto Start
- Status indicators: Run, Battery, Fault
- Restart button

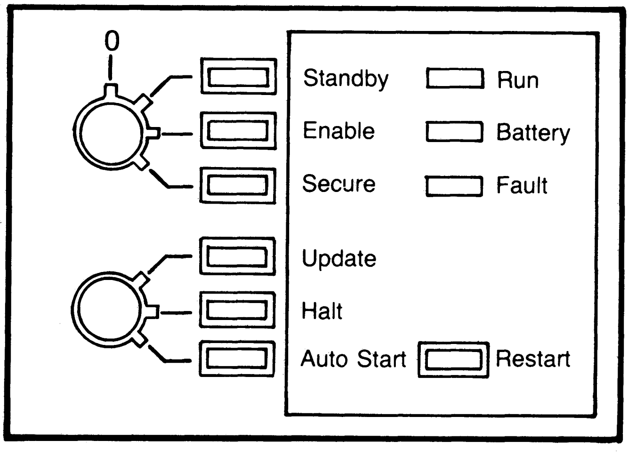


Figure 1–5: Control Panel with English Labels

Some VAX 8200/8300 control panels are labeled with symbols rather than English words. Figure 1–6 shows the correspondence.

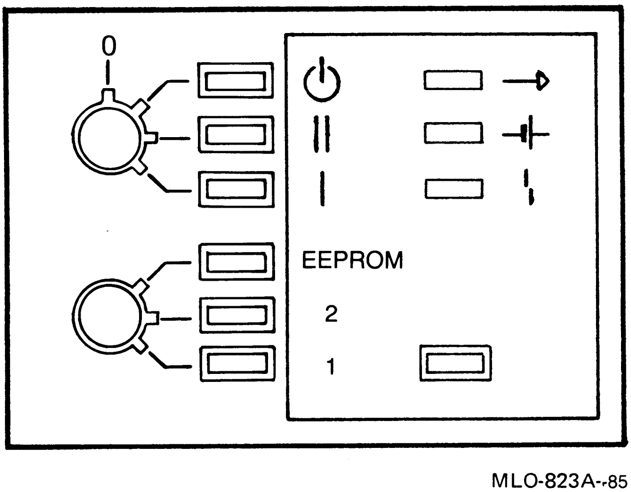
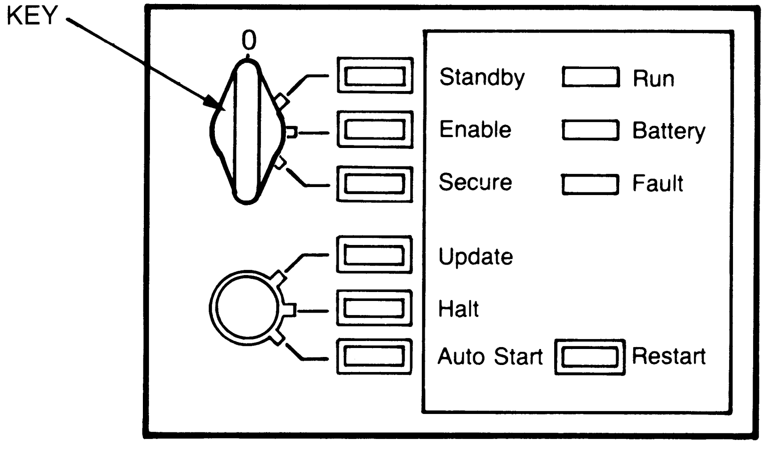


Figure 1–6: Control Panel with Symbols

Upper Key Switch

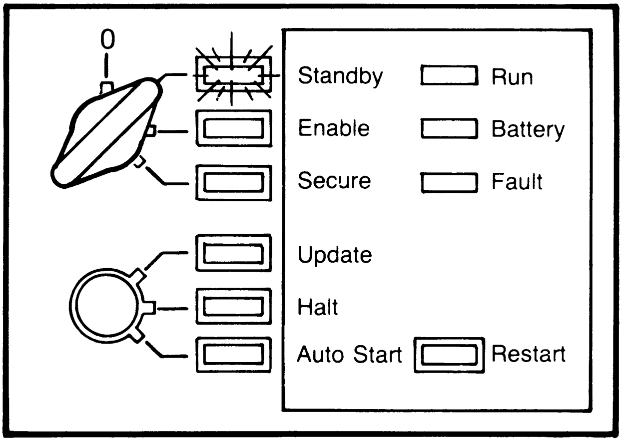
The upper key switch has four positions: Off, Standby, Enable, and Secure. You can change a position by inserting and turning a key.

- **Off**—Moving the switch to the Off position removes power from the entire system, including the battery backup unit.



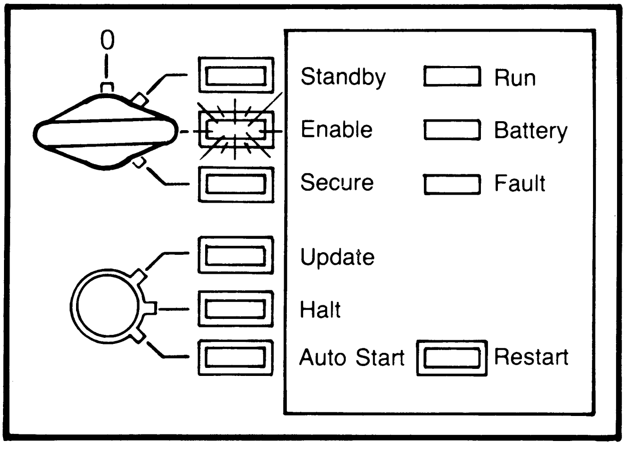
MLO-824A-.85

- **Standby**—Moving the switch to the Standby position turns on the power supply, the blower in the processor drawer, and system memory. The red light next to Standby is on. Standby is used for system maintenance.



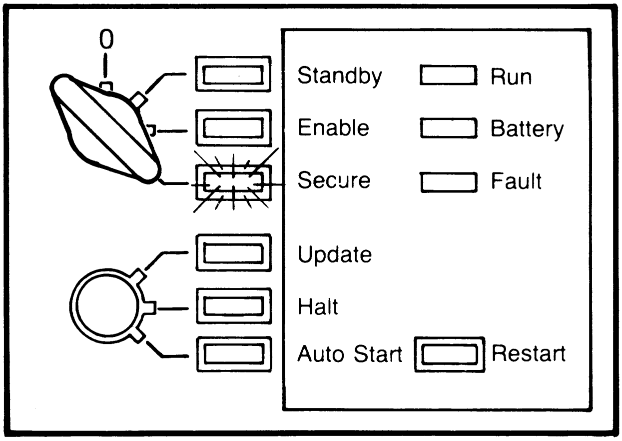
MLO-825A-85

- **Enable**—Moving the switch to the Enable position supplies power to the entire computer system; the yellow light next to Enable is on. You can enter console mode (see Section 2.7) or use the Restart button (see Section 2.3.1) only when the upper switch is in the Enable position. When you move the switch from Standby to Enable, the system tests itself.



MLO-826A-85

- **Secure**—Moving the switch to the Secure position maintains power to the entire system. The Secure position is for normal operation; the green light next to Secure is on. You cannot enter console mode or use the Restart button when the switch is in the Secure position.

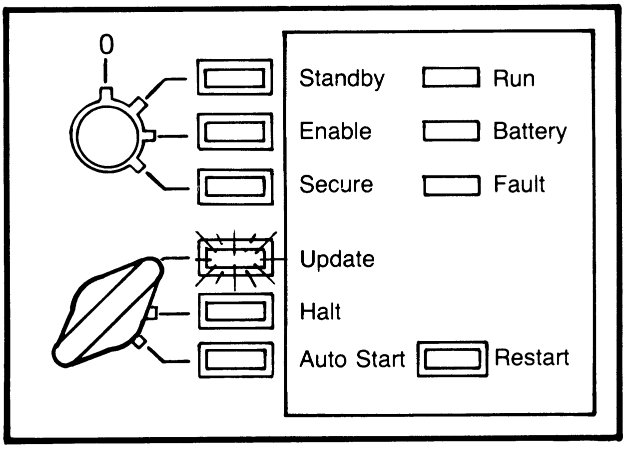


MLO-827A-85

Lower Key Switch

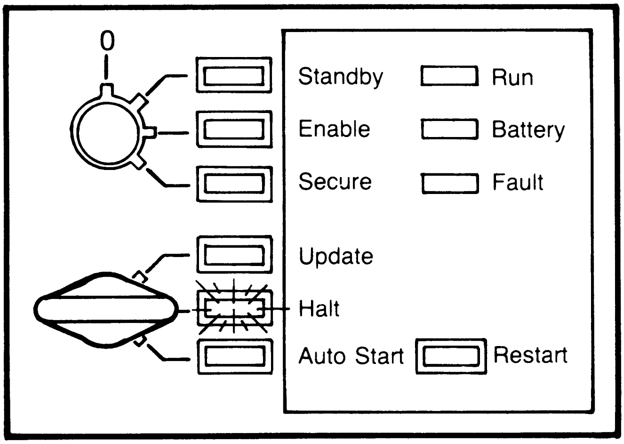
The lower key switch has three positions: Update, Halt, and Auto Start. You can change a position only by inserting and turning a key.

- **Update**—When the switch is in the Update position, you can update EEPROM data of the primary processor by using the EEPROM Utility or console commands. The switch does not need to be in the Update position when you update EEPROM data of an attached processor. The red light next to Update is on. This position prevents an automatic restart after a power failure (the processor halts in console mode).



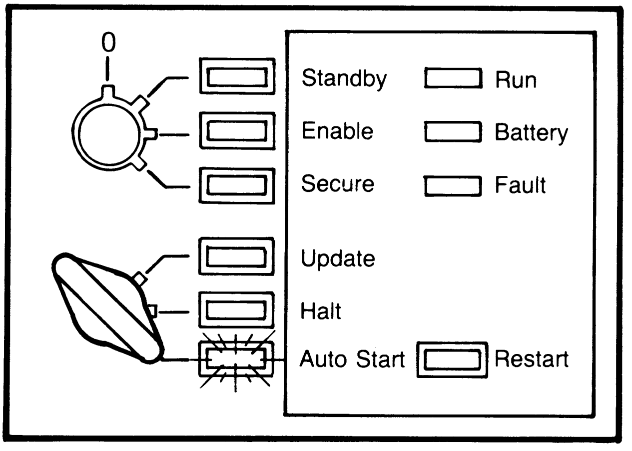
MLO-828A-85

- **Halt**—The Halt position prevents an automatic restart after a power failure (the processor halts in console mode). The yellow light next to Halt is on.



MLO-829A-85

- **Auto Start**—The switch should be in the Auto Start position for normal operation; the green light next to Auto Start is on. This position lets the processor automatically restart (warm start) or reboot (cold start) the operating system after an operating system failure or a power failure. A reboot occurs only if all VAXBI nodes pass self-test. See Sections 2.3.4 and 2.3.5.

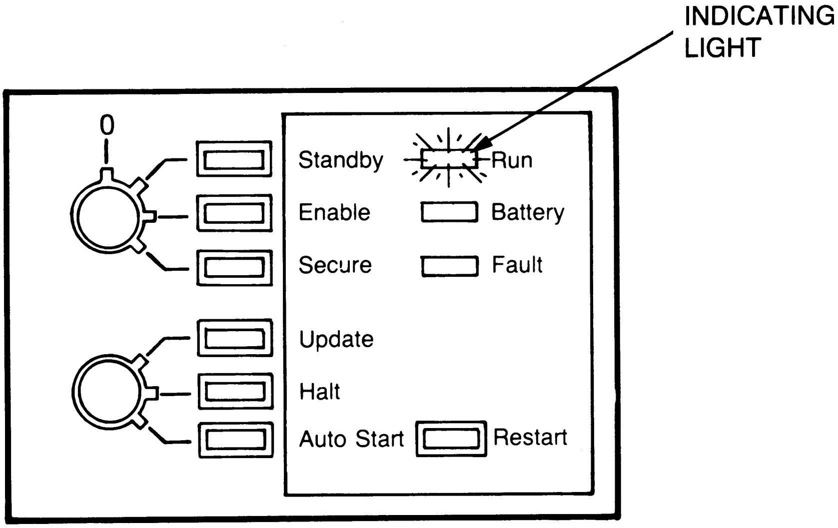


MLO-830A-85

Status Indicators

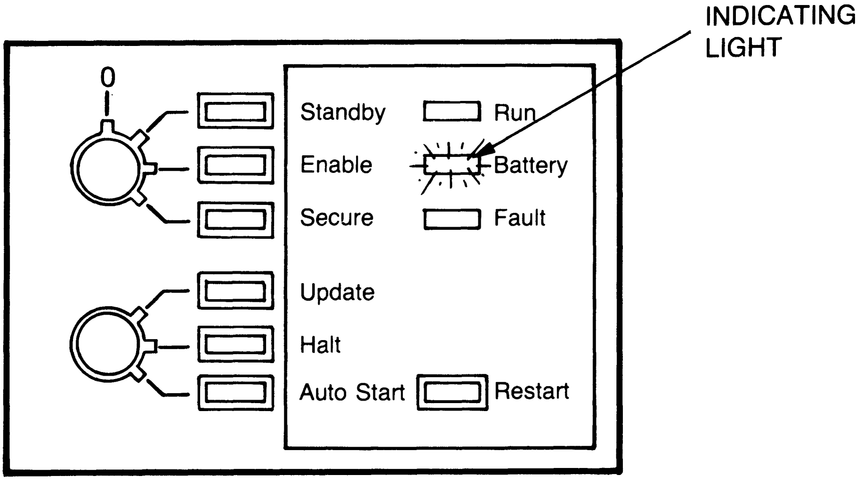
The system has three status indicators: Run, Battery, and Fault.

- **Run**—The Run light (green) is on when the VAX 8200/8300 is executing VAX instructions. When the system enters console mode, the Run light turns off.



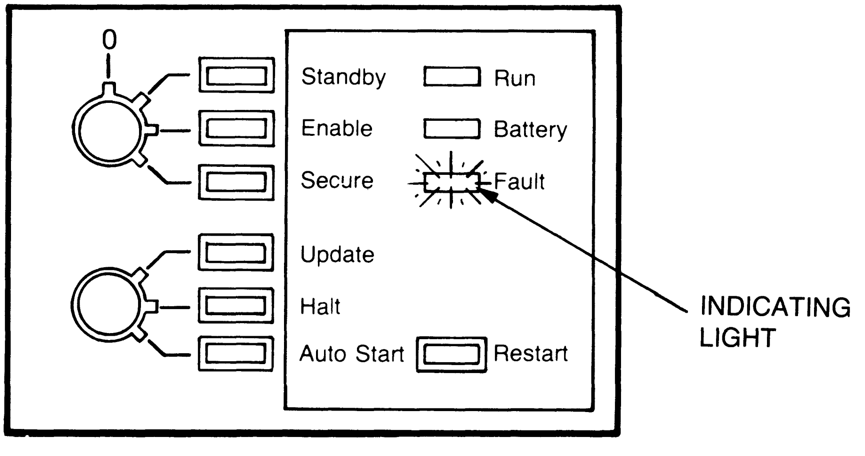
MLO-831A-85

- **Battery**—The Battery light (green) is on when the battery backup unit is fully charged. When flashing once a second, the light indicates that the unit is charging itself. When flashing 10 times a second, the light indicates that the unit is supplying power to the system. If your system does not have battery backup, this light is always off.



MLO-832A-85

- **Fault**—The Fault light (red) is on during self-test and turns off once self-test is successful. If the light remains on, the system has a hardware fault. See Section 2.8 for information on self-test.



MLO-833A-85

Restart Button

When you press the Restart button, the system runs self-test and the processor reboots the operating system (cold start). For the processor to reboot the operating system, the upper key switch must be set to Enable and the lower key switch must be set to Auto Start. If the system fails self-test, the processor does not reboot the operating system.

If the lower switch is in any other position when the upper switch is set to Enable, pressing the Restart button runs only self-test and the processor halts in console mode. When the upper key switch is not set to Enable, pressing the Restart button has no effect.

Table 1–1 summarizes the functions of the Restart button.

Table 1–1: Restart Button

Upper Key Switch	Lower Key Switch	Function
Enable	Auto Start	Runs self-test and reboots the operating system
Enable	Update or Halt	Runs self-test
Standby or Secure	Any position	Does not function

1.4 DUAL-DISKETTE DRIVE

You can use RX50 diskettes, such as the EEPROM Utility diskette, in the dual-diskette drive. Appendix D shows you how to use diskettes.

1.5 BATTERY BACKUP UNIT

The optional battery backup unit supplies power to system memory and to the cooling blower for several minutes during a power failure. With a battery backup unit, the computer can restart the operating system at the point where power failed, if power returns shortly (see Section 2.3.4). The Battery light on the control panel indicates the status of the unit. Without battery backup, the computer reboots the operating system after a power failure.

1.6 COMPONENTS IN THE MAIN CABINET

The back of the 12-slot and 24-slot main cabinets (shown in Figures 1–7 and 1–8 with the panel removed) have the following components:

- Four terminal connectors
- Main power switch
- Power light
- Power bus switch
- Interface options for peripheral devices

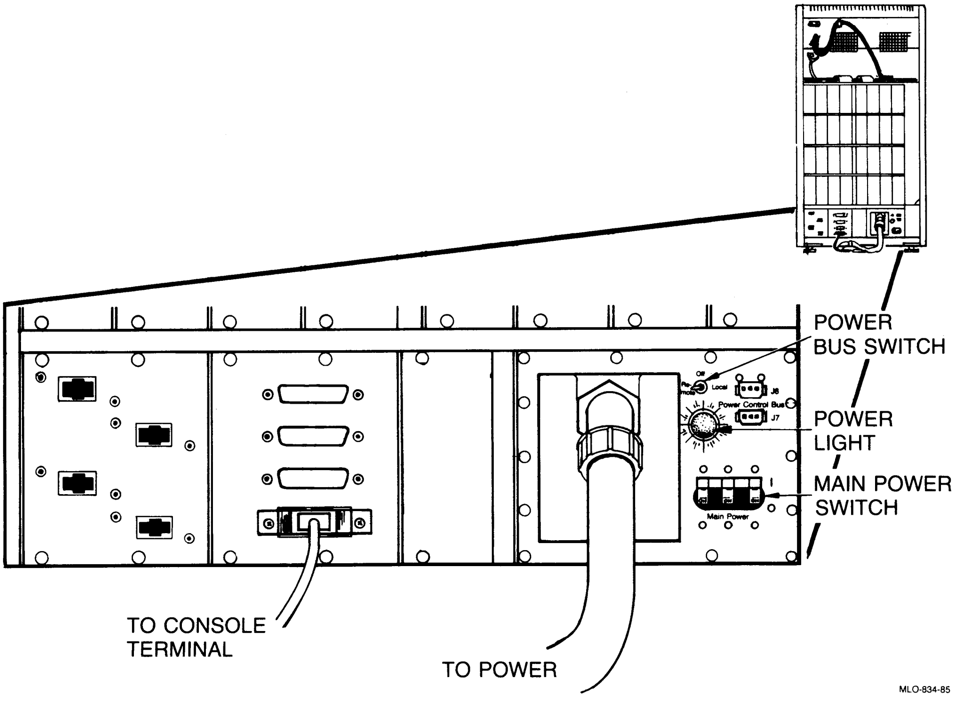


Figure 1–7: Back View of 12-Slot Main Cabinet with Panel Removed

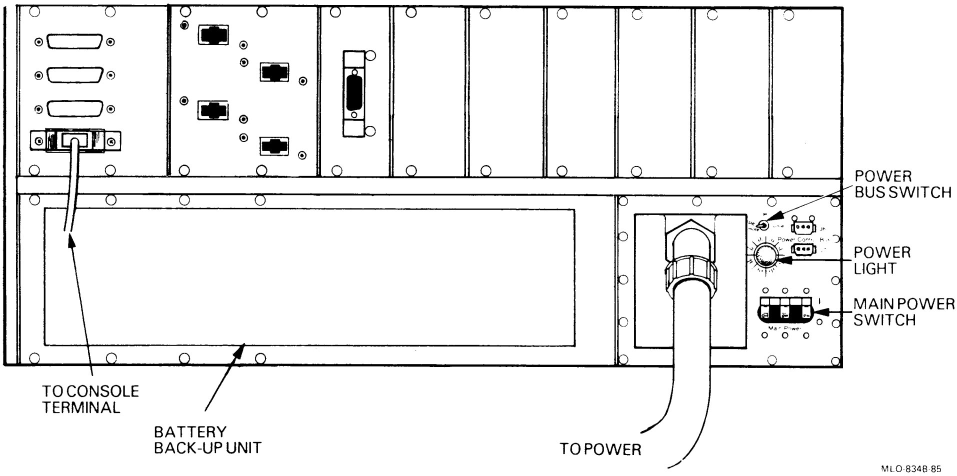


Figure 1–8: Back View of 24-Slot Main Cabinet with Panel Removed

The main power switch should be in the On position and the orange light should be lit, indicating that the main cabinet is plugged into a live electrical outlet. The power bus switch should be in the Remote position.

Terminal connector 0 is connected to the console terminal.

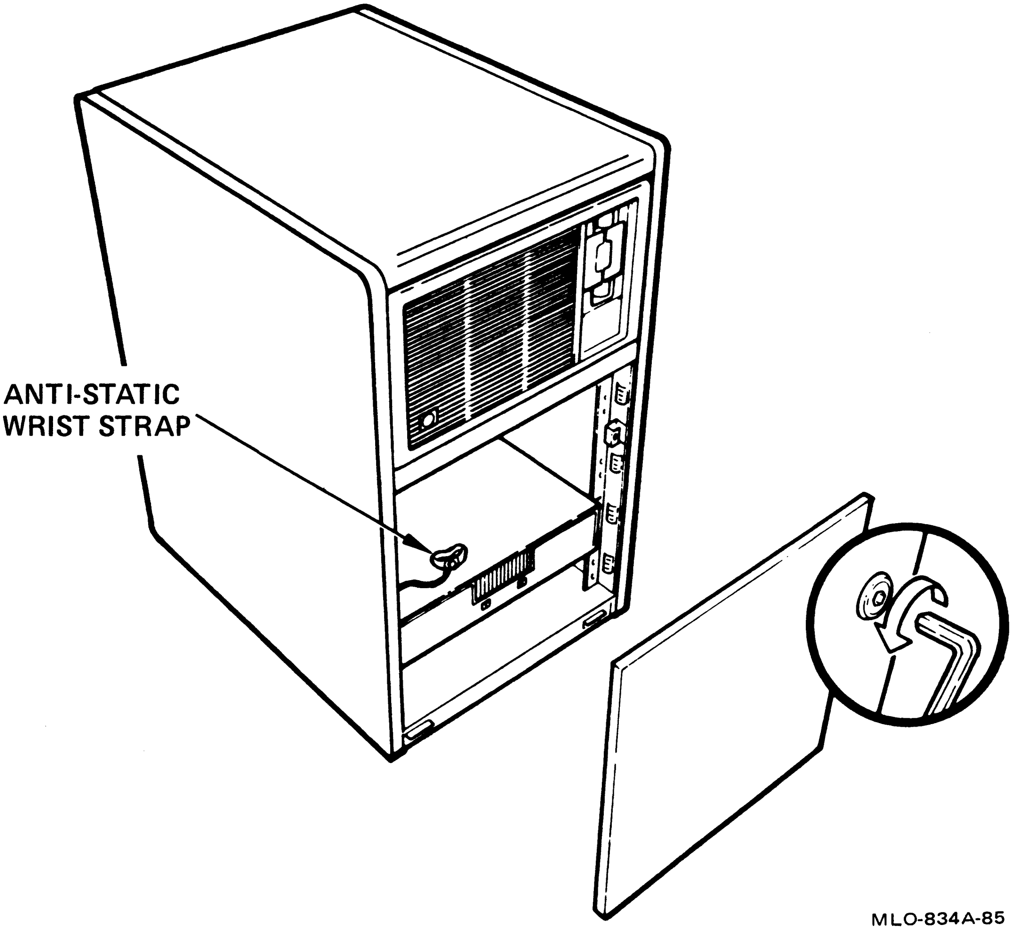
1.7 VAXBI BUS

The VAXBI bus contains the primary processor (and the attached processor), memory, input/output controllers, and adapters for other buses. VAX 8200/8300 systems use a 12-slot or 24-slot VAXBI bus. The 12-slot and 24-slot VAXBI buses are located in the main cabinet.

1.7.1 12-Slot VAXBI Bus

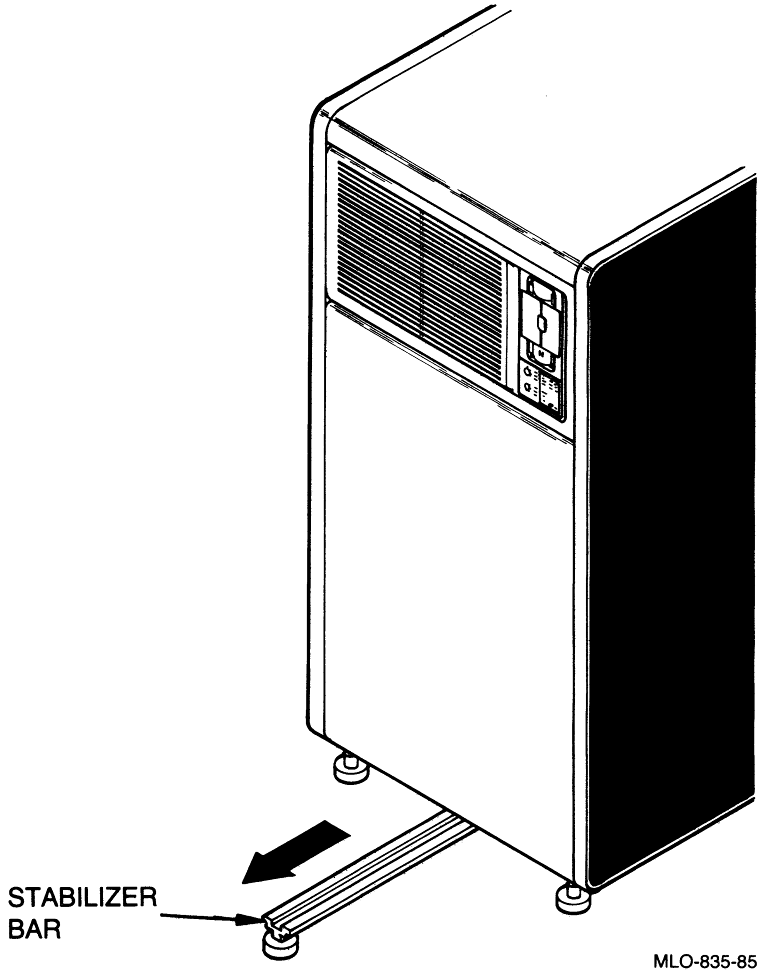
The 12-slot VAXBI bus is located in the processor drawer. You can push the drawer out from the main cabinet by following these steps:

1. Turn the upper key switch to the Off position.
2. Open the front panel on the main cabinet with a 5/32-inch hex key.

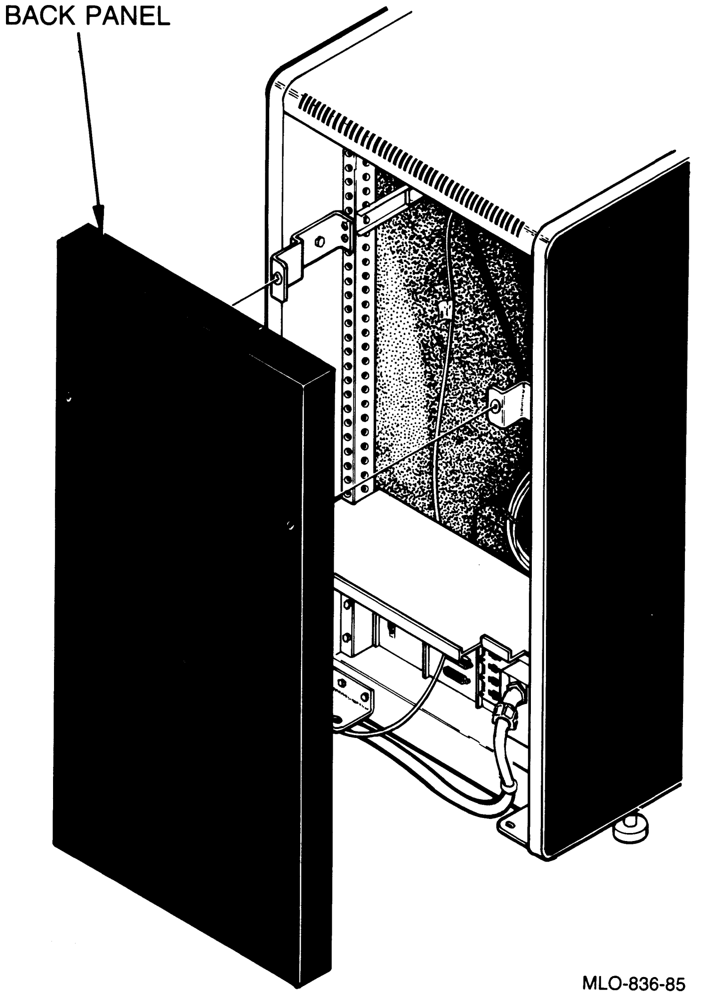


3. Wear the anti-static wrist strap, located at the base of the main cabinet. You must wear the wrist strap before opening the processor drawer to avoid damaging static-sensitive hardware.

4. Pull out the stabilizer bar at the bottom of the main cabinet.

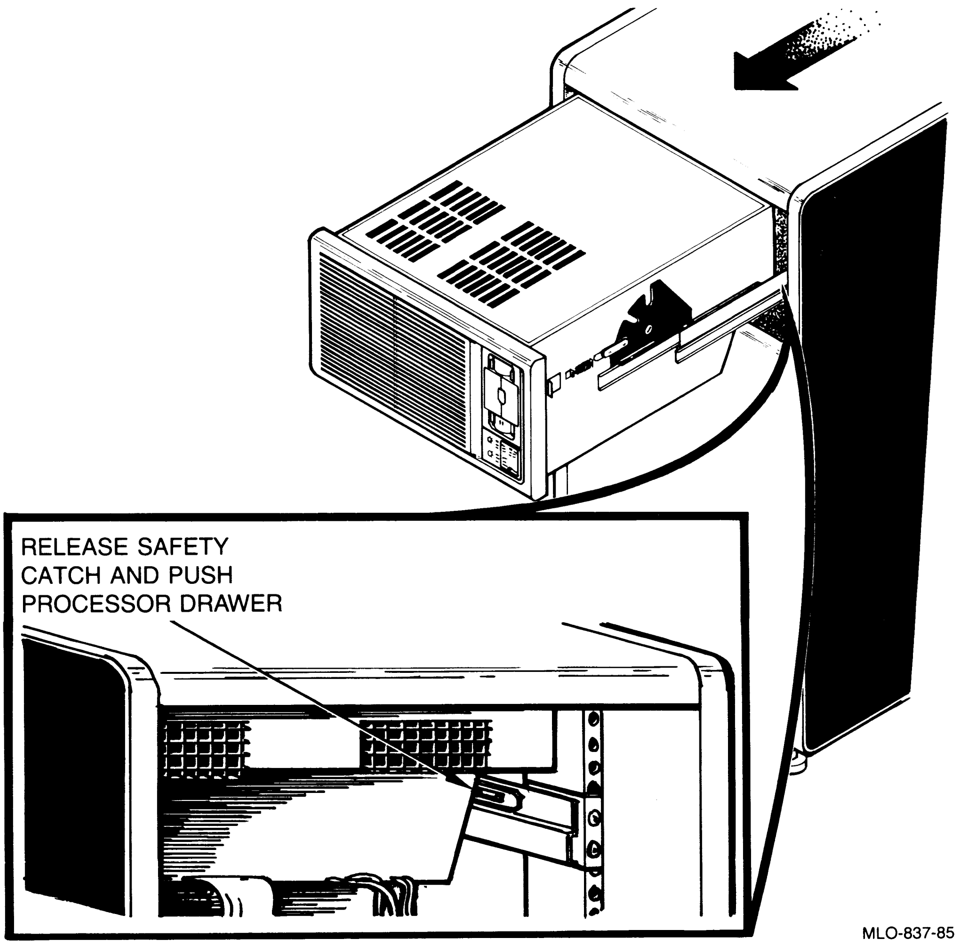


5. Remove the back panel with a 5/32-inch hex key.



MLO-836-85

6. Push the safety catch.

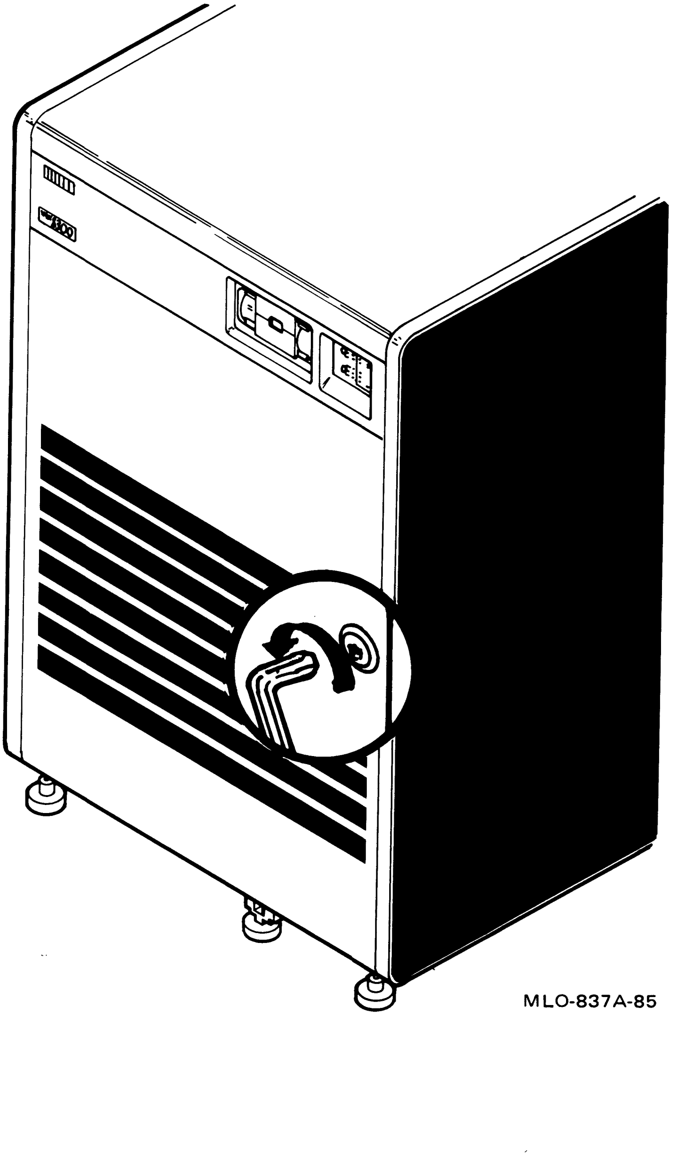


7. Carefully push out the processor drawer.

1.7.2 24-Slot VAXBI Bus

The 24-slot VAXBI bus is located in the bottom of the main cabinet. You can open the main cabinet by following these steps:

1. Turn the upper key switch to the Off position.
2. Open the front panel on the main cabinet with a 5/32-inch hex key.



MLO-837A-85

1.7.3 Power Lights

Power lights indicate the status of the power supply modules; two modules supply power to each 12-slot VAXBI section. The 12-slot VAXBI bus contains the power lights beneath the rightmost window, shown in Figure 1–9. The 24-slot VAXBI bus contains two sets of power lights, one for each 12-slot section; the power lights are shown in Figure 1–10.

Lights in other slots are self-test indicators, described in Chapter 2.

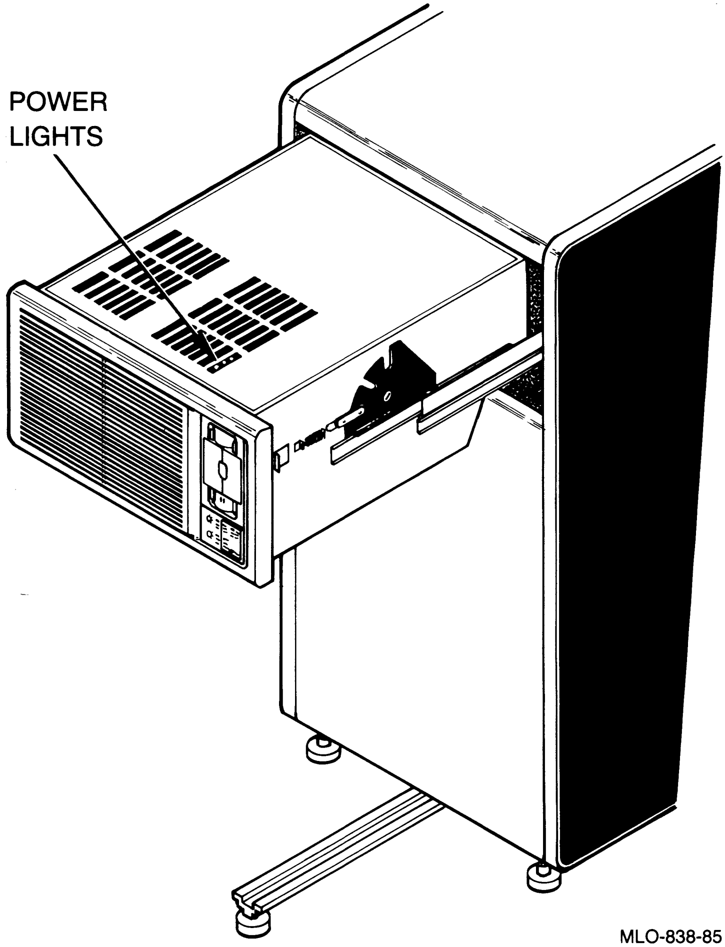


Figure 1–9: Power Lights for a 12-Slot VAXBI Bus

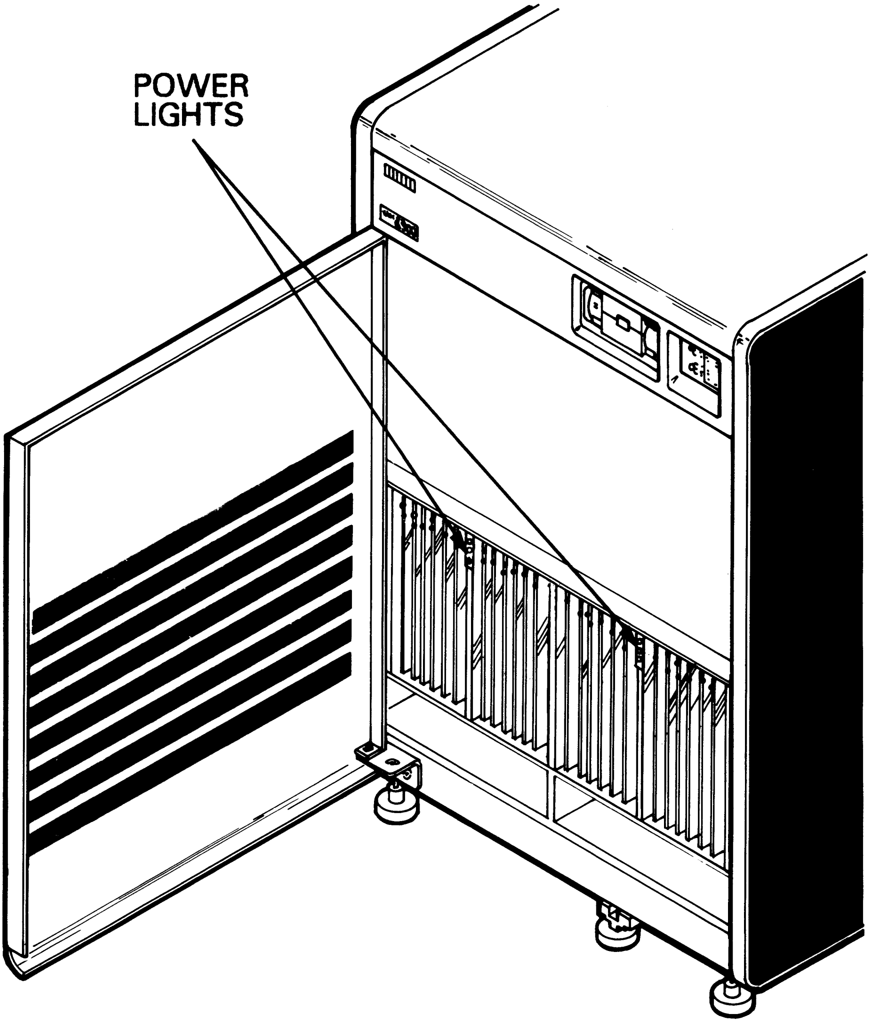


Figure 1–10: Power Lights for a 24-Slot VAXBI Bus

One set of power lights consists of one green and two red indicators. When the power cord is plugged into an electrical outlet and the upper key switch is in the Off position all power lights should be off. When the system is turned on, only green lights should be lit.

If one or more of the red lights are lit when the system is turned on, at least one of the power supply modules is faulty. After you turn on the system, the green lights, when lit, indicate that the power supply modules have AC power.

1.8 POWER SUPPLY CIRCUIT BREAKER FOR THE VAXBI BUS

The power supply circuit breaker switch, which controls power to the VAXBI bus, should be in the On position. If the temperature of the VAXBI bus exceeds 50° C or if the air flow to the processor drawer (12-slot system) or to the front of the main cabinet (24-slot system) is blocked, the switch automatically moves to the Off position so that power to the VAXBI bus is turned off. The 24-slot cabinet has two circuit breaker switches, one for each half of the VAXBI bus. Figures 1–11 and 1–12 show the circuit breaker switches for the 12-slot and 24-slot systems, respectively.

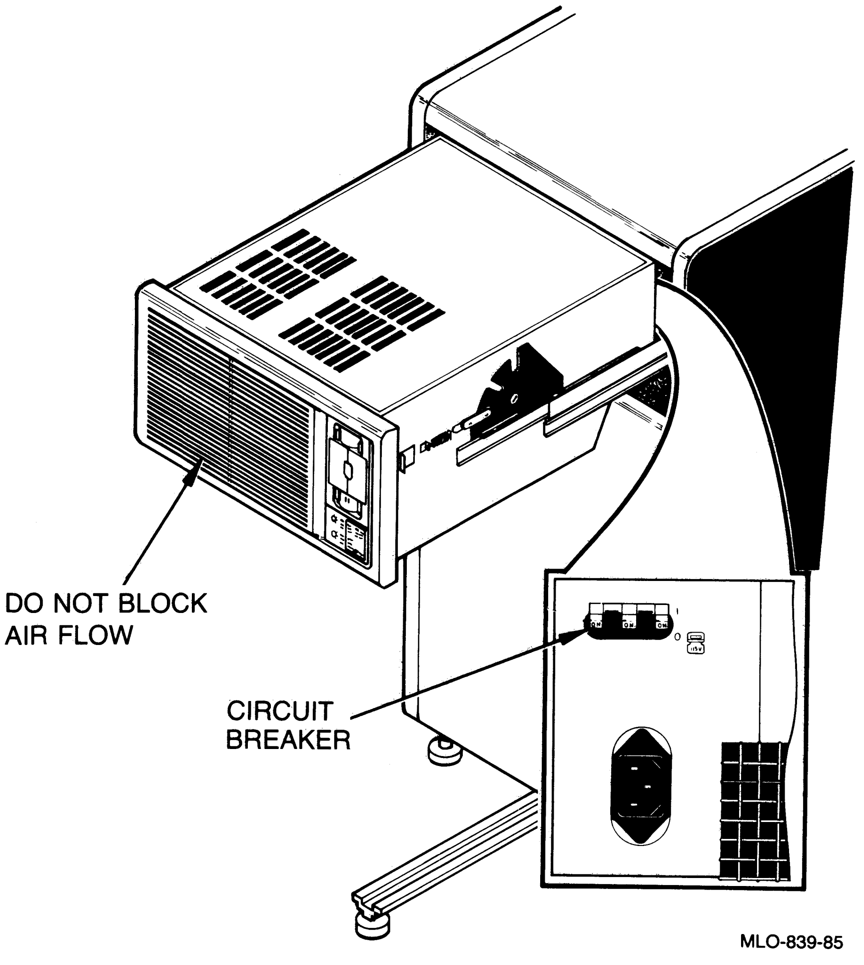


Figure 1–11: Circuit Breaker Switch for 12-Slot System

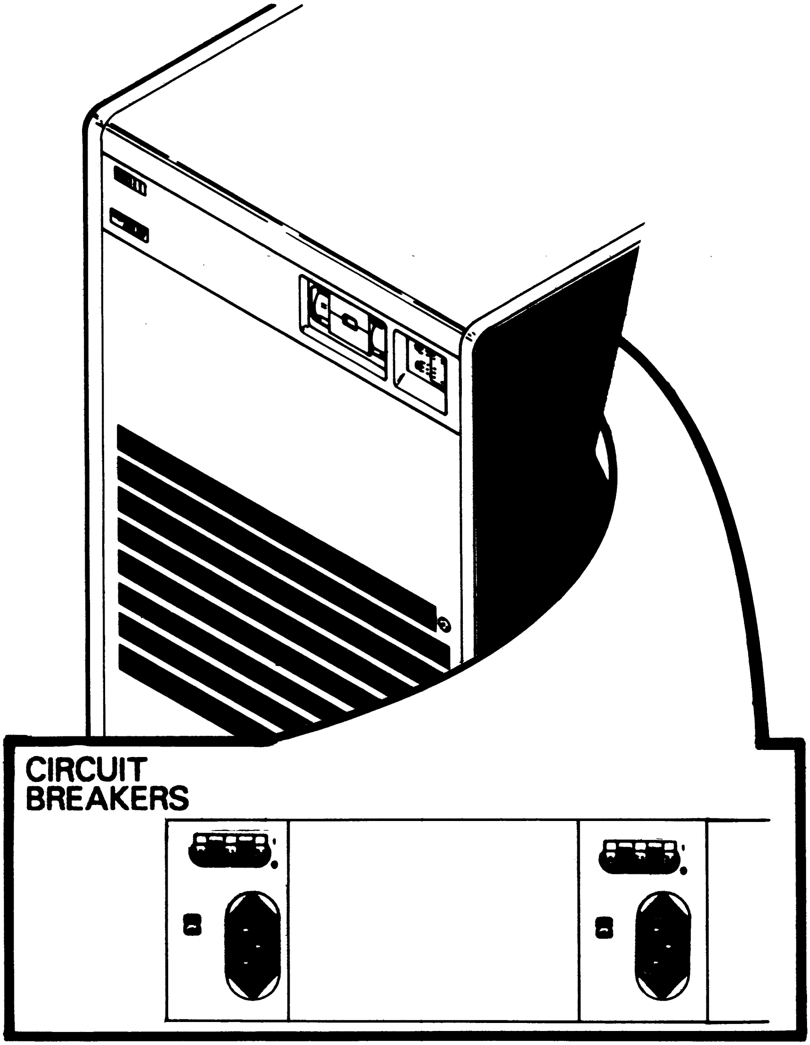


Figure 1–12: Circuit Breaker Switches for 24-Slot System

1.9 VAX 8200/8300 OPERATIONS KIT

The operations kit contains information to help you maintain and customize the VAX 8200/8300 system. The operations kit consists of six diskettes, a license to use them, four manuals, and a software product description. The diskettes are:

- **VAX 8200/8300 Console Flp (BL–FG81B–ME),** command files for booting VMS from a shared system disk on a VAXcluster and microcode for the CI adapter. See Chapter 2.
- **VAX 8200/8300 Util Prog Flp (BL–FG80B–ME),** the EEPROM Utility (and associated software) that lets you customize your VAX 8200/8300 system by modifying data stored in the EEPROM. See Chapter 3.
- **VAX 8200/8300 Diag Super + Auto (BL–FG79B–ME),** a program that lists system hardware. See Chapter 4.
- **VAX Disk Formatters (BL–FG84B–ME),** programs that let you reformat corrupted disks. See Appendix G.
- **VAX 8200/8300 Rel Notes (BL–FI09A–ME),** an ASCII file (RELEASE-NOTES.DOC) that contains system release notes, listing any changes to the system that have occurred since the writing of the manuals. You can read this file under VMS by inserting the Rel Notes diskette into console drive 1, mounting the console, and typing the VMS command:

  ```
  $ TYPE CSA1:RELEASE_NOTES.DOC.
  ```

- One blank diskette (BL–N402A–BK)

The manuals in the operations kit are:

- *VAX 8200/8300 Installation Guide* (AZ–GN5AC–TE)
- *VAX 8200/8300 Owner's Manual* (AZ–GN4AC–TE)
- *VAX 8200/8300 Mini-Reference* (AZ–GN6AB–TE)
- *VAXBI Options Handbook* (EB–27271–46)

1.10 SYSTEM ENVIRONMENT

The temperature and humidity of your system's environment should be maintained by an environmental control system. The VAX 8200/8300 should not be located near heaters or in direct sunlight. An operating VAX 8200/8300 requires the following environmental conditions:

- When the RCX50 diskette drive is in use, the system is restricted to:
  - Temperature from 15 to 32 degrees C (59 to 90 degrees F)
  - Relative humidity from 20% to 80%
  - Altitude from 0 to 2.4 km (0 to 8000 ft)
- When the RCX50 diskette drive is not in use, the system is restricted to:
  - Temperature from 10 to 40 degrees C (50 to 104 degrees F)
  - Relative humidity from 10% to 90%
  - Altitude from 0 to 2.4 km (0 to 8000 feet)

Operating the VAX 8200/8300 2

The VAX 8200/8300 system follows an automatic 3-step start-up sequence: the computer tests its hardware, initializes itself, then boots the operating system. You can also perform each of these steps separately by using console commands and switches on the control panel.

This chapter describes boot devices, explains how to start the VAX 8200/8300, how to reboot the operating system, and how to run and interpret the system self-test. The chapter also lists console commands and examples.

2.1 OVERVIEW OF BOOTING

The operating system or other software can be booted in one of four ways:

- Typing the console B command, optionally followed by the device type of a specific boot device
- Turning the control panel switches to the Auto Start and Secure positions (when power is off)
- Pressing the Restart button (after power is on)
- Automatic booting following a power or operating system failure

When you use the B command, you can specify a particular boot device; otherwise, the operating system is booted from the default boot device, which is set when you receive your system (see Section 2.1.1).

The three other booting methods boot the operating system from the default boot device.

2.1.1 Boot Devices

A boot device, usually a disk drive, contains a copy of the operating system software. The VAX 8200/8300 can boot an operating system from two types of boot devices:

- Local boot devices
- Shared boot devices (for VAXclusters)

A local boot device is connected directly to a computer and is accessible by only that computer. Figure 2–1 shows the configuration of a local boot device on a system.

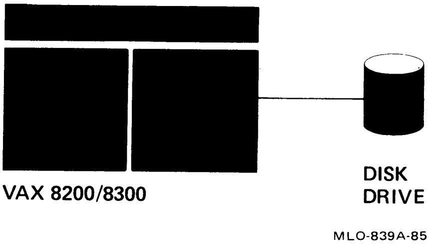


Figure 2–1: Local Boot Devices

A shared boot device, part of a VAXcluster, is connected to an HSC controller and is accessible by all computers that belong to the VAXcluster. Figure 2–2 shows the configuration of shared boot devices on a 2-node VAXcluster with one HSC controller. The first time you boot VMS you must use a special procedure, described in Section 2.4.

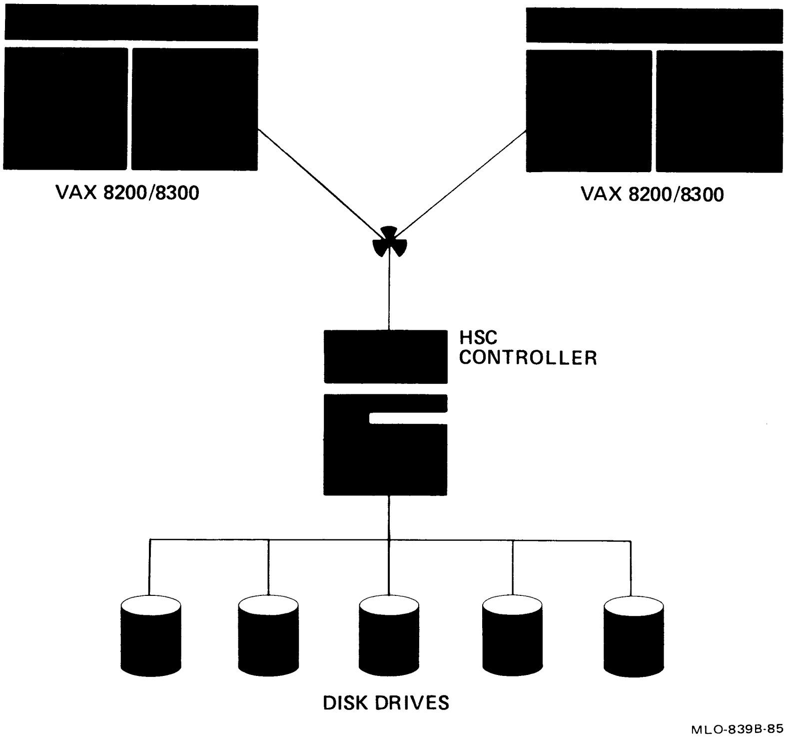


Figure 2–2: Shared Boot Devices

Your system uses a default boot device to boot the operating system when you do not specify a device or when the operating system is automatically booted. Information in the EEPROM (see Chapter 3) determines which boot device on your system is the default. The default boot device is set in the EEPROM when you receive your system. You can change the default by using the EEPROM Utility (see Chapter 3).

A system without an HSC controller uses a local boot device for its default. A system with an HSC controller uses the dual-diskette drive for its default; a file on a diskette in the dual-diskette drive directs the booting procedure to a particular boot device on the HSC controller. A system with an HSC controller can also use a local boot device for its default.

2.1.2 Booting Procedures

The booting procedures for local boot devices and shared boot devices differ in the sequence of steps. The methods of booting from each of the two devices, therefore, are different also.

From a Local Boot Device—The booting procedure for a local boot device consists of the following steps:

1. You type the B command, specifying a boot device. If you do not specify a device, the VAX 8200/8300 boots the operating system from the default boot device.
2. Boot code stored in the EEPROM directs the booting procedure to the boot device.
3. The boot block program stored in the boot device is loaded into system memory and, in turn, the boot block program loads the primary loader from the boot device into system memory. For VMS, the primary loader is VMB.EXE.
4. The primary loader loads the secondary loader, which, in turn, loads the operating system.

From a Shared Boot Device (VAXclusters)—The booting procedure for a shared boot device uses the console diskette (Console Flp diskette) and consists of the following steps:

1. You type the console B command, specifying the console diskette drive as the boot device. If the VAX 8200/8300 boots the operating system from the default boot device, the default boot device must be the console diskette drive. In both cases, the console diskette must be inserted in console drive 1.
2. Boot code stored in the EEPROM directs the booting procedure to the console diskette drive.

3. The boot block program stored on the console diskette is loaded into system memory and, in turn, the boot block program loads a program called BOOT58 from the diskette into system memory. BOOT58 lets you load files, execute command procedures, and deposit data in General Purpose Registers.
4. BOOT58 automatically loads and runs the command file DEFBOO.CMD. This command file, which you have customized for your system, loads the primary loader VMB.EXE from the diskette, then directs the booting procedure to a particular boot device connected to the HSC controller. The HSC boot device is specified in the command file DEFBOO.CMD.
5. VMB.EXE loads microcode from the console diskette into the CI adapter, then loads the secondary loader from the specified HSC boot device. The secondary loader loads VMS from the same device.

The first time you boot VMS from a shared boot device, the command file DEFBOO.CMD does not exist. You must, therefore, run the program BOOT58 interactively to enter data (Section 2.4.1). After you type the required data, the booting procedure continues as described. Once VMS is booted, you must customize the command file CIBOO.CMD for your system (Section 2.4.2), supplying the same data you typed in response to the BOOT58 prompts. You then rename the customized CIBOO.CMD to DEFBOO.CMD. When you boot VMS again from a shared boot device, BOOT58 automatically runs the command file DEFBOO.CMD.

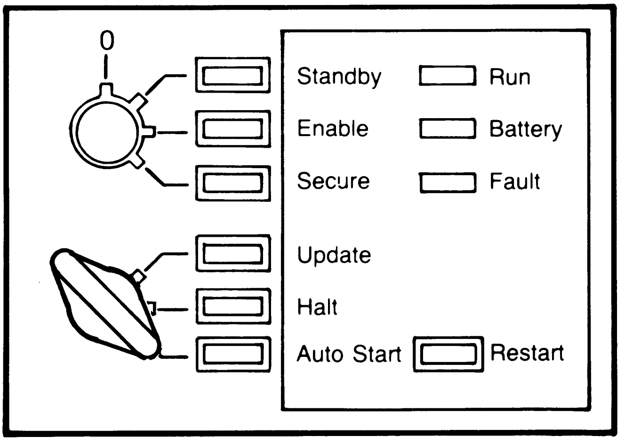
2.2 STARTING WHEN POWER IS OFF

The normal start, the easiest method for starting your VAX 8200/8300, boots the operating system from the default boot device (set in the EEPROM). You perform a normal start by turning the control panel switches so that all indicator lights are green; the upper switch is in the Secure position, the lower switch is in the Auto Start position. A normal start runs self-test and initializes the system, then boots the operating system. The first time you start your system, you can perform a normal start only with a local boot device. If your system belongs to a VAXcluster, you must use a special procedure for the first time (see Section 2.4). For subsequent starts, however, you can perform a normal start.

2.2.1 Normal Start from a Local Boot Device

To perform a normal start from a local boot device, follow these steps:

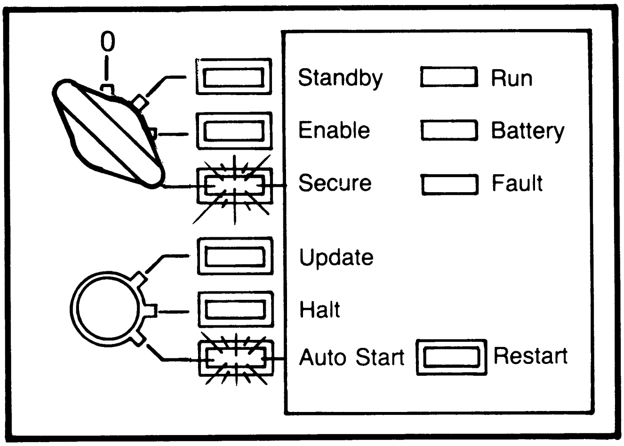
1. Insert a key into the lower control panel key switch. Turn the key clockwise to the Auto Start position.



MLO-840A-85

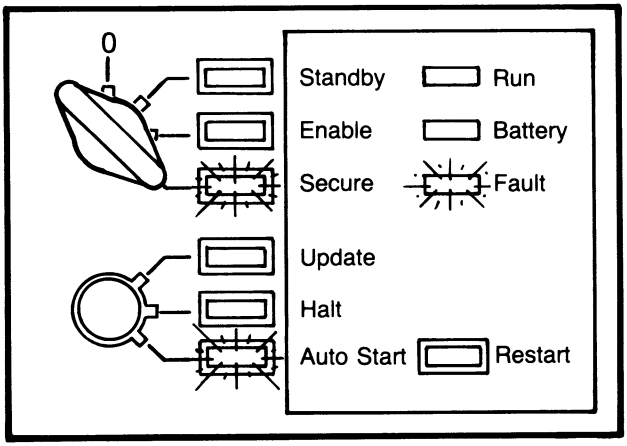
NOTE
If your VAX 8200/8300 has symbols instead of English words on the control panel, see Figures 1–5 and 1–6 for the correspondence.

2. Insert a key into the upper key switch. Turn the key clockwise to the Secure position so that the green light next to Secure is on. The green light next to Auto Start is also on.



MLO-841A-85

The VAX 8200/8300 tests its hardware. During self-test, the Fault light is on.



MLO-842A-85

After the processor passes self-test, the following display appears on the console terminal:

```
#ABCDEFGHIJK.MN#
```

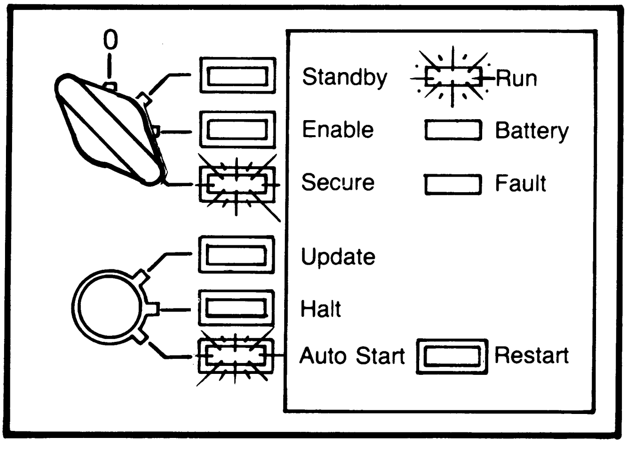
If the processor fails self-test, you see only a part of the string of letters. For example, you might see only #ABCD. See Section 2.8.3 for a complete description of the self-test display.

The system initializes itself: options on the system bus run their self-tests. If all the options pass self-test, the Fault light turns off, and you see a display like the following on the console terminal:

```
0 . 2 . 4 . . . 8 9 A B C D . .
```

The display should contain only hexadecimal digits and periods. If any option fails self-test, the Fault light remains lit, and at least one digit has a minus sign before it. See Section 2.8.3 for a complete description of this display.

The Run light turns on, and the processor boots the operating system.



MLO-843A-85

2.2.2 Normal Start from a Shared Boot Device (VAXclusters)

The first time you boot VMS from a shared boot device, you must use a special procedure, which includes the customization of the console diskette. Section 2.4 describes the first-time booting procedure and explains how to customize the console diskette. Once you customize the console diskette, you can perform a normal start from a shared boot device.

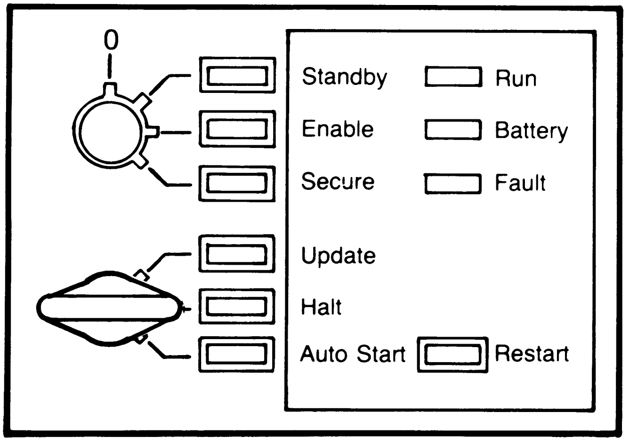
For a normal start from a shared boot device, the default boot device, which is set in the EEPROM, must be console drive 1 (CSA1). To perform a normal start, follow these steps:

1. Insert the customized console diskette (Console Flp diskette) into console drive 1 (CSA1). If you have not yet customized the diskette, see Section 2.4.
2. Follow the steps in Section 2.2.1, Normal Start from a Local Boot Device.

2.2.3 Booting from a Specified Local Boot Device

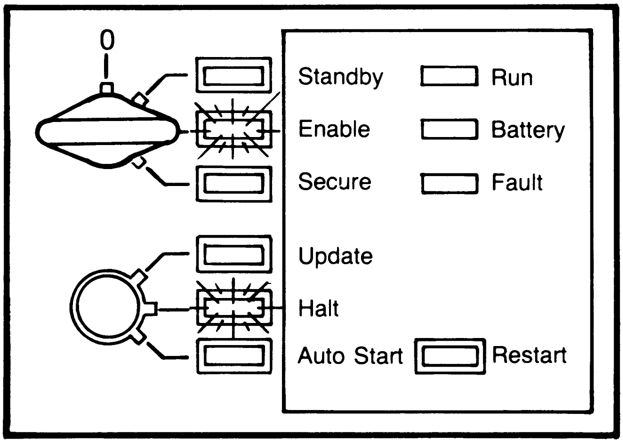
To turn on the VAX 8200/8300 and boot the operating system from a specific boot device, follow these steps. If your system is not part of a VAXcluster, skip step 1.

1. Insert the console diskette (Console Flp diskette) into console drive 1. (This step is necessary only for a system that is part of a VAXcluster.)
2. Insert a key into the lower control panel key switch. Turn the key clockwise to the Halt position.



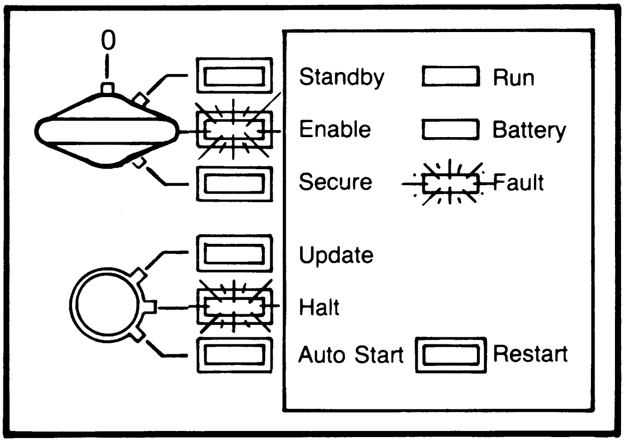
MLO-844A- 85

3. Insert a key into the upper key switch. Turn the key clockwise to the Enable position so that the yellow light next to Enable is on. The yellow light next to Halt is also on.



MLO-845A-85

The VAX 8200/8300 tests its hardware. During self-test, the Fault light is lit.



MLO-846A-85

If the processor passes self-test, you see the following display on the console terminal:

```
#ABCDEFGHIJK.MN#
```

If the processor fails self-test, you see only a part of the string of letters. For example, you might see only #ABCD. See Section 2.8.3 for a complete description of the self-test display.

The system initializes itself: options on the system bus run their self-tests. If all the options pass self-test, the Fault light turns off, and you see a display like the following on the console terminal:

```
0 . 2 . 4 . . . 8 9 A B C D . .
```

The display should contain only hexadecimal digits and periods. If an option fails self-test, the Fault light remains lit, and at least one digit has a minus sign before it. See Section 2.8.3 for a complete description of this display.

The processor halts in console mode. The console mode prompt > > > appears on the console terminal. Type the boot command:

```
>>> B ddnu (RET)
```

where:

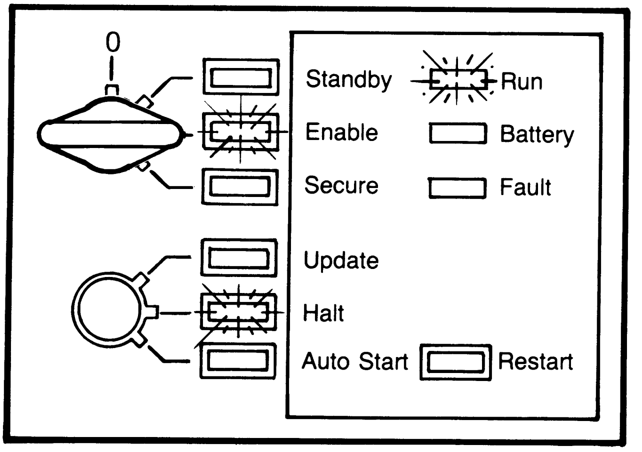
dd is the boot device type of the controller for the specific boot device (see Table 2–1)

n is the controller's VAXBI node number (hex) (see Table 2–1)

u is the unit number (1 digit) of the boot device (the unit number is printed on the front of the device when Field Service installs the system)

For example, if you want to boot the operating system from an RA60 disk (unit 0) attached to VAXBI node 4, you would type B DU40. If you do not specify a boot device, the operating system is booted from the default boot device.

After you type the B command, the Run light turns on and the processor boots the operating system.



MLO-847A-85

Table 2–1: Device Types and VAXBI Node Numbers of Boot Devices

Boot Device	Device Type	VAXBI Node Number
Local disk	DU	RA-type disk on system bus: use node number (in hex) of KDB50 adapter. The first KDB50 is normally node 4 for a system that is not part of a VAXcluster.*
		Disk on UNIBUS: use node number (in hex) of DWBUA (VAXBI-UNIBUS adapter). The DWBUA is normally node 0.*
Shared (HSC) disk	CS	Use A.
Diskette	CS	Use A.

* Determine the VAXBI node number by running the EEPROM Utility. See Section 3.5.1 about the VAXBI configuration.

When you specify a boot device with the B command, you must be sure to include all four characters in the specification, such as CSA1 or DU40.

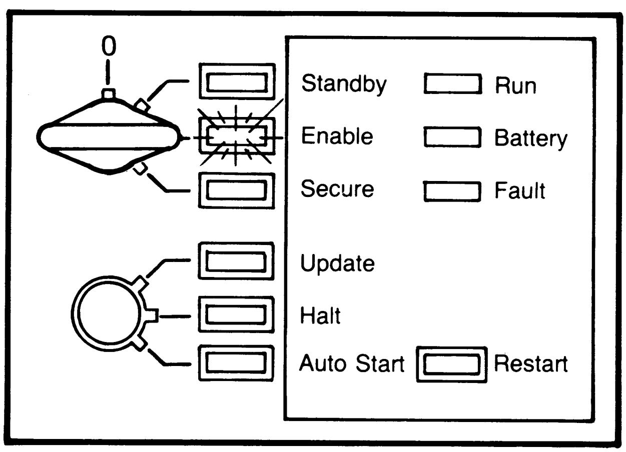
2.3 REBOOTING AFTER POWER IS ON

If power is on and the system halts (stops processing), you can reboot the operating system in one of two ways: by using the default boot device (normal reboot) or by specifying a boot device. Like the normal start, the normal reboot first runs self-test and initializes the system, then boots the operating system from the default boot device.

2.3.1 Normal Reboot from a Local Boot Device

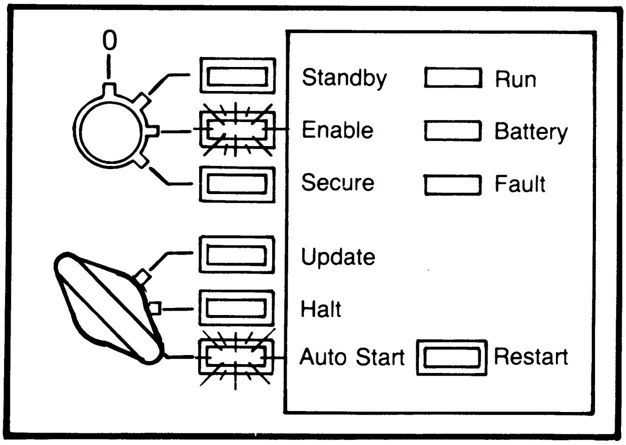
To reboot the operating system from a local boot device, follow these steps:

1. Turn the upper key switch to the Enable position so that the yellow light next to Enable is on.



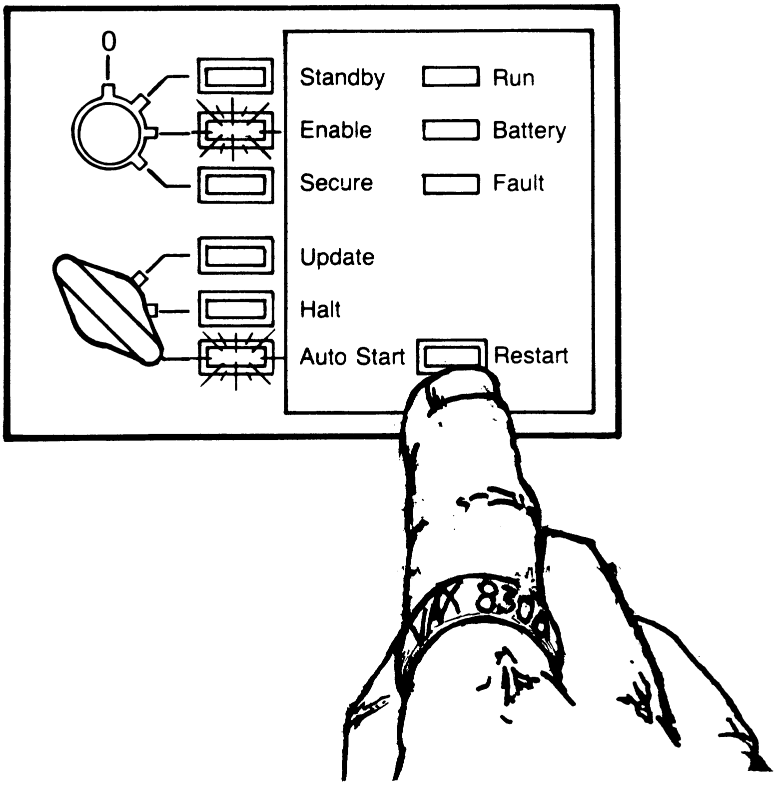
MLO-848A-85

2. Turn the lower key switch to the Auto Start position so that the green light next to Auto Start is on.



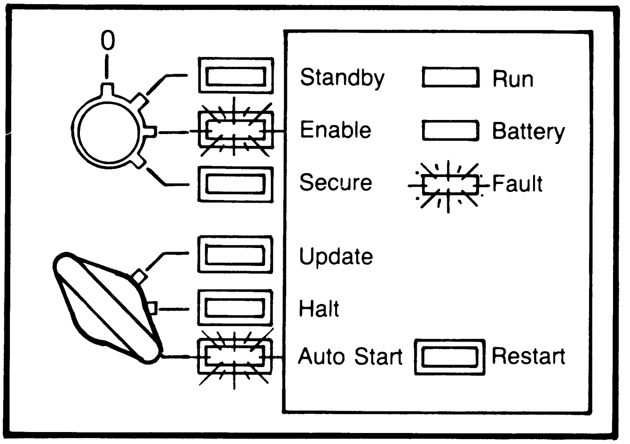
MLO-849A-85

3. Press the Restart button.



MLO-850A-85

The VAX 8200/8300 tests its hardware. During self-test, the Fault light is on.



MLO-851A-85

If the processor passes self-test, you see the following display on the console terminal:

```
#ABCDEFGHIJK.MN#
```

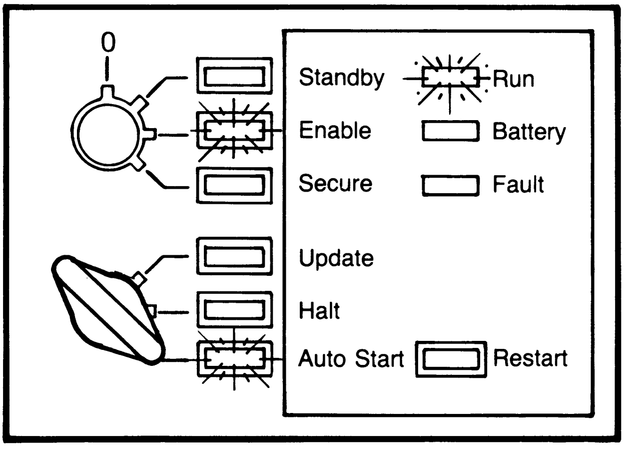
If the processor fails self-test, you see only a part of the string of letters. For example, you might see only #ABCD. See Section 2.8.3 for a complete description of the self-test display.

The system initializes itself: options on the system bus run their self-tests. If all the options pass their self-tests, the Fault light turns off and you see a display like the following on the console terminal:

```
0 . 2 . 4 . . . 8 9 A B C D . .
```

The display should contain only hexadecimal digits and periods. If any option fails self-test, the Fault light remains lit and at least one digit has a minus sign before it. See Section 2.8.3 for a complete description of this display.

The Run light turns on. The processor boots the operating system.



MLO-852A-85

2.3.2 Normal Reboot from a Shared Boot Device (VAXclusters)

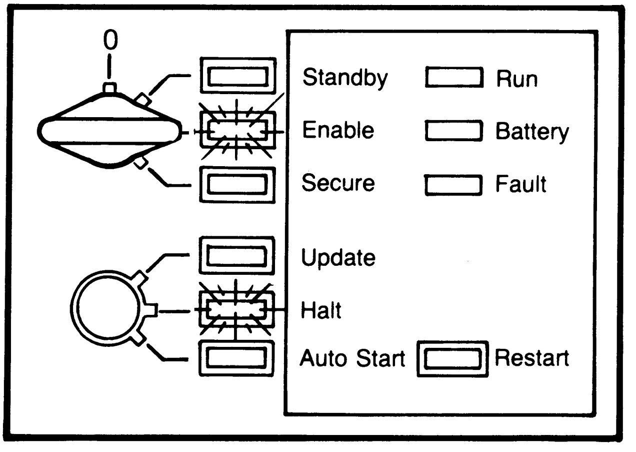
You can reboot VMS from a shared boot device by pressing the Restart button. The default boot device must be console drive 1 (CSA1). To reboot VMS from a shared boot device, follow these steps:

1. Insert the customized console diskette into console drive 1 (CSA1). If you have not yet customized the diskette, see Section 2.4.
2. Follow steps 1 through 3 in Section 2.3.1, Normal Reboot from a Local Boot Device.

2.3.3 Rebooting from a Specified Local Boot Device

With this method of rebooting, the processor and any options on the system bus do not run their self-tests. To reboot the operating system from a specific boot device, follow these steps. If your system is not part of a VAXcluster, skip step 1.

1. Insert the console diskette into console drive 1. (This step is necessary only for a system that is part of a VAXcluster.)
2. Turn the upper key switch to the Enable position so that the yellow light next to Enable is on. The lower key switch can be in any position, although Update is not recommended.



MLO-853A-85

3. Type CTRL/P at the console terminal. The console mode prompt >>> appears.
4. Type the boot command:

```
>>> B ddnu RET
```

where:

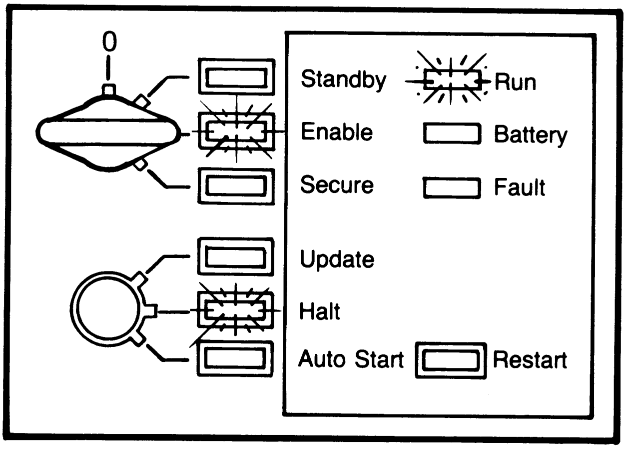
dd is the device type of the controller for the specific boot device (see Table 2–1)

n is the controller's VAXBI node number (hex) (see Table 2–1)

u is the unit number (1 digit) of the boot device (the unit number is printed on the front of the device when Field Service installs the system)

For example, if you want to boot the operating system from an RA81 disk (unit 0) attached to VAXBI node 4, you would type B DU40. If you do not specify a boot device, the operating system is booted from the default boot device.

After you type the B command, the status Run light turns on and the processor boots the operating system.



MLO-854A--85

2.3.4 Automatic Restarting After a Power Failure

The VAX 8200/8300 can automatically restart itself when power returns after a power failure. The operating system is restarted at the point where it stopped under three conditions:

1. Battery Backup supplied power to system memory during the power outage.
2. The lower key switch is set to Auto Start.
3. The upper key switch is set to Enable or Secure.

2.3.5 Automatic Rebooting

If a power outage lasts longer than approximately 10 minutes, an attempt to restart the operating system may fail. If the operating system cannot be restarted, the VAX 8200/8300 reboots the operating system from the default boot device. An automatic reboot always immediately follows a failed automatic restart. If your system is part of a VAXcluster and uses the console diskette drive as its default, you must have the console diskette inserted in console drive 1 for the automatic reboot to succeed.

An automatic reboot also occurs immediately after an operating system failure. The VAX 8200/8300 reboots the operating system under the following conditions:

1. The lower key switch is set to Auto Start.
2. The upper key switch is set to Enable or Secure.
3. The console diskette is in console drive 1 (a requirement for VAXclusters only).

2.4 FIRST-TIME BOOTING FROM A SHARED BOOT DEVICE (VAXCLUSTERS)

When you boot VMS for the first time from a shared boot device, you must run the program BOOT58 interactively. Once VMS is booted, you need to customize the command file CIBOO.CMD on the console diskette. After you customize CIBOO.CMD, you rename it to DEFBOO.CMD. For all subsequent starting or rebooting, the VAX 8200/8300 runs the command file DEFBOO.CMD to boot VMS from a specific boot device connected to the HSC controller. Therefore, after you have customized CIBOO.CMD and renamed it, all starting and rebooting from shared boot devices can use normal procedures (Sections 2.2 and 2.3).

2.4.1 Booting VMS Interactively with BOOT58

To boot VMS for the first time from a shared boot device, follow these steps:

1. Record the following data:
   - VAXBI node number of the CI adapter

     You can use the EEPROM Utility to show the VAXBI node number (see Chapter 3 about examining the VAXBI configuration).
   - CI node number of the HSC controller

     If your system has two HSC controllers, record both numbers. The CI node number is on the front of the HSC controller.
   - Unit number of a boot device connected to the HSC controller
   - Number of the local root directory specific to the VAX 8200/8300 that is booting VMS

The local root directory is [SYSr.SYSEXE], where r is a hexadecimal number between 1 and D. See the *VAX/VMS Guide to VAXclusters* for more information about root directories.

2. Insert the console diskette into console drive 1.
3. Insert a key into the lower control panel key switch. Turn the lower key switch to the Halt position.
4. Insert a key into the upper key switch. Turn the upper key switch to the Enable position.

   The VAX 8200/8300 tests its hardware. During self-test, the Fault light is on.

   If the processor passes self-test, you see the following display on the console terminal:

   ```
   #ABCDEFGHIJK.MN#
   ```

   If the processor fails self-test, you see only a part of the string of letters. For example, you might see only #ABCD. See Section 2.8.3 for a complete description of the self-test display.

   The system initializes itself: options on the system bus run their self-tests. If all the options pass their self-tests, the Fault light turns off and you see a display like the following on the console terminal:

   ```
   0 . 2 . 4 . . . 8 9 A B C D . .
   ```

   The display should contain only hexadecimal digits and periods. If any option fails self-test, the Fault light remains lit, and at least one digit has a minus sign before it. See Section 2.8.3 for a complete description of this display.

   The processor halts in console mode. The console mode prompt > > > appears on the console terminal.

5. Type the boot command:

   ```
   >>> B/R5:800 CSA1
   ```

   The B/R5:800 command loads the value 800 into General Purpose Register R5 to run the program BOOT58. The prompt BOOT58> appears on the console terminal. The following steps show how to enter required values into General Purpose Registers.

6. Type the command:

```
BOOT58> D/G 0 20
```

The command D/G deposits a value into a specified General Register. Here, you deposit the value 20(hex) into General Register 0. The value 20 is the device code for a device attached to an HSC controller.

7. Type the command:

```
BOOT58> D/G 1 n
```

The letter n represents the VAXBI node number (hex) of the CI adapter.

8. If your VAXcluster has one HSC controller, type the command:

```
BOOT58> D/G 2 p
```

The letter p represents the CI node number (hex) of the HSC controller.

If your VAXcluster has two HSC controllers, type the command:

```
BOOT58> D/G 2 0p0q
```

The letters p and q represent the CI node numbers (hex) of the two HSC controllers.

9. Type the command:

```
BOOT58> D/G 3 u
```

The letter u represents the unit number (hex) of the boot device connected to the HSC controller.

10. Type the command:

```
BOOT58> D/G 5 r0000000
```

The letter r represents the number (hex) of the local root directory specific to the system on the VAXcluster that is booting VMS.

11. Type the command:

```
BOOT58> D/G E 200
```

The number 200(hex) is the starting address of the program VMB.

12. Type the command:

```
BOOT58> LOAD VMB.EXE/START:200
```

This command loads the program VMB.EXE into memory. VMB is the primary loader for VMS.

13. Type the command:

```
BOOT58> START 200
```

This command starts VMB and the booting of VMS. You should see a display like the following on the console terminal:

```
VAX/VMS Version 4.4 01-JAN-86 00:00

%%%%%%%%%%% OPCOM 29-JUN-1986 10:29:40.72 %%%%%%%%%%%
Logfile has been initialized by operator _OPA0:
ogfile is SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;85

%SET-I-INSET, login interactive limit=64, current interactive value = 0
SYSTEM job terminated at 29-JUN-1986 10:29:57.46
```

14. Press (RET) on a terminal connected to the system. Log in to the SYSTEM account. See the *Guide to VAX/VMS System Management and Daily Operations* for information on logging in to the SYSTEM account.

2.4.2 Customizing the Command File CIBOO.CMD

After you boot VMS, you should customize the command file CIBOO.CMD and rename it DEFBOO.CMD. BOOT58 automatically runs DEFBOO.CMD when you boot VMS again. The file DEFBOO.CMD directs the booting procedure to a particular boot device connected to the HSC controller and VMS is booted from that device.

The customized command file should contain the same information that you typed in response to the BOOT58 prompts in Section 2.4.1. To customize the command file CIBOO.CMD, type the following commands on a video terminal:

1. `$ RUN SYS$SYSTEM:SYSGEN`

   The prompt SYSGEN> appears.

2. Type the following commands in response to the SYSGEN> prompt.

   ```
   SYSGEN> CONNECT CONSOLE
   SYSGEN> EXIT
   ```

   The dollar sign $ prompt appears.

3. Type:

   ```
   $ EXCHANGE
   ```

   The prompt EXCHANGE> appears.

4. Type the following commands in response to the EXCHANGE> prompt:

```
EXCHANGE> MOUNT CSA1:
EXCHANGE> COPY CSA1:CIBOO.CMD *
EXCHANGE> EXIT
```

   These commands copy the command file CIBOO.CMD from the diskette to the system disk so that you can modify parameters in the file. The EXIT command returns you to the dollar sign $ prompt.

5. Type:

```
$ EDIT CIBOO.CMD
```

   The file CIBOO.CMD appears on the screen:

```
!CIBOO.CMD : Boot command file to boot a VAX 8200/8300 from an HSC disk.
!
!
! Note: "n", "p" (and "q"), "u", and "r" are single hexadecimal characters
!
!D/G 0 20                    ! CI Port Device Type Code
!D/G 1 n                     ! n = CI adaptorTs VAXBI node number
!D/G 2 p                     ! Use the HSC controller at CI node p
D/G 2 0p0q                   ! Use either HSC controller at CI nodes p and q
!D/G 3 u                     ! u = Disk drive unit number
D/G 4 0                      ! Boot Block LBN (not used)
!D/G 5 r0000000              ! r = the system root [SYSr...]
D/G E 200
LOAD VMB.EXE/START:200
START 200
```

   Change the parameters n, p (and q), u, and r so they conform to your system. These letters represent the same parameters as those described in steps 6 through 13.

   The following file is an example of an edited CIBOO.CMD. In the example, the VAXBI node number is 5, the CI node number is 1, the unit number is 4, and the root directory number is 6.

```
!CIBOO.CMD : Boot command file to boot a VAX 8200/8300 from an HSC disk.
!
!
! Note: "n", "p" (and "q"), "u" and "r" are single hexadecimal characters
!
D/G 0 20                    ! CI Port Device Type Code
D/G 1 5                     ! n = CI adaptorTs VAXBI node number
D/G 2 1                     ! Use the HSC controller at CI node p
D/G 3 4                     ! u = Disk drive unit number
D/G 4 0                     ! Boot Block LBN (not used)
D/G 5 60000000              ! r = the system root [SYSr...]
D/G E 200
LOAD VMB.EXE/START:200
START 200
```

6. Type:

```
$ EXCHANGE
```

   The prompt EXCHANGE> appears.

7. Type the following commands:

```
EXCHANGE> COPY CIBOO.CMD CSA1:DEFBOO.CMD
EXCHANGE> EXIT

$ DISMOUNT CSA1:
```

   These commands rename the file CIBOO.CMD to DEFBOO.CMD.

2.5 USING AN ALTERNATE BOOT BLOCK WHEN BOOTING VMS FROM A LOCAL BOOT DEVICE

If a booting problem requires an alternate boot block, you can use the boot block on the console diskette when booting from a local boot device attached to the KDB50 adapter. To use the boot block on the diskette, you run a command file KDBBOO.CMD. You must customize this file before you can run it. The following steps show how to customize the command file, and how to run it:

1. Record the following data:
   - VAXBI node number of the KDB50 adapter
   - Unit number of the local boot device
   - Number of local root directory
2. Insert the console diskette into console drive 1.

3. On a video terminal from the SYSTEM account in VMS, type the following command:

```
$ EXCHANGE
```

The EXCHANGE> prompt appears.

4. Type the following commands to copy the command file KDBBOO.CMD to the SYSTEM account:

```
EXCHANGE> MOUNT CSA1:
EXCHANGE> COPY CSA1:KDBBOO.CMD
EXCHANGE> EXIT
```

5. Type the following command:

```
$ EDIT KDBBOO.CMD
```

The following file appears:

```
!KDBBOO.CMD : Boot command file to boot a VAX8200 from a KDB50 disk
! bypassing the boot block. (i.e. utilizing the boot
! block and VMB.EXE from the RX50 console device.)
!
! Note: "n", "u", and "r" represent single hexadecimal characters
!
!
D/G 0 21                        ! KDB50 Device Type Code
!D/G 1 n                        ! n = VAXBI node number of KDB50
!D/G 3 u                        ! u= Disk drive unit number
!D/G 5 r0000000                 ! r = number of system root directory
D/G E 200
LOAD VMB.EXE/START:200
START 200
```

Change the letters n, u, and r for your system.

6. After you edit the file, type the following command:

```
$ EXCHANGE
```

7. Type the following commands to copy the file back to the diskette:

```
EXCHANGE> COPY KDBBOO.CMD CSA1:KDBBOO.CMD
EXCHANGE> EXIT
$ DISMOUNT CSA1:
```

When you need to use the command file KDBBOO.CMD, run the program BOOT58 as described in Section 2.4.1. To run the command file, type @KDBBOO.CMD in response to the BOOT58> prompt.

2.6 STAND-ALONE BACKUP

Stand-alone backup lets you copy VMS from a tape or disk to an empty disk. The stand-alone backup diskettes are issued with the VMS software.

Before running stand-alone backup, you must first insert the console diskette into the console drive and run the command file CSABOO.CMD, which loads the program VMB. (If the system has a CI adapter, VMB loads microcode from the console diskette into the adapter.) The command file then instructs you to insert the first stand-alone backup diskette.

To use stand-alone backup, follow these steps:

1. Insert the console diskette into console drive 1.
2. Turn the upper key switch to the Enable position.
3. Type CTRL/P at the console terminal to enter console mode.
4. Type the command:

   ```
   >>> B/R5:800 CSA1
   ```

   The BOOT58> prompt appears.
5. Type the command:

   ```
   BOOT58> @CSABOO.CMD
   ```

   The following display appears:

   ```
   !
   ! Stand-alone CSA1 boot command file - CSABOO.CMD
   !
   !
   D/G 0 40                  ! DEVICE TYPE (CONSOLE STORAGE DEVICE)
   D/G 1 0
   D/G 2 0
   D/G 3 1                   ! UNIT NUMBER
   D/G 4 0                   ! BOOT BLOCK LBN (UNUSED)
   D/G 5 0                   ! SOFTWARE BOOT FLAGS
   D/G E 200                 ! ADDRESS OF WORKING MEMORY
   LOAD VMB.EXE/START:200    ! LOAD PRIMARY BOOTSTRAP
   START 200                 ! AND START IT
   ```
6. The command file instructs you to remove the console diskette and insert the first stand-alone backup diskette to begin the stand-alone backup procedure.

For information on stand-alone backup, refer to the *VAX/VMS Guide to System Management and Daily Operations.*

2.7 CONSOLE COMMANDS

You can use VAX 8200/8300 console commands (such as B and T) only on the console terminal (the terminal connected to serial line unit 0 of the processor) and only when the terminal is in console mode. Console mode has exclusive use of the system.

Type the commands in response to the console mode prompt >>>. To enter console mode, set the upper key switch to the Enable position and type CTRL/P on the console terminal.

Number codes in the form ?nn, where nn is a 2-digit number in hexadecimal, can appear on the console terminal after a command. These codes indicate an error or a halted processor. See Appendix B of this manual or the *VAX 8200/8300 Mini-Reference* for descriptions of these codes.

If you type a character not allowed by the console, the console terminal beeps.

Table 2–2 lists the console commands. Examples of console commands appear after the table.

Table 2–2: Console Commands

Command	Description
B [boot device]	BOOT: Boots operating system from default boot device or from specified device.
C	CONTINUE: Starts processor executing VAX instructions, beginning at address currently in the program counter (PC).
D [address] [data]	DEPOSIT: Deposits data into specified address. See Appendix E for a complete description of the D command.
E [address]	EXAMINE: Prints the data contained at the specified address. See Appendix E for a complete description of the E command.
I	INITIALIZE: Initializes the processor. See the *KA820 Processor Technical Manual* for a complete description of processor initialization.
N	NEXT: Causes processor to execute the macro instruction at address currently in the program counter. Use this command to step through macrocode one instruction at a time.
S [address]	START: Starts processor executing macro instructions, beginning at specified address. With no address, the S command functions like the C command.
T	TEST: Initializes processor and runs self-test. See Section 2.4.
X [address] [no. of bytes]	Reserved for use by DIGITAL. Reads or writes specified number of bytes at specified address.
Z [VAXBI node]	FORWARD: Logically connects the console to a specified VAXBI node. Lets you forward subsequent commands, except CTRL/P, to the specific VAXBI node. Typing CTRL/P after typing the Z command stops any further forwarding. To forward a CTRL/P, press ESC before typing CTRL/P.
! [comments]	Lets you record comments when you are using a hard-copy terminal. The system ignores all characters typed after !.

(Continued on next page)

Table 2–2: Console Commands (Cont.)

Command	Description
CTRL/P	Causes processor to enter console mode. When the processor is already in console mode, CTRL/P aborts the processing of the last command typed and stops the forwarding of commands (if forwarding was invoked with the Z command).
CTRL/Q	Resumes printing on console terminal that was stopped with CTRL/S.
CTRL/S	Stops printing on console terminal. You cannot type a command after CTRL/S until you type CTRL/Q or CTRL/P.
CTRL/U	Deletes the current line. The system then prompts you for another command.
BREAK	Selects the baud rate for the console. Each time you press BREAK, you advance the console baud rate to the next value. The baud rates are 150, 300, 600, 1200, 2400, 4800, 9600, and 19200. The console baud rate and the console terminal baud rate should match. To show the current console terminal baud rate, press the Set-Up key on the console terminal and follow the instructions given in the terminal's user guide. The console baud rate default is 1200.
ESC CTRL/P	Forwards a CTRL/P to a VAXBI node specified by the Z command.
RET	Terminates a command. All commands must be terminated by RET except ESC, BREAK, and CTRL characters.

Example of a Console Command Sequence

`>>> T`	! Run self-test.
`>>>Z F`	! Forward subsequent commands to ! processor on node F.
`>>>` ESC CTRL/P	! Ensure that processor on node F is in ! console mode.
`>>> E/E 3`	
`P 20098176 08`	! Examine the RCX50 self-test ! enable/disable data contained ! in the attached processor's EEPROM.
`>>> D/E 3 18`	! Deposits 18(hex) into the EEPROM of the ! attached processor to disable the RCX50 ! self-test.
`>>>` CTRL/P	! Stop the forwarding of commands to ! processor on node F. ! Subsequent commands ! are processed by the primary ! processor.
`>>>B DU45`	! Boot operating system from disk on ! node 4, unit 5.
`>>>` CTRL/P	! Abort last command.
`>>> B`	! Boot operating system from default ! boot device.

2.8 TROUBLESHOOTING WITH SELF-TEST

If you have trouble starting or operating your system, the VAX 8200/8300 self-test can help you diagnose the problem; a DIGITAL Field Service representative can then quickly repair the hardware. The VAX 8200/8300 self-test verifies that the processor hardware and the VAXBI nodes are functioning properly. Self-test runs automatically when you turn on the system; you can also run self-test from the console terminal. This section tells you how to run self-test, when it occurs, what hardware is tested, and how to interpret self-test indicators and displays.

2.8.1 Self-Test Overview

Two areas of the system are tested: the primary processor and the nodes on the system bus (the VAXBI bus). The VAXBI bus connects the primary processor to options, such as I/O controllers, bus adapters, memory, and other processors.

Nodes are addresses on the VAXBI bus and are represented by hexadecimal numbers. A node exists for each VAXBI option. If the option is a bus adapter, the node includes the bus, its attached devices, and the adapter itself.

Each VAXBI option has its own self-test. Self-tests specific to VAXBI options are described in the technical manuals for the options.

The self-test runs under five conditions, described in Table 2–3.

Table 2–3: When Self-Test Occurs

Action	Upper Key Switch Position	Lower Key Switch Position	System Response
Turn on the system.	Secure or Enable	Auto Start	The system runs self-test, then boots the operating system.
Turn on the system.	Enable	Halt	The system runs self-test and halts in console mode.
Press the Restart button.	Enable	Auto Start	The system runs self-test, then boots the operating system.
Press the Restart button.	Enable	Halt	The system runs self-test and halts in console mode.
Type T at the console terminal.	Enable	Auto Start or Halt	The system runs self-test, clears VAXBI memory, and halts in console mode.

In all cases, self-test runs no longer than 10 seconds.

Characters printed on the terminal and lights on the control panel and in the processor drawer indicate the success of self-test. See Section 2.8.3.

2.8.2 Running Self-Test

To verify the integrity of primary processor hardware and the VAXBI nodes after the system has power, you can run self-test from the console terminal or from the control panel. Both methods clear system memory.

2.8.2.1 From the Console Terminal

1. Turn the upper key switch to the Enable position.
2. Type CTRL/P to enter console mode. The >>> prompt appears.
3. Type T to begin self-test.

Section 2.8.3 lists the hardware that is tested and describes self-test indicators and displays.

2.8.2.2 From the Control Panel

1. Turn the upper key switch to the Enable position.
2. Turn the lower key switch to the Halt position.
3. Press the Restart button on the control panel.

   NOTE
   If the lower key switch is in the Auto Start position, pressing the Restart button causes the primary processor to boot the operating system, in addition to running self-test. (See Chapter 1 for more information on the Restart button.)

2.8.3 Interpreting Self-Test Results

Characters printed on the console terminal, the Fault light on the control panel, and lights in the processor drawer indicate the status of self-test. When self-test begins, all indicators and displays report simultaneously.

2.8.3.1 Letters Printed on Console Terminal: Primary Processor Self-Test — As each section of hardware on the primary processor passes self-test, a letter or period corresponding to that hardware appears on the console terminal. The hardware and the corresponding characters are as follows:

Processor Hardware Tested	Printed Letter or Period
Control store checksum	A
IE chip internals	B
DAL interface	C
M chip internals	D
Backup translation buffer	E
Cache memory	F
IE chip/M chip interactions	G
PCntl internals	H
Boot code checksum	I
Boot RAM	J
F Chip	K
Reserved for DIGITAL	.
Diskette controller	M
BIIC (VAXBI interface chip)	N

The pound sign # appears at the start and end of the processor self-test. If any hardware is disabled or not installed, a period replaces the letter for that hardware.

> **NOTE**
> **An additional pound sign # or other characters may appear before the self-test display. These characters can be ignored.**

If all available hardware is installed and enabled, the following display appears on the terminal:

```
#ABCDEFGHIJK.MN#
```

If self-test encounters an error, the console error code ?40 appears below the sequence of letters (see Appendix B). The last letter printed indicates the last processor section that passed self-test. For example, suppose you see the following:

```
#ABC
?40
```

This display tells you that the M chip (D) has failed self-test. The last section to pass was the DAL interface. Call DIGITAL Field Service and explain at what point the processor self-test failed.

Detailed results of self-test for an attached processor do not appear on the console terminal. If an attached processor fails self-test, the attached processor's VAXBI node number has a minus sign before it.

2.8.3.2 Fault Light: VAXBI Node Self-Tests — The Fault light turns on during self-test, indicating that some VAXBI nodes have not successfully completed their individual self-tests. When all nodes have passed their self-tests, the Fault light turns off. If the Fault light remains on for more than 10 seconds, a node has failed self-test. A fault on the node could be the result of hardware errors in attached devices, transmission lines, or a VAXBI option. Check the numbers printed on the console terminal (Section 2.8.3.3) and the module self-test lights (Section 2.8.3.4).

2.8.3.3 Numbers Printed on the Console Terminal: VAXBI Node Self-Tests — While the primary processor is testing itself, other VAXBI nodes test themselves. Self-test prints a series of hexadecimal numbers and dots below the letters (Section 2.8.3.1) on the console terminal. A dot indicates a node that is not used (nonexistent). Each number is the address of a particular VAXBI node and indicates that the node has passed self-test.

A number preceded by a minus sign indicates that the node has failed self-test. Like the Fault light, the minus sign indicates a fault in a VAXBI node or the VAXBI bus itself.

If a node fails self-test so that you cannot use the EEPROM Utility, call DIGITAL Field Service. If the faulty hardware still allows you to boot software and identify nodes, use the EEPROM Utility (Chapter 3) to determine what node has failed its self-test. In the General section of the EEPROM Utility, the VAXBI configuration (see Section 3.5.1) shows node numbers and corresponding hardware. Find the hardware description of the node number with the minus sign. Then call DIGITAL Field Service and explain what hardware is broken.

For example, suppose you see the following:

```
0 . 2 . 4 . -6 . 8 9 A B C -D . .
```

This display tells you that:

- Nodes 0, 2, 4, 8, 9, A, B, and C passed self-test.
- Nodes 6 and D failed self-test.
- Nodes 1, 3, 5, 7, E, and F are not used.

Use the EEPROM Utility to find what options and devices (if any) are included in nodes 6 and D.

When the primary processor and all other VAXBI nodes have passed their self-tests, you should see a display like the following:

```
#ABCDEFGHIJK.MN#

0 . 2 . 4 . 6 . 8 9 A B C D . .
aaaaaaaa
?01

PC = bbbbbbbb
```

The letters aaaaaaaa represent the amount of available system memory in hexadecimal notation. For example, 00400000 indicates 4 megabytes (4096 kilobytes) of memory. The letters bbbbbbbb represent the address in the Program Counter. The halt code ?01 indicates that self-test completed successfully (see Appendix B).

2.8.3.4 Module Lights in Processor Drawer: VAXBI Node Self-Tests — The VAXBI bus consists of at least six slots, each of which can contain a VAXBI module.

A VAXBI option, such as an adapter, consists of one or more modules. Each VAXBI module has a yellow self-test light. When lit, a yellow light indicates that the option has passed its self-test. An unlit light on any module indicates that the option is broken.

The self-test lights for the 12-slot and 24-slot systems are shown in Figures 2–3 and 2–4, respectively.

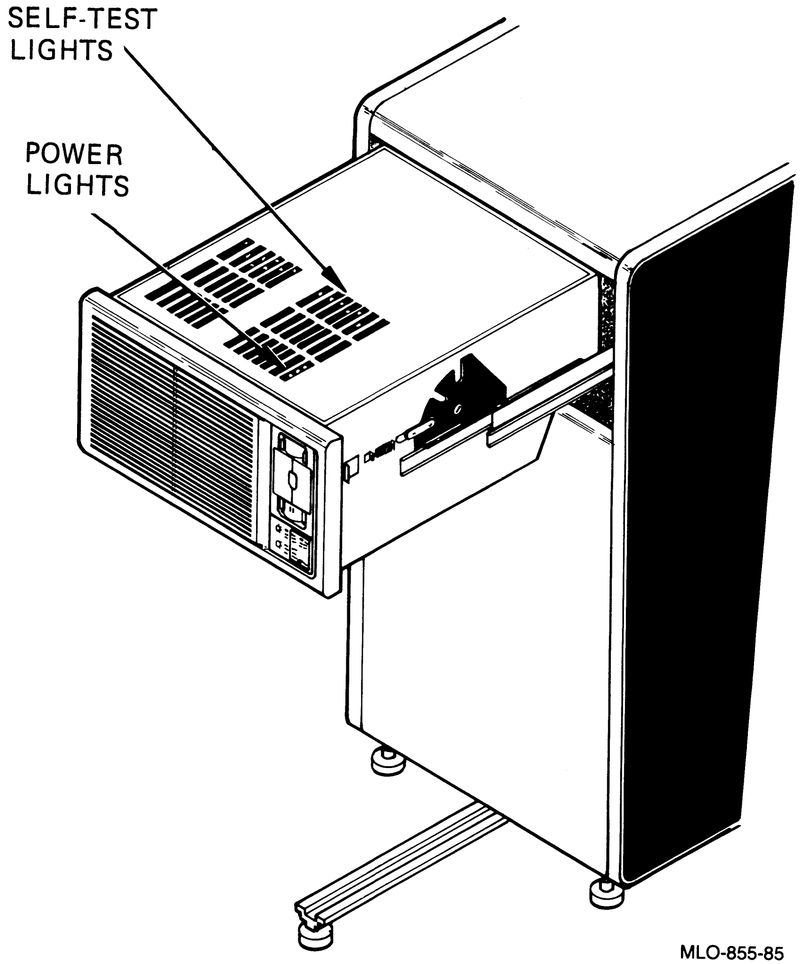


Figure 2–3: Self-Test Lights for VAXBI Modules in the 12-Slot VAXBI Bus

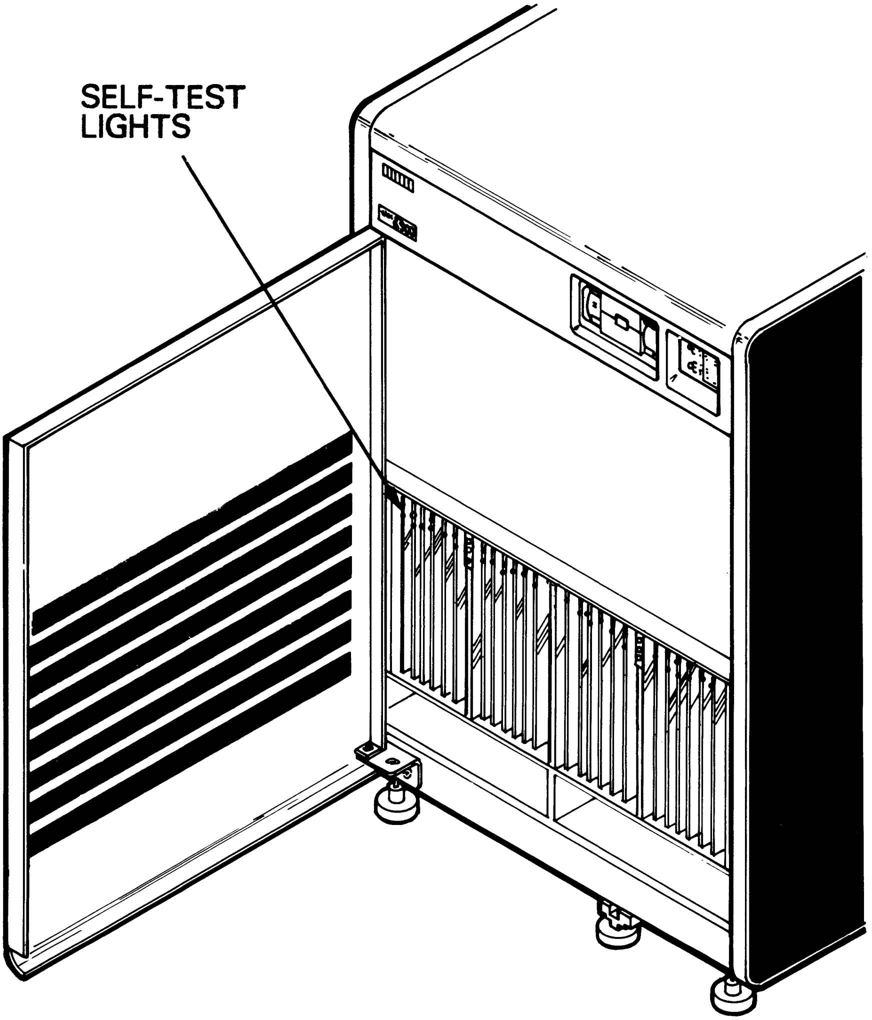


Figure 2–4: Self-Test Lights for VAXBI Modules in the 24-Slot VAXBI Bus

The rightmost VAXBI slot holds the primary processor module. Next to its yellow self-test light, the primary processor module has a red light that turns off when the processor starts executing VAX instructions. The red light remains on if the processor fails self-test.

An attached processor also has a red light and yellow self-test light that operate in the same way as those on the primary processor.

2.8.4 Fast Self-Test for Real-Time Applications

In most situations, you will want the system to use the normal 10-second self-test, which gives maximum coverage of system hardware. If you are operating your system in a real-time environment, however, you might want a very fast recovery from a power failure. In this case, use the fast self-test, which performs an abbreviated test on all VAXBI nodes and tests only cache memory and the F chip in the primary processor. The fast self-test requires less than 0.25 seconds to complete. To make the primary processor run the fast self-test, you must change a jumper connection, as shown in Appendix C, and set the VAXBI self-test timeout to 0 by running the EEPROM Utility. A prompt for the VAXBI self-test timeout appears in the General section of the EEPROM Utility dialog. Changing the VAXBI self-test timeout to 0 with the EEPROM Utility actually sets the timeout at 250 milliseconds.

The fast self-test and the normal self-test run in the same situations described in Table 2–3.

The EEPROM Utility 3

The EEPROM Utility lets you display and modify information stored in the two EEPROMs (Electrically Erasable Programmable Read-Only Memory) on the VAX 8200/8300 processor. Examples include displaying the configuration of VAXBI nodes and changing the default boot device and the default console baud rate. The EEPROM Utility operates as a dialog; by answering questions printed on the terminal, you can change or examine data. The VAX 8200/8300 Util Prog Flp diskette, included with the operations kit, contains the EEPROM Utility.

The following sections discuss why you might need to run the program, describe how to run it, and provide examples from the EEPROM Utility dialog.

3.1 WHEN TO USE THE EEPROM UTILITY

You should use the EEPROM Utility when you need to examine or change the data shown in Table 3–1. Most data stored in the EEPROMs rarely needs to be changed.

The EEPROM Utility lets you easily change data, but you should be careful. If you make a mistake, however, you can abort the session with the dialog. You can also save a copy of correct EEPROM data on the EEPROM Utility diskette; then, if you change EEPROM data and later discover a mistake, you can restore EEPROM data from a copy of correct data.

You can always examine EEPROM data without changing it. On most systems, you will need to change or examine only the data shown in the examples in Section 3.5.

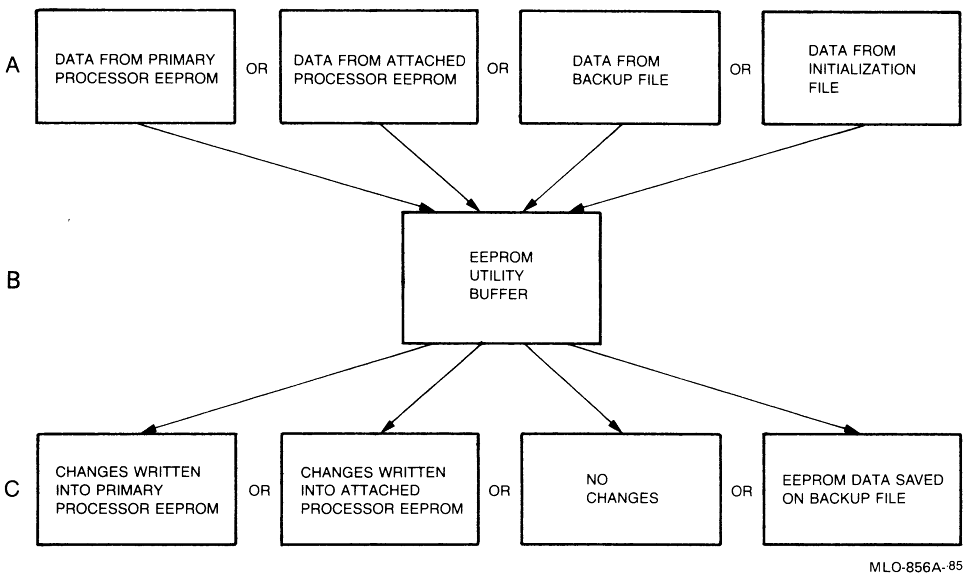
3.2 OVERVIEW OF THE EEPROM UTILITY

The EEPROM Utility dialog is divided into eight sections. Table 3–1 summarizes, for each section, the data you can examine or change.

Table 3–1: Sections of the EEPROM Utility Dialog

Section	Summary
Startup	Display general help. Choose source of data to be loaded in EEPROM Utility Buffer.
General	Display a table that shows what VAXBI options are on the VAXBI nodes. Display a summary of important EEPROM data. Examine or modify the following: • Processor serial number • RCX50 self-test (enable or disable) • VAXBI self-test timeout • F chip (enable or disable) • Cache Memory (enable or disable)
Console	Examine or modify the following: • Default console baud rate • Logical console VAXBI node number
Boot	Display summary of boot code. Change the default boot device. Verify that boot code does not contain errors. Display boot code in hexadecimal notation. Add or remove boot code. Change the DU boot device controller's interrupt and polling address.
Microcode Patch	Examine or modify processor and patch revision numbers. Verify that patches do not contain errors. Load a new set of patches. List patches. Add a single patch. Remove a single patch.
Hexadecimal Examine/Deposit	Modify any data in the EEPROM, using Examine and Deposit commands.
Ethernet	Examine or modify the following: • Ethernet load server address • Expected boot message • Number of bytes compared in boot message for recognition • First offset to boot message where comparison begins • Second offset to boot message where comparison begins
End of Pass	Make another pass of the dialog. Save EEPROM data on the diskette. Write changes into EEPROM. Exit without making changes.

When you change or examine EEPROM data, the EEPROM Utility protects that data by using a temporary workspace called a buffer. In the first section of the dialog, you load EEPROM data into the buffer from one of four sources (described in Section 3.3.5); you then make all subsequent changes on the data in the buffer. In the last section of dialog, you can write the buffered data into an EEPROM, write a backup file of EEPROM data on the diskette, or, if you decide not to make any changes, simply abort the session with the EEPROM Utility. Therefore, changes you make using the EEPROM Utility are not copied immediately into the EEPROMs.



A — Load the buffer with EEPROM data from a specified source.

B — Modify or examine data in the buffer by answering dialog questions.

C — Write buffered data into EEPROMs, save data on diskette, or exit without making changes.

Figure 3–1: How the EEPROM Utility Changes EEPROM Data

Because the EEPROMs contain important system data, you should be careful when using the EEPROM Utility. To avoid mistakes, follow some basic steps:

1. Look at Table 3–1 and the examples in Section 3.5 and decide what EEPROM data you want to examine or change.
2. Run the EEPROM Utility by following the procedure described in Section 3.4.
3. Find the section of EEPROM Utility dialog that covers the EEPROM data you want to examine or change.
4. Examine or change EEPROM data by answering dialog questions.
5. If you have made changes to the data, write the changes into the EEPROMs. Later, when you are satisfied that the VAX 8200/8300 system runs properly with these changes, you can make a backup copy of the EEPROM data.

   If you have only examined data, abort the session with the dialog.
6. Exit the EEPROM Utility.

3.3 HOW TO USE THE EEPROM UTILITY

This section explains how to use the EEPROM Utility to examine or change the data you have chosen. Explanations cover using commands, obtaining help, choosing a section of dialog and answering questions, loading the buffer, and exiting the EEPROM Utility.

Section 3.4 explains how to run the EEPROM Utility.

3.3.1 EEPROM Utility Commands

Table 3–2 describes commands you can use during the EEPROM Utility dialog. For example, you can return to the previous question, skip to the End of Pass section, and temporarily halt the printing of data.

Table 3–2: EEPROM Utility Commands

Command	Description
∧	Displays previous question.
>	Skips over any remaining sections to the End of Pass section.
h or H	Displays a general help list. The current question appears again.
?	Displays help specific to the question. Not all questions have help available.
CTRL/C	Aborts the EEPROM Utility and exits to the VAX Diagnostic Supervisor. If typed during the listing of patches or boot code, CTRL/C tops the listing, and the next question appears. After you type CTRL/C, the listing may continue to be printed for a moment before stopping. Do not type a second CTRL/C, as this will abort the EEPROM Utility.
CTRL/U	Deletes all characters typed on the current line. The current question appears again.
CTRL/S	Temporarily halts the printing of data. Type CTRL/Q to resume printing.
CTRL/Q	Resumes printing that was stopped by a CTRL/S.
CTRL/R	Displays the current line again.
CTRL/P	Aborts both the EEPROM Utility and the Diagnostic Supervisor and exits to console mode. The console mode > > > prompt appears.

3.3.2 How to Obtain Help

To obtain help specific to a question, type a question mark ? in response to the question. Not all questions have help available.

You can obtain a general help list by typing H any time during the dialog.

3.3.3 How to Choose a Section

Type a question mark ? at the beginning of a section to list the section's parameters. Respond NO to skip over one section to the next.

The beginning of each section (except Startup and End of Pass) is marked by the EE> symbol, followed by a question, such as:

```
EE> Are you interested in the general section Y/N {N} ? ?

This section provides routines to
  Display VAXBI configuration
  Display Summary of EEPROM data
  Perform EEPROM Sanity check
  Examine/modify the following:
       CPU Serial number
       VAXBI self-test timeout
       F Chip enable
       RCX50 self-test enable
       Cache enable
```

3.3.4 How to Answer Questions

Answer the questions by typing YES (or Y) or NO (or N) or by entering information required for examination or modification of data.

You can use the default response for the question or prompt by pressing the RETURN key:

- The default response for all questions is NO, represented by the N in braces. For example:

  ```
  Do you want help Y/N {N} ?
  ```

- Default responses for prompts are enclosed in braces. For example:

  ```
  Enter node number in Hex: [02] :
  ```

If you type an invalid response to a question or prompt, an error message appears and the current line is displayed again. Some questions will appear only if you answered the previous one in a specific way.

If you want to return to the previous question, type a caret ∧. You can use this command to verify that a change you made is correct. For an example of verifying changes, see Section 3.5.2.

3.3.5 How to Load the Buffer

You load the EEPROM Utility buffer in the first section of dialog, Startup. There, the dialog asks you what EEPROM data you want to load into the EEPROM Utility buffer. Choose the data you want to examine or modify. You can load data from one of four sources:

- Primary processor EEPROMs
- Attached processor EEPROMs
- A file on the diskette (KAINITx.SYS) called the initialization file
- A backup file (EEPROM.IMA) you have previously written on the diskette by using the EEPROM Utility

EEPROM—In most cases, you will load data from the EEPROMs, either the primary processor EEPROMs or the attached processor EEPROMs.

Initialization File—You should load data from the initialization file only when you have not previously saved EEPROM data and you believe that much of the data in the EEPROMs contains errors. Use this file to rebuild EEPROM data. The initialization file contains EEPROM data that is basic for all systems; the file contains neither customization for your system nor changes you previously made to the EEPROM data. After you load the initialization file, you may have to update some of the data in the buffer. Note that loading this file does not erase it from the diskette.

Appendix A lists the values for parameters of the initialization file.

Backup File (saved image file)—You should load data from the backup file if you have discovered an error in EEPROM data. The backup file, which you can write in the End of Pass section, contains correct EEPROM data. If you have not yet written the backup file, it contains the data in the initialization file. Note that loading the backup file does not erase it from the diskette. Section 3.3.6 describes how to write a backup file.

Startup Section Dialog

The dialog in the Startup section is as follows:

```
Source for loading EEPROM work buffer is the primary processor
Want to change this Y/N {N} ?
```

Respond NO if you want the EEPROM Utility to load its buffer with data from the EEPROM on the primary processor.

Respond YES if you want the EEPROM Utility to load its buffer with data from a source other than the primary processor. Choose the source of data in one of the following questions.

```
Want to load initialization file KAINITx.SYS Y/N {N} ?
```

Respond YES to load the buffer with data from the initialization file. The EEPROM Utility takes approximately five seconds to load the initialization file into its buffer.

Because some processors (early versions) have one EEPROM and others have two, there are two versions of KAINITx.SYS. Processors with one EEPROM use KAINIT.SYS; processors with two EEPROMs use KAINIT-2.SYS. Most processors have two EEPROMs. You do not need to specify the version number when loading the initialization file; the EEPROM Utility automatically loads the correct version.

Respond NO if you want to load data from a previously written backup file or from EEPROMs on an attached processor.

```
Want to load buffer from EEPROM saved image file on CSA1 Y/N {N} ?
```

Respond YES to load data from the backup file on the diskette. The EEPROM Utility takes approximately 5 seconds to load the initialization file into its buffer.

Respond NO if you want to load data from EEPROMs on an attached processor.

```
Want to load buffer from EEPROM of an attached processor Y/N {N} ?
```

Respond YES if you want to load data from EEPROMs on an attached processor. The EEPROM Utility prints a table like the following that shows which VAXBI nodes contain processors:

```
NODE NUMBER              KA820 PROCESSOR TYPE
---------------------------------------------------
02                       KA820-PRIMARY   2 EEPROMs
0F                       KA820-ATTACHED  2 EEPROMs

Enter node number in Hex:  [02] :  0F
```

3.3.6 How to Exit the EEPROM Utility

To exit the EEPROM Utility at any point in the dialog, type a right arrow >, which brings you to the last section, End of Pass.

In the End of Pass section, you can make another pass of the dialog, abort the session with the dialog (making no changes), or write the changes you have made to three places:

- EEPROMs from which you loaded data
- EEPROMs on a different processor
- Backup file (EEPROM data is saved on the file EEPROM.IMA on the diskette)

EEPROM—You can write EEPROM data to primary processor EEPROMs or attached processor EEPROMs.

Backup File (saved image file)—You should write the backup file (EEPROM.IMA) on the diskette only when you are sure that EEPROM data is correct because the EEPROM Utility overwrites the previous version of the backup file. After you change EEPROM data by using the EEPROM Utility, make sure that the VAX 8200/8300 system runs properly with these changes. When you are satisfied, use the EEPROM Utility to write the backup file with this correct EEPROM data. You should follow this procedure to ensure that the backup file contains correct data.

You can later use the backup file to restore data if you discover an error in the EEPROMs.

Do not change or examine EEPROM.IMA without using the EEPROM Utility; changing this file could prevent the EEPROM Utility from operating. If you copy the EEPROM Utility program to another diskette, the file EEPROM.IMA must remain contiguous.

End of Pass Section Dialog

The dialog in the End of Pass section is as follows:

```
Want to make another pass Y/N {N} ?
```

Respond YES to return to the General section. If you have modified data, you should review the changes by making another pass of the dialog.

Respond NO if you want to write changes to an EEPROM or file, or to abort the session.

```
Want to abort this session Y/N {N} ?
```

Respond YES to exit the EEPROM Utility if you have only examined data. The contents of the buffer are deleted.

Respond NO if you want to save your changes on the diskette or write them into an EEPROM.

```
Want to write buffer to image file EEPROM.IMA on CSA1 Y/N {N} ?
```

Respond YES to write a backup file of EEPROM data.

Respond NO if you want to write your changes into an EEPROM.

```
Want to write changes back to the same EEPROM Y/N {N} ?
```

Respond YES to write your changes into the EEPROMs from which you loaded the buffer. This question appears only if you loaded data from a processor. To write data into the EEPROMs on the primary processor, the lower key switch on the control panel must be in the Update position. No specific switch position is required for writing data to the EEPROMs of an attached processor.

Respond NO if you want to write your changes into another EEPROM.

```
Want to select destination node for outputting changes Y/N {N} ?
```

Respond YES if you want to write your changes into EEPROMs other than those in the previous question. The lower key switch does not provide protection for an EEPROM on an attached processor. Therefore, the switch does not need to be in the Update position for the EEPROM Utility to write data into an EEPROM on an attached processor.

The EEPROM Utility prints a table like the following that shows which VAXBI node numbers contain processors:

```
NODE NUMBER              KA820 PROCESSOR TYPE
-------------------------------------------------
02                       KA820-PRIMARY   2 EEPROMs
0F                       KA820-ATTACHED  2 EEPROMs

Enter destination node number in Hex: {02} :
```

3.4 RUNNING THE EEPROM UTILITY

The EEPROM Utility runs under software called the VAX Diagnostic Supervisor (VDS). To run the EEPROM Utility, you must first boot VDS; then you can begin the dialog. When running, the EEPROM Utility has exclusive use of the system.

To run the EEPROM Utility, follow these steps:

1. If you want to make a backup file of EEPROM data, remove the write-protect tab from the VAX 8200/8300 Util Prog Flp diskette (EEPROM Utility diskette).

   Otherwise, leave the write-protect tab on the diskette.

2. Insert the EEPROM Utility diskette into console drive 1, as shown in Appendix D.
3. Turn the lower key switch to the Halt position. Do not turn the switch to the Update position until you are ready to write data into an EEPROM. The EEPROM Utility instructs you when to turn the switch to the Update position.
4. Turn the upper key switch to the Enable position.
5. Type CTRL/P at the console terminal to enter console mode. The console mode prompt >>> appears.
6. Type B CSA1 in response to the >>> prompt.

After approximately 30 seconds, the following display appears:

```
                    VAX DIAGNOSTIC SOFTWARE
                          PROPERTY OF
                 DIGITAL EQUIPMENT CORPORATION

              ***CONFIDENTIAL AND PROPRIETARY***

Use Authorized Only Pursuant to a Valid Right-to-Use License
Copyright, Digital Equipment Corporation, 1985. All Rights Reserved.

DIAGNOSTIC SUPERVISOR. ZZ-EBSAA- 9.0-389 29-JUN-1986 00:00:03
DS>
```

DS> is the VAX Diagnostic Supervisor prompt. You can obtain help about running the EEPROM Utility by typing HELP EBUCA in response to the DS> prompt.

7. Type @EBUCA in response to the DS> prompt. After approximately 20 seconds, you can begin the EEPROM Utility dialog. The following display appears:

```
Program: EBUCA - VAX 8200/8300 EEPROM UTILITY AND VAXBI CONFIGURATOR,
revision 1.2, 1 test, at 00:00:23:27.
Testing: KA0

Do you want help Y/N (N) ?
```

8. Answer dialog questions to examine or change EEPROM data. See Section 3.3 about how to use the EEPROM Utility.
9. Type EXIT in response to the prompt DS> to exit to console mode.

3.5 EXAMPLES OF EXAMINING AND CHANGING DATA

The following sections provide examples of data most commonly examined or changed. Other data rarely needs to be examined or changed.

3.5.1 Examining the VAXBI Configuration and EEPROM Summary (General Section)

The VAXBI configuration shows what VAXBI option is on a particular VAXBI node. If self-test fails (see Chapter 2), you should examine this configuration to determine what option might be faulty.

A summary of important EEPROM parameters appears immediately after the VAXBI configuration.

The following example shows how to examine both data summaries and how to exit the EEPROM Utility without making changes.

```
                              ** EBUCA STARTUP **

Do you want help Y/N {N} ? NO

Source for loading EEPROM work buffer is the primary processor
Want to change this Y/N {N} ?

EE> Are you interested in the general section Y/N {N} ? YES

Want to display VAXBI configuration matrix Y/N {N} ? YES

---------------------------------------------------------
NODE   REVISION CODE   DEVICE CODE & TYPE
---------------------------------------------------------
00        0216           0102      DWBUA
02        2019           0105      KA820 CPU
06        0513           010E      KDB50
0B        0000           0001      MS820 MEMORY
0D        0000           0001      MS820 MEMORY
0E        0000           010B      CIBCI
0F        201B           0105      KA820 CPU

Want to display a summary of EEPROM contents (From Work Buffer) Y/N {N} ? YES

---------------------------------------------------------
EEPROM SUMMARY
---------------------------------------------------------
KA820 serial number is NI5520009
CPU REV: 01
PATCH REV: 1
Console default BAUD rate is 1200
VAXBI self-test timeout in seconds is 10
The VAXBI node number of the logical console is 02
F CHIP is enabled
RCX50 self-test is enabled
CACHE is enabled
The default boot device is DU40
Boot Command Parser - Version 100
Boot Device Type CS - Version 101
Boot Device Type DU - Version 101
EEPROM sanity check is good
---------------------------------------------------------

KA820 serial number is NI5520009
Want to change this Y/N {N} ? >

                              * END OF PASS *
Want to make another pass Y/N {N} ?
Want to abort this session Y/N {N} ? YES

...End of run, 0 errors detected, pass count is 1,
   time is 29-JUN-1986 00:03:17.95
```

The preceding displays show examples of the VAXBI configuration and the EEPROM summary.

The revision code, shown in the VAXBI configuration, identifies the option's revision level and is implementation dependent. The device code and type indicate the VAXBI option on a particular VAXBI node.

Version numbers next to boot device types, shown in the EEPROM summary, are version numbers of boot code for the device type.

The EEPROM sanity check is a test verifying that certain EEPROM locations contain correct values. If the values are incorrect, the EEPROM Utility asks if you want to correct them in the General section. Respond YES, and the EEPROM Utility automatically corrects these values.

3.5.2 Changing the Default Console Baud Rate (Console Section)

The following example shows how to change the default console baud rate and how to write the change into the EEPROMs.

When DIGITAL Field Service installs the VAX 8200/8300 system, the default console baud rate is 1200.

```
                         ** EBUCA STARTUP **

Do you want help Y/N {N} ? NO

Source for loading EEPROM work buffer is the primary processor
Want to change this Y/N {N} ?

EE> Are you interested in the general section Y/N {N} ?

EE> Are you interested in the console section Y/N {N} ? YES

Console default BAUD rate is 1200
Want to change this Y/N {N} ? YES

BAUD rate: 150
Want this Y/N {N} ?
BAUD rate: 300
Want this Y/N {N} ?
BAUD rate: 600
Want this Y/N {N} ?
BAUD rate: 1200
Want this Y/N {N} ?
BAUD rate: 2400
Want this Y/N {N} ? YES
```

```
The VAXBI node number of the logical console is 02
Want to change this Y/N {N} ? ^

Console default BAUD rate is 2400
Want to change this Y/N {N} ? NO

The VAXBI node number of the logical console is 02
Want to change this Y/N {N} ? >

                                   * END OF PASS *
Want to make another pass Y/N {N} ?
Want to abort this session Y/N {N} ?
Want to write buffer to image file EEPROM.IMA on CSA1 Y/N {N} ?

Want to write changes back to the same EEPROM Y/N {N} ? YES
Put control panel switch in update position
Are you ready to continue Y/N {N} ? YES

Update could take several minutes-DO NOT ABORT DURING UPDATE
** Operation was SUCCESSFUL **
Want to exit the utility Y/N {N} ? YES

...End of run, 0 errors detected, pass count is 1,
   time is 29-JUN-1986 00:03:17.95
```

The question about baud rate repeats for each available baud rate, from lowest to highest. The baud rates are 150, 300, 600, 1200, 2400, 4800, 9600, and 19200. Respond YES to change the default console baud rate to the value listed in the question. Respond NO if you want a higher value.

After changing the baud rate, you can type a caret to return to the previous question and verify that the baud rate was changed. Once you have verified the new baud rate, type a right arrow > to get to the End of Pass section.

3.5.3 Changing the Default Boot Device (Boot Section)

The following example shows how to change the default boot device and how to write the change into the EEPROM.

When DIGITAL Field Service installs the VAX 8200/8300 system, the default boot device is DU40.

```
                              ** EBUCA STARTUP **

Do you want help Y/N {N} ?

Source for loading EEPROM work buffer is the primary processor
Want to change this Y/N {N} ?

EE> Are you interested in the general section Y/N {N}?
```

```
EE> Are you interested in the console section Y/N {N} ?

EE> Are you interested in the boot code section Y/N {N} ? YES

Want to display boot summary Y/N {N} ? YES

--------------------------------------------------------
BOOT SUMMARY
--------------------------------------------------------

The default boot device is DU40

Boot command parser - Version 100
Start Add 20090104
Last Add 20090169

Boot Device Type DU - Version 101
Start Add 20090214
Last Add 20090313

Boot Device Type CS - Version 101
Start Add 20090170
Last Add 20090213

--------------------------------------------------------------

The default boot device is DU40
Want a different default boot device type Y/N {N} ?

The default boot device is DU40
Want to modify the default boot device node and unit number Y/N {N} ? YES
Enter node and unit number of form nu: [40] : 42

Want to verify boot code against files on disk Y/N {N} ? ^

The default boot device is DU42
Want to modify the default boot device node and unit number Y/N {N} ? >

                                 * END OF PASS *
Want to make another pass Y/N {N} ?
Want to abort this session Y/N {N} ?

Want to write changes back to the same EEPROM Y/N {N} ? YES
Put front panel switch in update position
Are you ready to continue Y/N {N} ? YES

Update could take several minutes-DO NOT ABORT DURING UPDATE
** Operation was SUCCESSFUL **
Want to exit the utility Y/N {N} ? YES

...End of run, 0 errors detected, pass count is 1,
   time is 29-JUN-1986 00:03:17.95
```

The boot summary shows the available boot device types and the location of their boot code. Boot code, programs necessary for each boot device type, allows you to boot software from a device. The summary also lists version numbers of boot code.

The boot command parser coordinates the booting of your operating system and other software. When you boot software, the boot command parser uses your command to find the specific (or default) boot device.

In the preceding example, only the unit number of the default boot device was changed. You can also change the entire boot device designation. Boot devices are in the form ddnu, where dd is the device type, n is the VAXBI node number, and u is the unit number of the controller. For example, DU90 is a disk on node 9, unit 0. (Table 2–1 lists boot device types and their corresponding devices.)

3.5.4 Loading a New Set of Microcode Patches (Microcode Patch Section)

DIGITAL may issue an updated set of patches on another VAX 8200/8300 Util Prog Flp diskette. If you receive a new set of patches, you need to load the set into the EEPROM.

A patch replaces a section of the base microcode. The base microcode, permanently fixed in read-only memory, is a program that the VAX 8200/8300 computer uses to perform basic hardware functions. DIGITAL uses patches to modify and update the microcode. Each patch corresponds to a section of the base microcode. As it executes the microcode, the processor checks each word for a corresponding patch. If one is found, the processor executes the patch instead of the base microcode.

When you turn on the system or press the Restart button or use the console T command, the patches are loaded from the EEPROM into the random-access memory of the control store. The control store is the combination of microcode in read-only memory and patches in random-access memory.

Patches are stored in the EEPROM. Copies of the patches are also stored on the EEPROM Utility diskette in the file KApppp.PAT, where pppp is the patch revision number. When you load new patches, you load them from this file into the EEPROM by using the EEPROM Utility.

A new set of patches is issued on a new EEPROM Utility diskette. To load a new set of patches into the EEPROM, insert the new diskette into console drive 1 and follow these steps:

1. Turn the upper key switch to the Enable position.
2. Turn the lower key switch to the Halt position.
3. Type CTRL/P at the console terminal to enter console mode. The console mode prompt >>> appears.
4. Type B CSA1 in response to the >>> prompt.

   After approximately 30 seconds, the following display appears:

```
                    VAX DIAGNOSTIC SOFTWARE
                          PROPERTY OF
                 DIGITAL EQUIPMENT CORPORATION

              ***CONFIDENTIAL AND PROPRIETARY***

Use Authorized Only Pursuant to a Valid Right-to-Use License
Copyright, Digital Equipment Corporation, 1985. All Rights Reserved.

DIAGNOSTIC SUPERVISOR. ZZ-EBSAA- 9.0-389 29-JUN-1986 00:00:03
DS>
```

   DS> is the VAX Diagnostic Supervisor prompt.

5. Type DIR in response to the DS> prompt. A list of files, like the following example, is printed:

```
CSBOOKA.SYS;2    DUBOOKA.SYS;5    EBSAA.EXE;325    EBUCA.COM;5
EBUCA.EXE;1      EBUCA.HLP;7      EEPROM.IMA;1     KA0021.PAT;1
KA8BOO.SYS;3     KAINIT.SYS;5     KAINIT2.SYS;1
```

   The file KApppp.PAT, where pppp is the patch revision number (decimal), contains the patches. Record the file's name; this file contains the new set of patches.

6. Type @EBUCA in response to the DS> prompt to start the EEPROM Utility. Find the microcode patch section:

```
                          ** EBUCA STARTUP **

Do you want help Y/N {N} ?

Source for loading EEPROM work buffer is the primary processor
Want to change this Y/N {N} ?

EE> Are you interested in the general section Y/N {N} ?

EE> Are you interested in the console section Y/N {N} ?

EE> Are you interested in the boot section Y/N {N} ?

EE> Are you interested in the microcode patch section Y/N {N} ? YES

Want to examine/modify CPU and Patch revision information Y/N {N} ?

Want to compare patches in buffer against patch file on disk Y/N {N} ?

Want to load a new set of microcode patches Y/N {N} ? YES
Enter file specification: KA0021.PAT

Want to list patches Y/N {N} ? >

                           ** END OF PASS **
Want to make another pass Y/N {N} ?
Want to abort this session Y/N {N} ?
Want to write buffer image file EEPROM.IMA on CSA1 Y/N {N} ?
Want to write changes back to the same EEPROM Y/N {N} ? YES
Put control panel switch in update position
Are you ready to continue Y/N {N} ? YES

Update could take several minutes-DO NOT ABORT DURING UPDATE
                           * Operation was SUCCESSFUL **
Want to exit the utility Y/N {N} ? YES

...End of run, 0 errors detected, pass count is 1,
   time is 29-JUN-1986 00:15:24.19
```

As the patches are loaded, the EEPROM Utility uses the processor revision number to verify that the patches are compatible with the processor. The patch file KApppp.PAT contains a list of processor revision numbers that are compatible with the set of patches. If the patches are incompatible, they are not loaded, and an error message appears on the terminal. The patch revision number, stored in the EEPROM, is automatically updated when a new set of patches is loaded.

7. Type EXIT in response to the DS> prompt to return to console mode.
8. Press the Restart button or type the T command to load the patches from the EEPROM into the control store.

3.5.5 Writing a Backup File of EEPROM Data (End of Pass Section)

After you have made changes to EEPROM data and are satisfied that the system runs properly with these changes, you should use the EEPROM Utility to write a backup file of correct EEPROM data (see Section 3.3.6). The following example shows how to write a backup file.

```
                              ** EBUCA STARTUP **

Do you want help Y/N {N} ?

Source for loading EEPROM work buffer is the primary processor
Want to change this Y/N {N} ?

EE> Are you interested in the general section Y/N {N} ? >

                                   END OF PASS *
Want to make another pass Y/N {N} ?
Want to abort this session Y/N {N} ?
Want to write buffer to image file EEPROM.IMA on CSA1 Y/N {N} ? YES
** Operation was SUCCESSFUL **
Want to exit the utility Y/N {N} ? YES

...End of run, 0 errors detected, pass count is 1,
   time is 29-JUN-1986 00:15:24.19
```

3.5.6 Restoring EEPROM Data from Backup File (Startup Section)

If you discover an error in EEPROM data, you can restore correct data from the backup file. Then you can write this data into an EEPROM. The following example shows how to restore EEPROM data.

```
                              ** EBUCA STARTUP **

Do you want help Y/N {N} ?

Source for loading EEPROM work buffer is the primary processor
Want to change this Y/N {N} ? YES

Want to load initialization file KAINITx.SYS Y/N {N} ?

Want to load buffer from EEPROM saved image file on CSA1 Y/N {N} ? YES
EE> Are you interested in the general section Y/N {N} ? >

                                 * END OF PASS *
Want to make another pass Y/N {N} ?
Want to abort this session Y/N {N} ?
```

```
Want to select destination node for outputting changes Y/N {N} ? YES

NODE NUMBER              KA820 PROCESSOR TYPE
-----------------------------------------------
02                       KA820-PRIMARY   2 EEPROMs
0F                       KA820-ATTACHED  2 EEPROMs

Enter destination node number in Hex: [02] : 02
Put control panel switch in Update position

Are you ready to continue Y/N {N} ? YES

Update could take several minutes-DO NOT ABORT DURING UPDATE
** Operation was SUCCESSFUL **
Want to exit the utility Y/N {N} ? YES

...End of run, 0 errors detected, pass count is 1,
   time is 29-JUN-1986 00:03:17.95
```

3.6 CHECKSUMS

The EEPROM Utility uses checksums to verify that certain data is correct. If this data is not correct, the EEPROM Utility prints an error message.

A checksum, generated from an algorithm performed on data, is a constant used to verify that the data is correct. Boot code, the set of microcode patches, and the control store each have a unique checksum. These three checksums are stored in the EEPROMs.

When you turn on the system, the processor recalculates these three checksums and compares them to the corresponding checksums stored in the EEPROMs. If a comparison is not an exact match, the processor self-test fails.

The EEPROM Utility also recalculates the three checksums and compares them to the stored values in the EEPROM data in the buffer. Checksums for the patches and boot code are recalculated from the patches and boot code in the buffer; the control store checksum is recalculated from the control store on the primary processor. A comparison error should almost never occur. If one does, the EEPROM Utility asks if you want the checksum stored in the EEPROM Utility buffer to become the correct value (a new checksum). Do not change the checksum, however, unless you know what caused the error. In most cases, if the checksum comparison fails, you should load new EEPROM data.

Identifying System Hardware 4

The VAX 8200/8300 operations kit includes the diskette VAX 8200/8300 Diag Super + Auto. This diskette contains a program called the autosizer, which lets you print a list of all system devices. Before you can run the autosizer, you must boot the VAX Diagnostic Supervisor from the same diskette.

You can boot the VAX Diagnostic Supervisor only when the VAX 8200/8300 is in console mode. When running, the autosizer has exclusive use of the system. To print the list of devices, follow these steps:

1. Insert the VAX 8200/8300 Diag Super + Auto diskette into console drive 1, as shown in Appendix D.
2. Turn the upper key switch to the Enable position.
3. Keep the lower key switch in the Auto Start position.
4. Type CTRL/P at the console terminal to enter console mode. The console mode prompt >>> appears.
5. Type T/M RET in response to the console mode prompt to boot the VAX Diagnostic Supervisor. The following display appears:

```
                    VAX DIAGNOSTIC SOFTWARE
                          PROPERTY OF
                 DIGITAL EQUIPMENT CORPORATION

               ***CONFIDENTIAL AND PROPRIETARY***

Use Authorized Only Pursuant to a Valid Right-to-Use License
Copyright, Digital Equipment Corporation, 1985. All Rights Reserved.
DIAGNOSTIC SUPERVISOR. ZZ-EBSAA- 9.0-389 29-JUN-1986 11:32:56
DS>
```

6. Type RUN EVSBA (RET) in response to the prompt DS>. EVSBA is the autosizer program. The following display appears:

```
...Program: EVSBA - AUTOSIZER LEVEL 3, revision 2.1001, 3 tests, at
   11:33:10.17

...End of run, 0 errors detected, pass count is 1,
   time is 29-JUN-1986 11:34:23.46
DS>
```

7. Type SHOW DEVICE/BRIEF in response to the prompt DS>. The /BRIEF qualifier leaves out information relevant only to diagnostics.

   A list of system devices, like the following example, is printed, where the left column shows the devices' generic names and the right column shows the devices' specific names.

```
_CSA0                         console
_DW0                          DWBUA
_DUA                          UDA50
_DUA0                         RA81
_KA0                          KA820
_MSA0                         TU80
_DYA                          RX211
_DYA0                         RX02
_DYA1                         RX02
_DJA1                         RA60
_DJA2                         RA60
_XEA0                         UNA11
_XMA0                         DMR11
_DAA                          LESI
_XGA0                         DMF32S
_TXA                          DMF32A
_LCA                          DMF32P
_TXB                          DHU11
_DUB                          KDB50
```

8. Type EXIT in response to the prompt DS>. The system returns to console mode.

Multiprocessing with the VAX 8300 5

The VAX 8300 is a multiprocessing system designed for compute-intensive applications. The VAX 8300 uses two processors. The processor in the rightmost VAXBI slot is the primary processor. The second processor is called the attached processor. Both processors can run VMS processes; the primary processor runs all VMS system service calls.

This chapter describes how to boot VMS with multiprocessing, how to correct EEPROM data if the attached processor fails self-test, how to disable the attached processor, and how to replace a broken primary processor with the attached processor. This chapter also explains how to use console commands with the attached processor.

5.1 BOOTING VMS WITH MULTIPROCESSING

After you boot VMS by following one of the procedures described in Section 2.2 or 2.3, you need to type two additional commands to start the second processor (attached processor). The first command defines a logical name so that VMS can use the multiprocessing software. The second command starts the attached processor. You can add these commands to the VMS command file SYSTARTUP.COM (see the *Guide to VAX/VMS System Management and Daily Operations*).

To boot VMS with multiprocessing, follow these steps:

1. Follow one of the booting procedures described in Chapter 2.
2. After you log in to the SYSTEM account, type the following commands:

```
$ DEFINE/SYSTEM/EXEC MP MP8SS

$ START/CPU
```

5.2 MULTIPROCESSING GUIDELINES

To function properly, the attached processor must contain the following data in its EEPROM:

- The RCX50 self-test is disabled.
- The node number of the logical console is 02 (the primary processor's node number).

If the attached processor's RCX50 self-test is enabled, the attached processor will always fail self-test. If the attached processor's logical console node number is not 02, you cannot communicate with the attached processor with console commands and the attached processor may not function. You can correct the RCX50 self-test by using the console D command (Section 5.2.1) and the logical console node number by moving the attached processor and running the EEPROM Utility (Section 5.2.2).

You should follow some special guidelines when using multiprocessing:

- If a minus sign appears next to the attached processor's node number during self-test (see Section 2.8.3), the attached processor has failed self-test; you should verify that attached processor's RCX50 self-test is disabled by using the console E command. If the RCX50 self-test is enabled, the attached processor will always fail self-test.
- If you cannot communicate with an attached processor that failed self-test, a qualified service technician can place the attached processor in the primary processor's VAXBI slot, then run the EEPROM Utility to correct EEPROM data.
- If the attached processor is broken, you can disable it with the VMS STOP command.
- If the primary processor is broken, you can temporarily replace it with the attached processor until the primary processor is repaired.
- When you change the attached processor's EEPROM data by using the EEPROM Utility, you do not need to put the lower key switch in the Update position. Because the switch provides no protection for the attached processor's EEPROM data, you should be careful when making changes to the data; verify that your changes are correct before you write them into the attached processor's EEPROM. Note that if the attached processor is

placed in the primary processor's VAXBI slot (Section 5.2.4), the attached processor becomes the primary processor, and the switch must be in the Update position to modify EEPROM data.

- To provide a means for bypassing the primary processor, you should set the VMS TIMEPROMPTWAIT parameter to a small value, such as 60, by running the SYSGEN program under VMS (see Section 5.2.4.1).

5.2.1 Correcting EEPROM Data If an Attached Processor Fails Self-Test

If the attached processor fails self-test, you should verify that its EEPROM data is correct. The attached processor's RCX50 self-test must be disabled. If the attached processor's RCX50 self-test is enabled, the attached processor will fail the system self-test. The EEPROM contains the RCX50 self-test enable/disable information.

Because the EEPROM Utility cannot communicate with the EEPROM of an attached processor that failed self-test, you must use the console E command to examine the EEPROM data. You can disable the RCX50 self-test by using the console D command.

Suppose you see the following display after self-test completes:

```
#ABCDEFGHIJK.MN#
0 . 2 . 4 . . . . . . -B C D . .
00400000
?03

PC = 00000002
```

Node B, the attached processor, has failed self-test. You should examine the attached processor's EEPROM data, and if the RCX50 self-test is enabled, you should disable it. To disable the RCX50 self-test for an attached processor on node B, you would type the following console commands:

1. `>>> Z B`
   `PC = 00000002`

   The Z command logically connects the console to the attached processor on node B. Subsequent console commands are processed by the attached processor.

2. `>>>` (ESC) (CTRL/P)

   This command ensures that the attached processor is in console mode.

3. `>>> E/P 20080004`
```
      P     20080004      0401380B
```

This command examines the VAXBI address space to verify that you can communicate with the attached processor. The rightmost digit of the data contained at the address 20080004 should be the node number of the attached processor. In the previous example, the data 0401380B verifies the connection to the attached processor on node B.

If you cannot communicate with the attached processor, the rightmost digit will be a 2 (the primary processor).

NOTE
If you cannot communicate with an attached processor that has failed self-test, see Section 5.2.2.

4. `>>> E/E 3`
```
      P     20098176      08
```

This command examines the RCX50 self-test enable/disable data contained in the attached processor's EEPROM. The data 08 indicates that the RCX50 self-test is enabled. The data 18 would indicate that the RCX50 self-test is disabled.

5. `>>> D/E 3 18`

This command deposits the data 18 into the EEPROM to disable the RCX50 self-test. Be careful not to deposit any other value.

6. `>>>` [CTRL/P]

This command disconnects the console from the attached processor. Subsequent commands are processed by the primary processor.

After you have disabled the RCX50 self-test of the attached processor, run the system self-test again to make sure the attached processor passes.

5.2.2 Communicating with the Attached Processor If It Fails Self-Test

If the attached processor has failed self-test for a reason other than an enabled RCX50 self-test, you may not be able to communicate with the attached processor in its current VAXBI slot. In this case, a qualified service technician can place the attached processor in the primary processor's VAXBI slot and run the EEPROM Utility to correct EEPROM data.

5.2.3 Disabling the Attached Processor

If the attached processor is broken, you can disable it by using the VMS STOP command. After you disable the attached processor, the VAX 8300 runs as a single-processor system (the VAX 8200).

To disable the attached processor, type the command:

```
$ STOP/CPU
```

5.2.4 Running a System with a Broken Primary Processor

If the primary processor is broken, you can temporarily run the system by using the attached processor. Your system can then function as a single-processor system until the original primary processor is repaired. You can run the system in one of two ways: bypass the primary processor by booting VMS on the attached processor with the console Z command, or place the attached processor in the primary processor's VAXBI slot.

5.2.4.1 Bypassing the Primary Processor — You can use the console Z command to bypass a broken primary processor and boot VMS on the attached processor. For some failures of the primary processor, you may not be able to use this procedure. This procedure places the following restrictions on the system:

- You cannot use the console terminal to log in to VMS. In addition, information that usually appears on the console terminal is lost. For example, if VMS fails while running, machine check information is lost.
- You cannot boot VMS from an HSC disk. Booting from an HSC disk involves the use the RCX50 diskette drive that is directly connected to the primary processor (the rightmost VAXBI slot). However, you can boot VMS from a local disk.
- VMS cannot gain access to the watch chip to get the time of day. Instead, VMS prompts you for the time of day, but because you cannot use the console terminal, you cannot enter the data. Therefore, the TIMEPROMPTWAIT parameter should have been changed to a small value, such as 60, by running the VMS SYSGEN program. If this parameter is set to a small value, VMS waits briefly for you to enter the time of day and then uses the (approximate) time recorded on the system disk.

- The system cannot automatically restart or reboot VMS following a power failure.
- The Run light is not lit, and the Fault light remains lit.

To use the Z command to replace the primary processor, follow these steps:

1. Type Z n in response to the console mode prompt, where n is the VAXBI node number of the attached processor.
2. Type (ESC) (CTRL/P) to ensure that the attached processor is in console mode.
3. Type the B command to boot VMS. You will not see any VMS displays on the console terminal after you type the B command. You might see the error code ?4C, indicating an error in the primary processor. However, this error should not prevent the system from booting VMS.
4. Log in to VMS on another terminal.
5. Type STOP/CPU under VMS so that VMS does not consider the broken primary processor to be the attached processor.

5.2.4.2 Manually Replacing the Primary Processor — A qualified service technician can remove the primary processor and replace it temporarily with the attached processor. This procedure does not place any restrictions on the system.

5.2.5 Changing EEPROM Data of Different Processor Versions

Some processors (early versions) have one EEPROM. If your system contains a processor with one EEPROM and a processor with two EEPROMs (the normal version), the EEPROM Utility prints a warning before you load and write data. Specify from which processor you will load EEPROM data, or to which processor you will write EEPROM data. You cannot update EEPROM data of both versions in a single session with the EEPROM Utility.

The following display appears when you run the EEPROM Utility and the system contains two different versions of processors:

```
                              ** EBUCA STARTUP **

Do you want help Y/N {N} ?

%WARNING-System has CPU types with one and two EEPROMs.
```

```
For updates, specify the type of your target. For examining only, specify the
type of your source. Do not update both CPU types in a single session.

NODE NUMBER              KA820 PROCESSOR TYPE
-------------------------------------------------
02                       KA820-PRIMARY   1 EEPROM
0F                       KA820-ATTACHED  2 EEPROMs

Does desired CPU type have 1 EEPROM? YES

Source for loading EEPROM work buffer is the primary processor Want to change
this Y/N {N} ? NO

EE> Are you interested in the general section Y/N {N} ?
```

In the example, you examine or change EEPROM data of the primary processor. If you change the EEPROM data, you can write the changes only to the primary processor:

```
                                  * END OF PASS *
Want to make another pass Y/N {N} ?
Want to abort this session Y/N {N} ?
Want to write buffer to image file EEPROM.IMA on CSA1 Y/N {N} ?
Want to write changes back to same EEPROM Y/N {N} ? YES
Put control panel switch in update position
Are you ready to continue Y/N {N} ? YES

Update could take several minutes-DO NOT ABORT DURING UPDATE
** Operation was SUCCESSFUL **
Want to exit the utility Y/N {N} ? YES

...End of run, 0 errors detected, pass count is 1,
   time is 29-JUN-1986 00:03:17.95
```

You cannot load EEPROM data from one processor version and write the changed data to an EEPROM of another version. EEPROM data must be loaded and written to the same processor version.

There are two versions of the initialization file KAINITx.SYS. A processor with one EEPROM uses KAINIT.SYS; a processor with two EEPROMs uses KAINIT2.SYS. Do not write data from the initialization file for one processor version to a different version.

The backup file contains data you have previously written; before you load data from the backup file, you must be sure of the backup file's processor version. You cannot load data from a backup file created from one processor version and write that data to a different processor version.

5.3 USING CONSOLE COMMANDS WITH THE ATTACHED PROCESSOR

You can use console commands with the attached processor by first typing the Z command. The Z command logically connects the console to a specific VAXBI node and directs subsequent commands to that node. To use console commands with the attached processor, you need the attached processor's VAXBI node number. You can find the node number by running the EEPROM Utility and examining the VAXBI configuration (see Section 3.5.1).

To use console commands with the attached processor, follow these steps:

1. Turn the upper key switch to the Enable position.
2. Press CTRL/P on the console terminal. The system enters console mode, and the console mode prompt >>> appears.
3. Type the command:

   ```
   >>> Z n
   ```

   The letter n represents the attached processor's VAXBI node number.
4. Type ESC CTRL/P to ensure that the attached processor is in console mode.
5. Type console commands that you want processed by the attached processor.

5.3.1 Forwarding a CTRL/P

To forward a CTRL/P to the attached processor, type the following:

```
>>> ESC CTRL/P
```

The ESC key forwards a CTRL/P only after you have used the Z command.

5.3.2 Stopping the Forwarding of Console Commands

After you have used the Z command, you can stop the forwarding of console commands to the attached processor by typing the following:

```
>>> CTRL/P
```

All subsequent console commands are processed by the primary processor.

EEPROM Parameters A

Table A–1 lists the EEPROM data that you can change by using the EEPROM Utility. The parameters are shown in order of their appearance in the EEPROM Utility. Default values are those present in the initialization file (KAINITx.SYS).

Table A–1: User-Modifiable EEPROM Data

Data	Description
Processor serial number	Serial number of primary processor. No default.
RX50 dual-diskette drive self-test	Enable or disable. Default: enabled.
Self-test timeout	Maximum amount of time a VAXBI node can take to test itself. Default: 10 seconds.
F chip	Chip that enhances the floating-point performance of the processor. Enable or disable. Default: enabled.
Cache memory	Memory that speeds processing by storing common memory references. Enable or disable. Default: enabled.
Console baud rate	The rate of transmission for console data. Default: 1200 baud.
Logical console node number	VAXBI node number to which attached processors send console messages. Default (in hexadecimal): 02.
Default boot device	Default device for booting software. Default device type: DU40.
Boot code	Programs stored in the EEPROM that the processor uses to boot the operating system from a specific device.

(Continued on next page)

Table A–1: User-Modifiable EEPROM Data (Cont.)

Data	Description
Interrupt and polling (IP) address for DU boot device controller	UNIBUS address for DU boot device controller when two or more DU boot device controllers are on the same UNIBUS. Default (in octal): 772150.
Processor and microcode patch revision numbers	Version numbers of processors and sets of patches. No default.
Microcode patches	Additions or replacements for sections of microcode.
Ethernet load server address	Default: 000000002800.
Boot message	Default: 000C0260 4F4F4206 24454D54 42420021 42424242 42424242.
Number of bytes compared for boot message recognition	Default: 14.
First offset to boot message where comparison begins	Default: 6.
Second offset to boot message where comparison begins	Default: 26.

Halt Codes and Error Codes B

Halt codes appear on the console terminal when the processor halts (stops processing). Error codes appear when the processor is in console mode. Both types of codes are in the form ?nn, where nn is a 2-digit number in hexadecimal. Table B–1 shows the halt codes; Table B–2, the error codes.

Table B–1: Halt Codes

Halt Code	Meaning
?01	Self-test successfully completed.
?02	CTRL/P was typed at the console terminal when the upper key switch was set to Enable.
?03	Restart or reboot occurred after power failure or after the Restart button was pressed.
?04	The processor attempted to push a state onto the interrupt stack during an interrupt or exception and found that the interrupt stack was mapped as NO ACCESS or as NOT VALID.
?05	The processor attempted to report a machine check to the operating system, and a second machine check occurred.
?06	The processor executed a halt instruction while in kernel mode.
?07	A vector in the System Control Block (SCB) has bits 1 and 0 set, an invalid state.
?08	The VAX 8200/8300 has no user Writable Control Store (WCS), but software has tried to use it.
?0A	The processor executed a change-mode instruction when bit 26 of the Processor Status Longword (PSL) was set.
?0C	A hard memory error occurred while the processor was trying to read an exception or an interrupt vector.

Table B–2: Error Codes

Error Code	Meaning
?40	Self-test failed when the system was turned on or when the Restart button was pressed.
?41	The signal BI AC LO L was not deasserted within 10 seconds.
?42	A restart (warm start) or an attempt to boot failed because the bootstrap-in-progress flag was already set.
?43	The processor could not find a 64K-byte block of good memory while trying to boot the operating system.
?44	Unrecognized console command or boot device specification was typed.
?45	A memory reference was not allowed with a D, E, or S console command.
?46	The processor was unable to find the specified internal processor register of a D or E command.
?47	The address specified in the D/E or E/E command was outside the valid range of 0 to 17(hex), or the writing of data onto the EEPROM by the D command was not allowed.
?48	A checksum error occurred in the command or data during an X command.
?49	The console was unable to forward a character to a remote VAXBI node during a Z command.
?4A	An error occurred during the loading of the control store patches from the EEPROM.
?4B	A checksum error occurred during verification of a boot program in the EEPROM.
?4C	A hardware error occurred.

Setting Jumpers for Fast Self-Test C

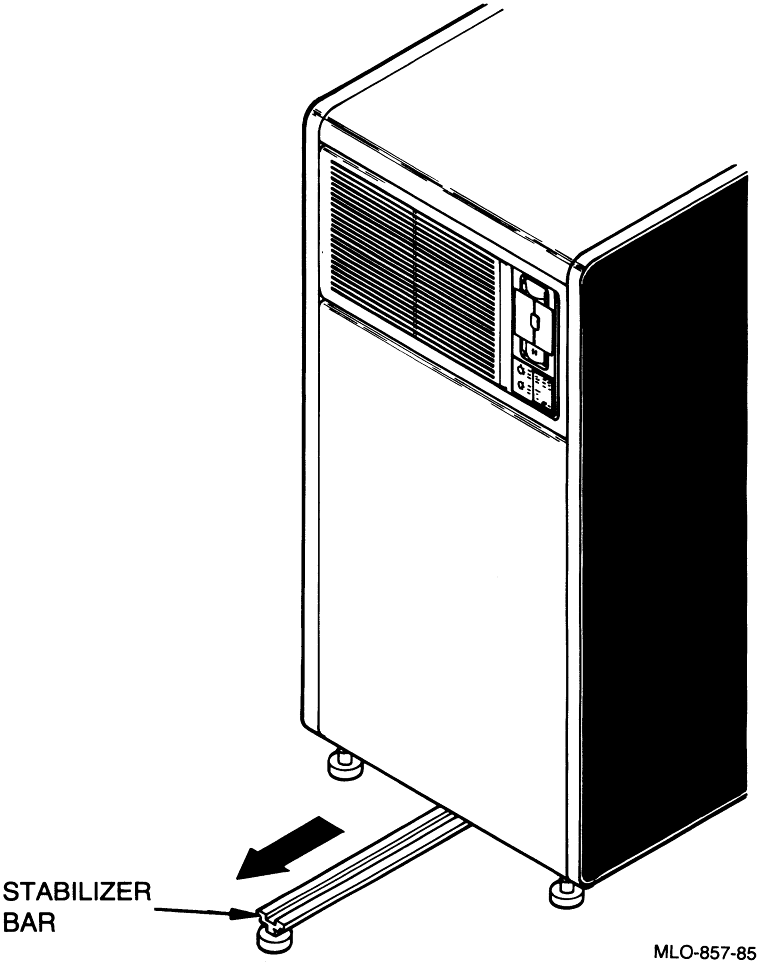
Use the fast self-test only if your system is running in a real-time environment and needs to recover quickly from a power failure (see Section 2.8.4). The following two procedures describe how to set the jumpers for the 12-slot system and the 24-slot system, respectively.

C.1 SETTING JUMPERS FOR A 12-SLOT SYSTEM

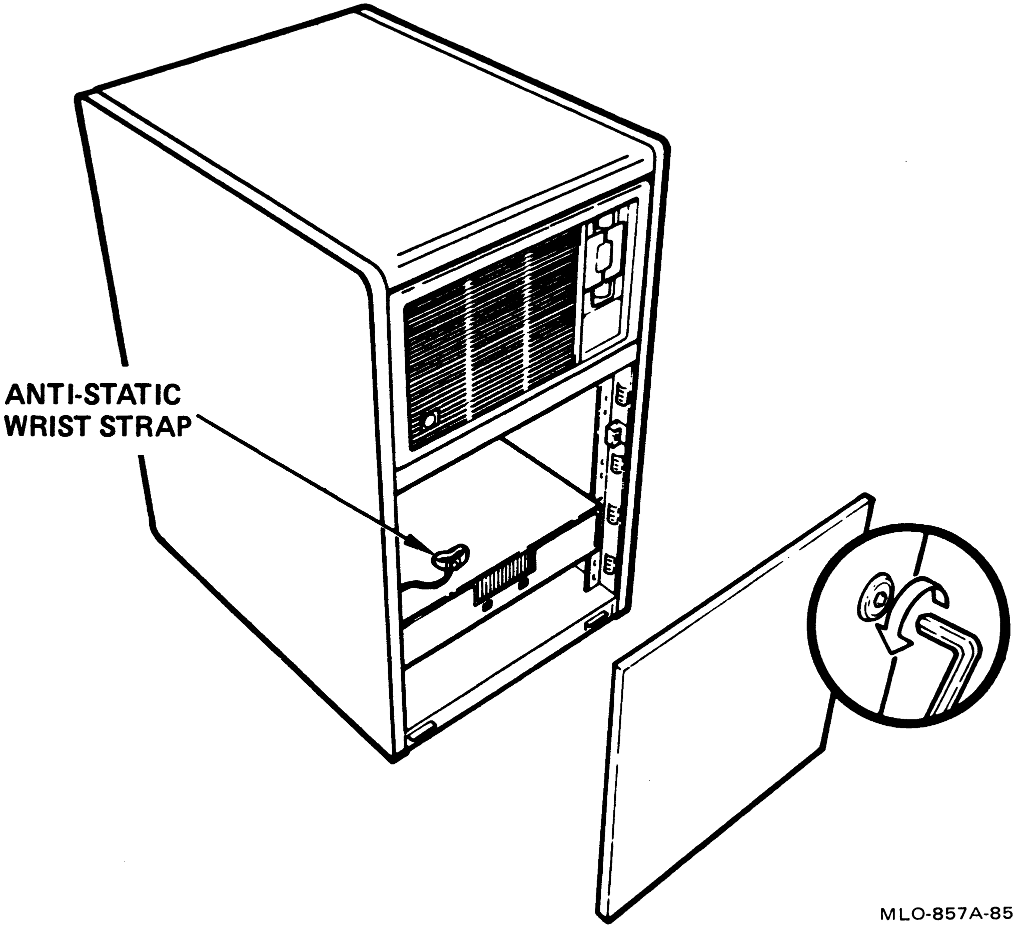
To set the jumpers for a 12-slot system, follow these steps:

1. Turn the upper key switch to the Off position.
2. Turn the main power switch to the Off position, and unplug the power cord from the electric outlet.

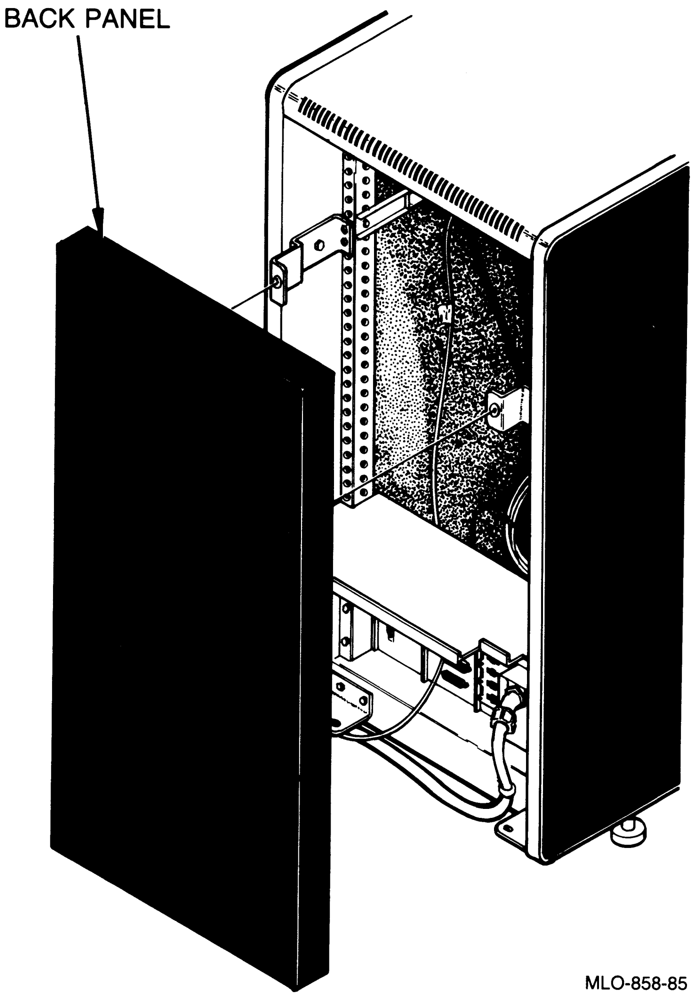
3. Pull out the stabilizer bar at the bottom of the main cabinet.



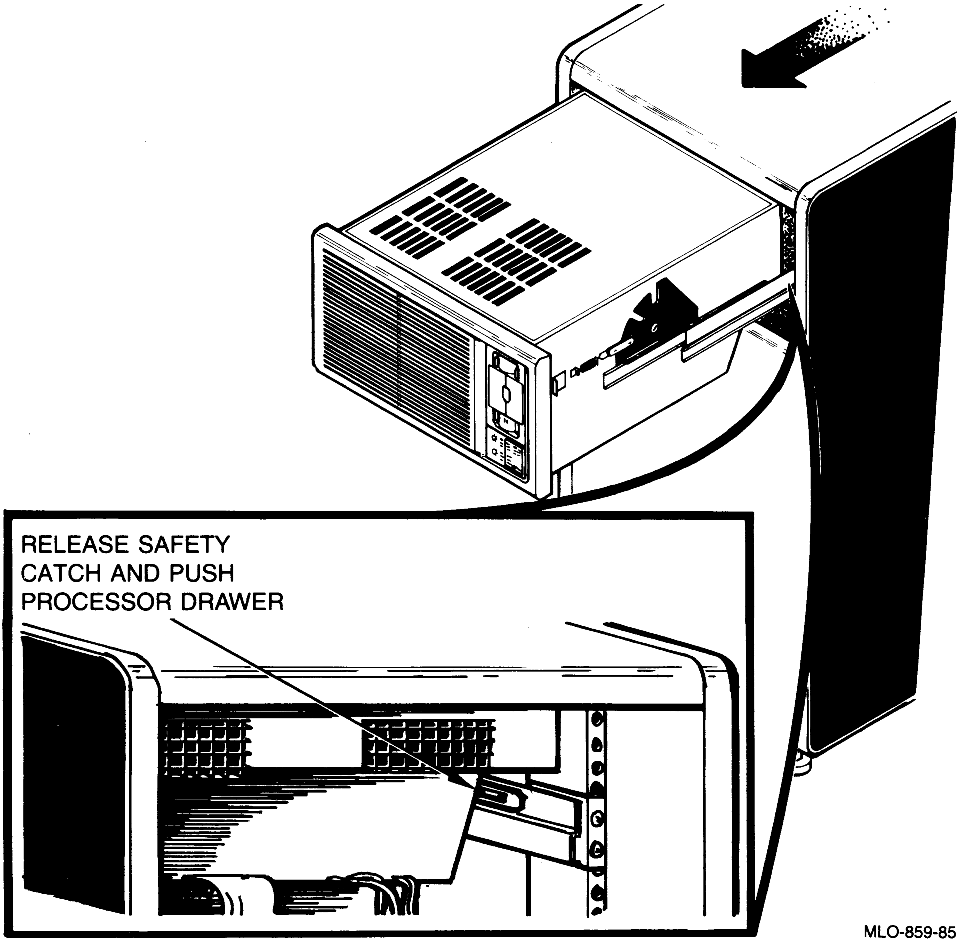
4. Remove the front panel with a 5/32-inch hex key to avoid damaging cables when rotating the processor drawer.



5. Remove the back panel with a 5/32-inch hex key.

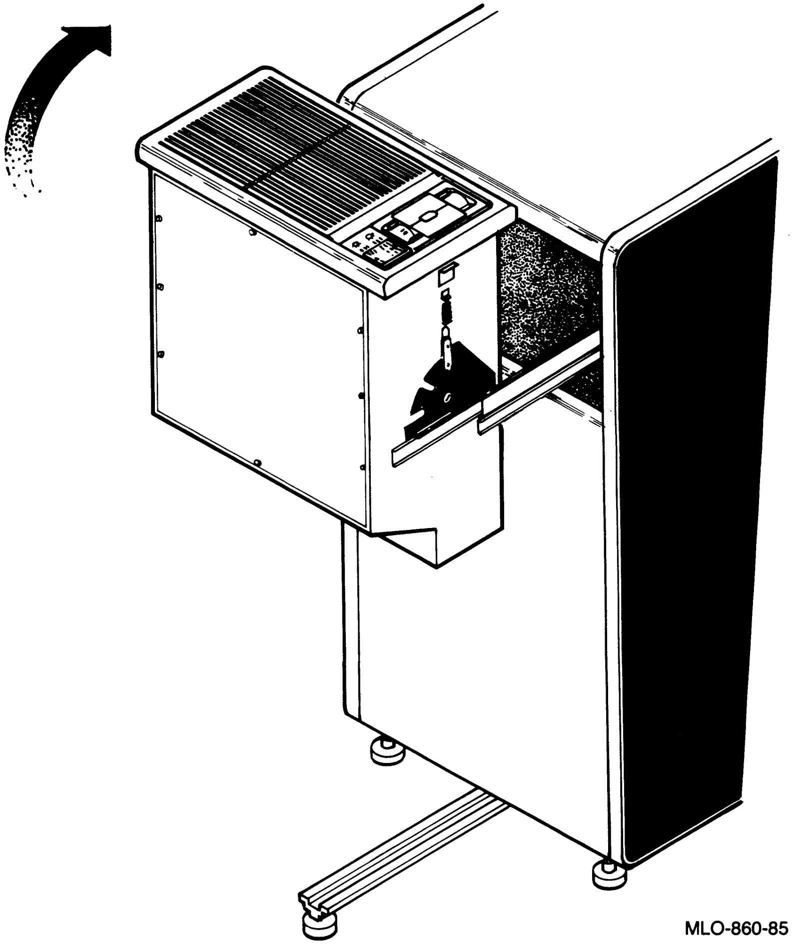


6. Wear the anti-static wrist strap, located at the bottom of the main cabinet. You should wear the wrist strap whenever you open the main cabinet to avoid damaging static-sensitive hardware.
7. Push the safety catch.



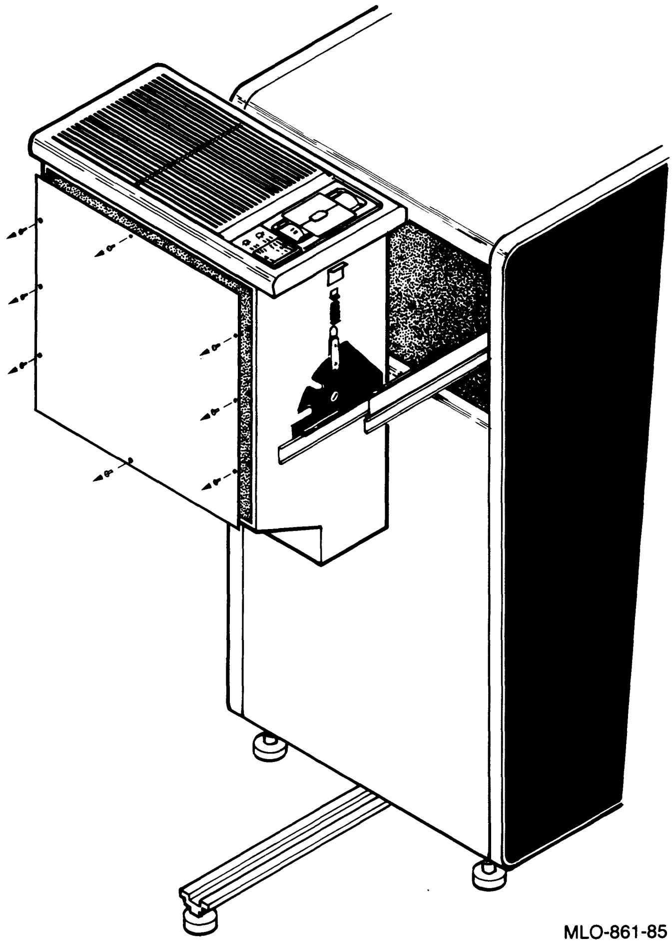
MLO-859-85

8. Push out the processor drawer from behind.
9. Pull latches and rotate the drawer upward to the vertical position.



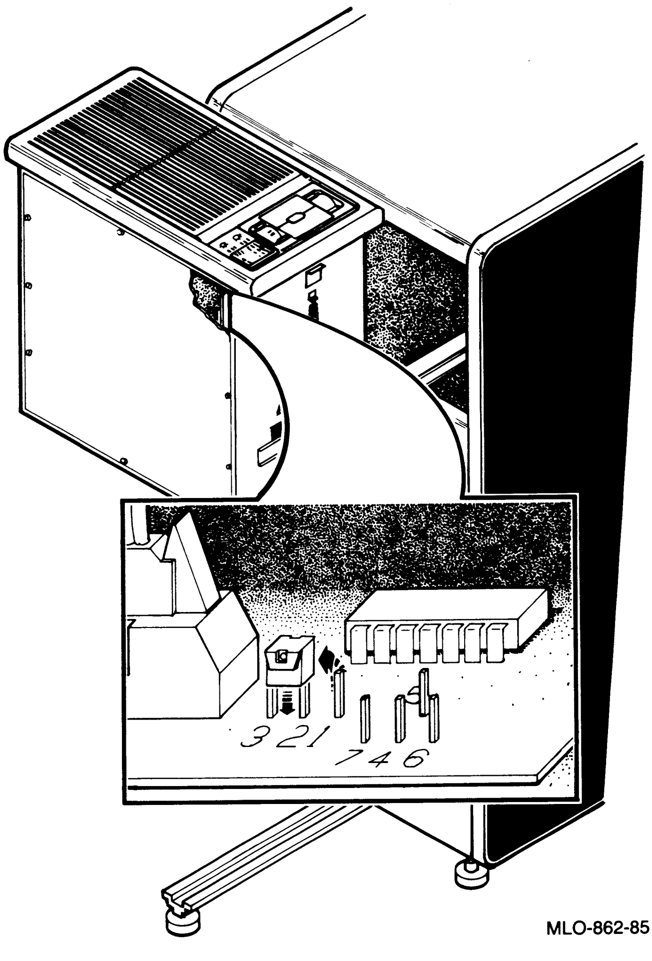
MLO-860-85

10. Remove the bottom panel of the processor drawer.



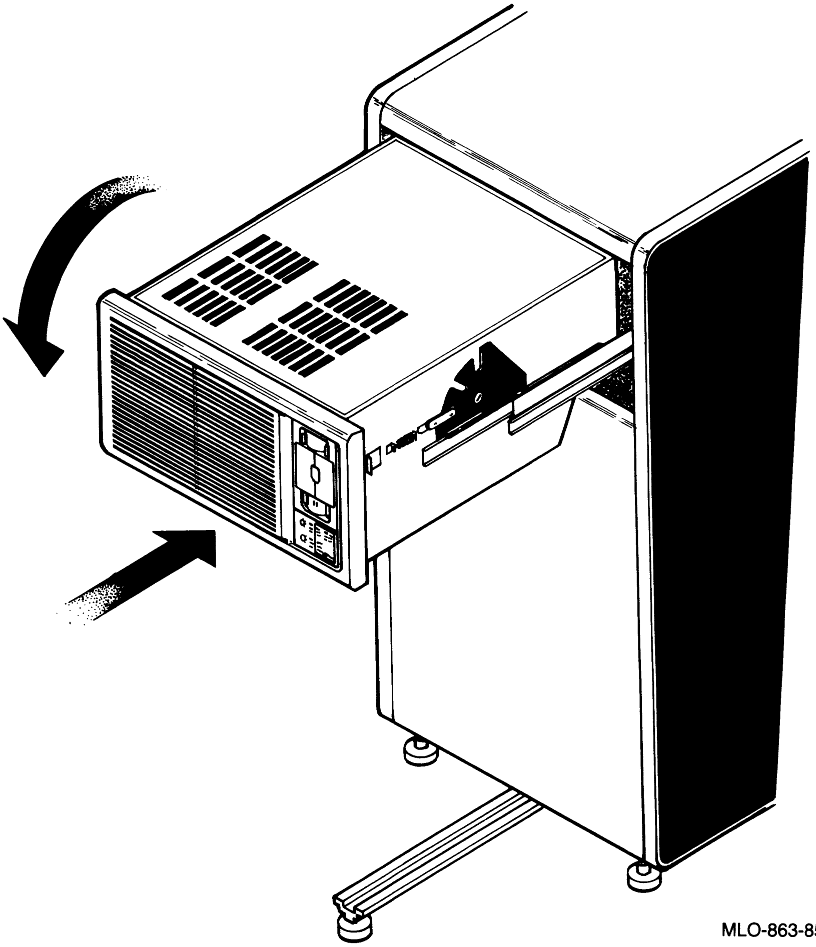
MLO-861-85

11. Move the blue jumper cap covering pins 1 and 2 so that it covers pins 2 and 3.



MLO-862-85

12. Replace the bottom panel, lower the drawer to the horizontal position, and push it back inside the main unit.



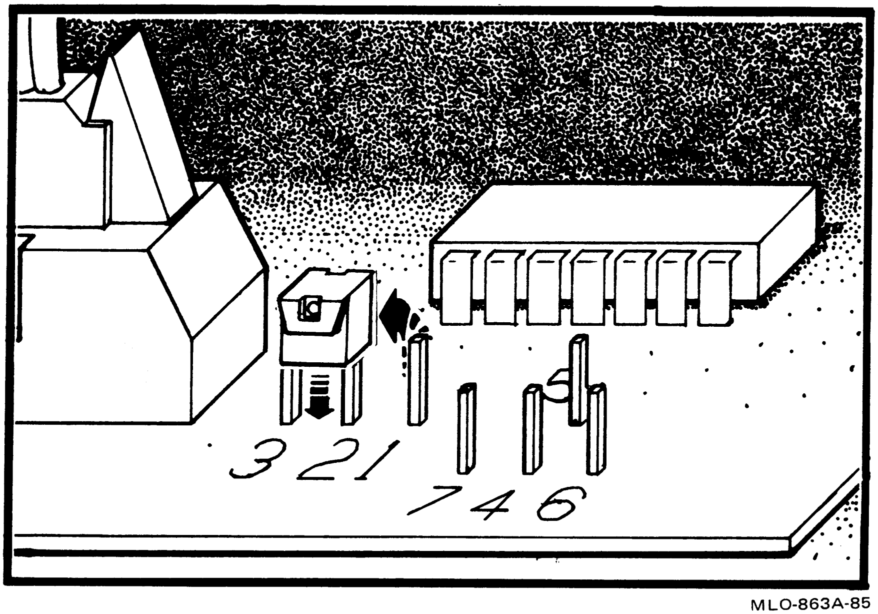
MLO-863-85

13. Replace the back cover. Push the stabilizer bar back inside.

C.2 SETTING JUMPERS FOR A 24-SLOT SYSTEM

To set the jumpers for a 24-slot system, follow these steps:

1. Turn the upper key switch to the Off position.
2. Turn the main power switch to the Off position and unplug the power cord from the electric outlet.
3. Remove the screws holding the top cover. Remove the cover by lifting its front end and sliding the cover toward the back of the cabinet.
4. Remove the cable from RCX50 controller board and slide the board toward the back of the cabinet. This reveals the PCM board, where the jumpers are located.
5. Move the blue jumper cap covering pins 1 and 2 so that it covers pins 2 and 3.



MLO-863A-85

6. Connect the cable to the RCX50 controller board and slide the board back in place.
7. Replace the cover and the screws.

Using the RX50 Dual-Diskette Drive D

The RX50 dual-diskette drive is on the main cabinet, next to the control panel. The drive can hold two diskettes. Two drives with locking doors let you load and unload diskettes. For a 12-slot system, the left drive is drive 1 (CSA1), and the right drive is drive 2 (CSA2). For a 24-slot system, the upper drive is drive 1 (CSA1), and the lower drive is drive 2 (CSA1).

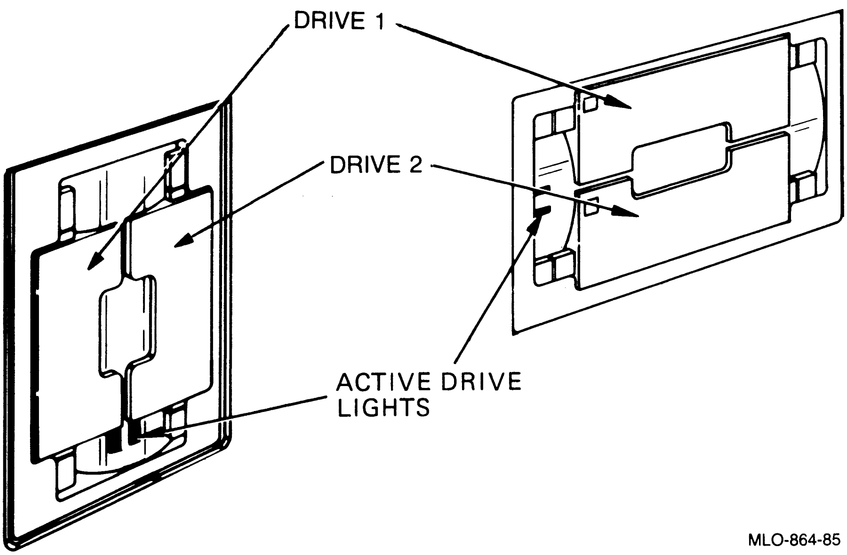


Figure D–1: The RX50 Dual-Diskette Drive

D.1 DISKETTE HANDLING AND STORAGE

The diskette is permanently enclosed in a plastic cover. When not in use, the diskette should be kept in its protective envelope. Diskettes must be formatted according to standards established by DIGITAL.

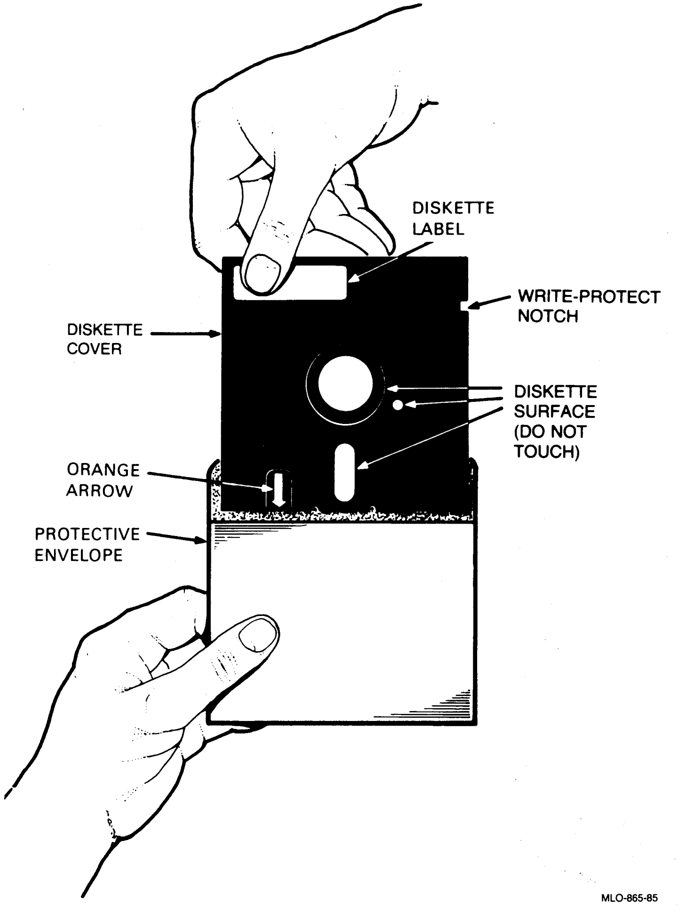


Figure D–2: The Diskette

Incorrect handling and storage of diskettes can result in damage to a diskette (and/or the RX50 recording head) and in the eventual loss of data. The following points should be observed:

- Always keep a diskette in the protective paper envelope when the diskette is not in use.
- Do not fold or bend the plastic cover.
- Do not touch the recording surfaces of a diskette. When handling a diskette out of its envelope, touch only the diskette's top (label area).
- Store diskettes upright (with the labels at the top) in proper storage containers.
- Label diskettes with a felt-tipped pen (not a ballpoint pen or a pencil) to avoid damaging them.
- Do not store a diskette in direct sunlight or near heaters. The temperature of the storage area should not fall below 50°F (10°C) or rise above 125°F (52°C).
- Store diskettes away from strong magnetic fields and do not let the diskettes touch any steel objects. Magnetic fields (produced by motors, generators, and transformers) can erase data.

D.2 PROTECTING DISKETTES FROM ACCIDENTAL OVERWRITING

The RX50 has a write-protect feature to prevent loss of data by accidental overwriting. To protect data on a diskette, cover the slot on the side of the plastic cover with a write-protect tab. Diskettes are supplied with a packet of adhesive-backed tabs.

Remove the tab when you want to write data on a diskette.

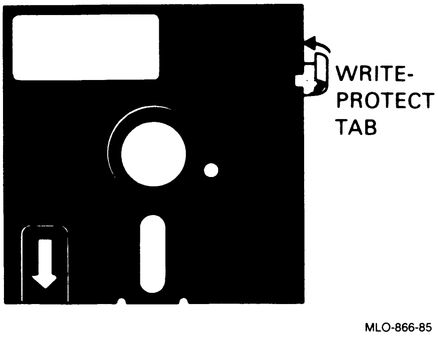
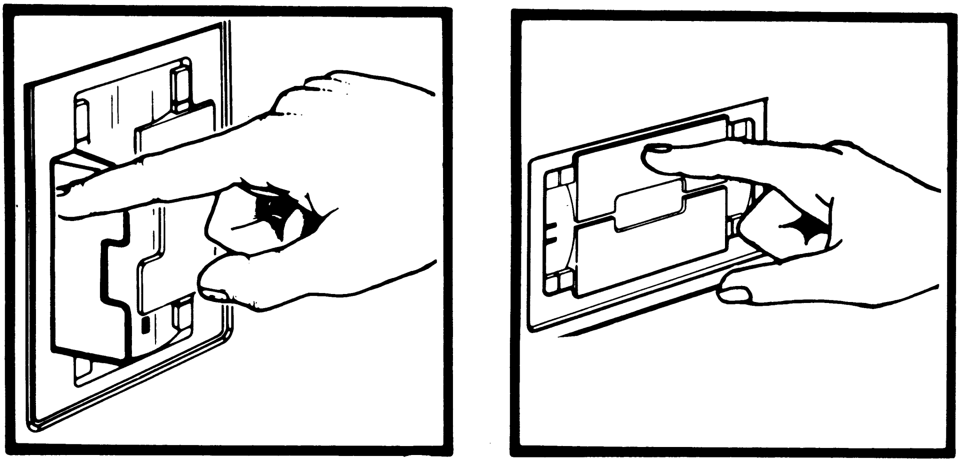


Figure D–3: Write Protecting Diskettes

D.3 INSERTING DISKETTES

Do not attempt to open a door while a drive's activity light is on, as this might damage the drive heads. You must wait for the light to go off, since this indicates that disk activity has finished.

To open the door to drive 1 (CSA1), press the left (or upper) half of the left or upper drive door. To open the door to drive 2 (CSA2), press the right (or lower) half of the right or lower drive door.



MLO-867-85

Figure D–4: Opening RX50 Diskette Drive Doors

To insert a diskette into the left (or upper) drive, make sure that the write-protect notch is facing downward (or to the left). For the right (or lower drive), the notch should be facing upward (or to the right). When you use diskettes made by DIGITAL, line up the orange arrow on the diskette with the orange alignment bar on the drive.

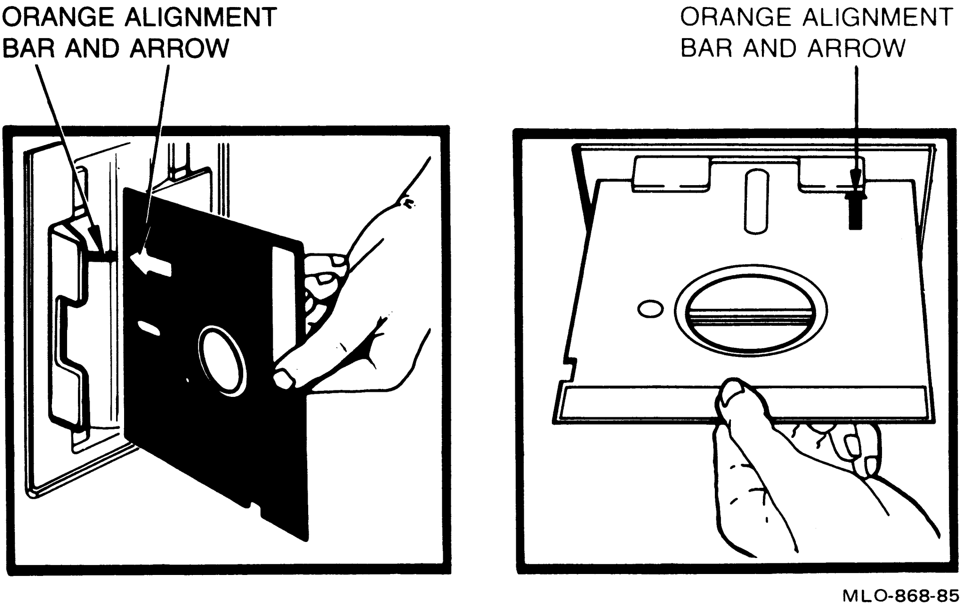
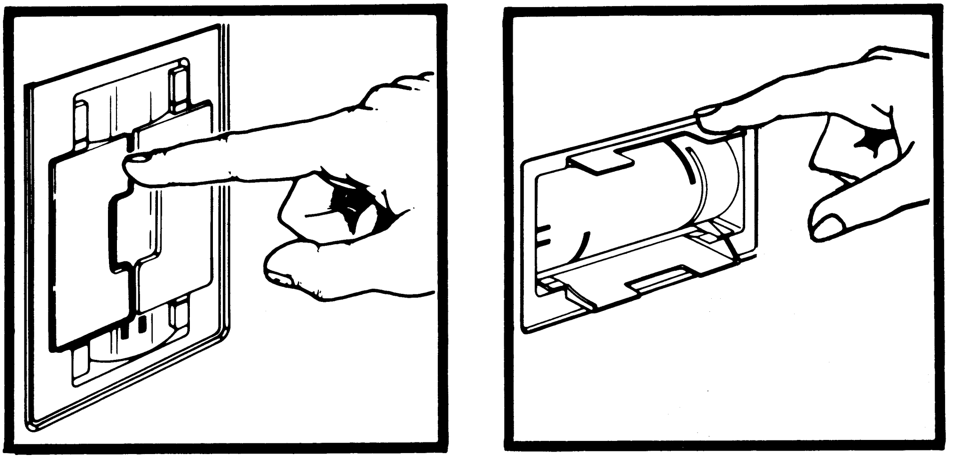


Figure D–5: Inserting Diskettes

D.4 CLOSING DRIVE DOORS

Close the door to drive 1 by carefully pushing it to the right (or down). Close the door to drive 2 by pushing it to the left (or up). Both doors should be flush with the front cover. Do not use excessive force; the door should close easily. Always close the door after inserting a diskette or after removing one.



MLO-869-85

Figure D–6: Closing Drive Doors

Deposit and Examine Console Commands E

The D (Deposit) and E (Examine) commands let you write and read data in memory and registers almost anywhere in the computer system (exceptions include the VAXBI node private spaces and boot code in the EEPROM). You can use these commands only when the console terminal is in console mode and is displaying the console mode prompt >>>.

In response to the D command, console microcode deposits the data in the specified address. You must specify an address and some data with the D command; if you omit either, the error message ?44 appears on the console terminal (see Appendix B).

In response to the E command, console microcode reads the contents of the address you specify. The console terminal displays a character describing the address space (P, I, G, E, or M), the address, and the data. The address argument is optional in the E command.

Qualifiers, described in Section E.2, make D and E commands more specific. A qualifier is a slash mark followed by a letter, such as /M in the Examine command E/M. The D and E commands use the same qualifiers, except that the D command does not use the /M qualifier.

E.1 TYPING THE COMMANDS

Type the commands in response to the console mode prompt >>>.

To deposit data in the EEPROM, you must set the lower control panel switch to the Update position. No specific position is required for depositing data in any other locations or for examining data.

E.1.1 Depositing Data

To deposit data, use the following format:

D<qualifiers> [address] [data] (RET)

where:

<qualifiers>	are data size and/or address space qualifiers (see Section E.2)
[address]	is the hexadecimal address of the location into which data will be deposited
[data]	is data (in hexadecimal) to be deposited.

E.1.2 Examining Data

To examine data, use the following format:

E<qualifiers> [address] (RET)

where:

<qualifiers>	are data size and/or address space qualifiers (see Section E.2)
[address]	is the hexadecimal address of the location to be examined

NOTE
You can use the symbolic address P to deposit or examine data in the Processor Status Longword (PSL). The data size is longword, and the microcode ignores any data size specified qualifier. Following a D command to the PSL, the default address is set to the PSL for a subsequent E command. Depositing data in the PSL may leave the processor in an unpredictable state when it returns to program I/O mode (running under the operating system).

E.2 QUALIFIERS

The two types of qualifiers are those specifying an address space and those specifying a data size. You can type the qualifiers in any order after the command, as shown in Section E.1.

The qualifiers for the D and E commands are optional. The default address space is physical, the default data size is longword, and the default address is 0. These defaults exist under only three conditions:

- After processor initialization
- After entry into console mode
- After execution of an N command

Under any other condition, the defaults for address space and data size are those used in the last D or E command; the default address is the last address plus the last data size used in a D or E command.

E.2.1 Data Size Qualifiers

The data size qualifiers are as follows:

Qualifier	Data Size
/B	Byte
/W	Word
/L	Longword

E.2.2 Address Space Qualifiers

Address space is divided into six categories:

- Physical addresses
- Virtual addresses
- Internal processor register addresses
- General-purpose register addresses
- EEPROM option addresses
- M chip internal registers

Table E–1 shows the address space qualifiers.

Table E–1: Address Space Qualifiers

Qualifier	Address space
/P	Physical memory. You can use data size qualifiers /B, /W, or /L with /P.
/V	Virtual memory. If memory mapping is not enabled, the microcode treats the address as physical. When you examine a virtual memory location by using E/V and memory mapping is enabled, microcode displays the physical address corresponding to the virtual address you request. You can use the data size qualifiers /B, /W, or /L with /V.
/I	Internal processor registers (IPRs) accessible to software with MTPR and MFPR instructions. The data size is longword, and the microcode ignores any specified size qualifier.
/G	General-purpose registers R0-R15. The data size is longword, and the microcode ignores any specified size qualifier.
/E	Customer options section of the EEPROM. The data size is byte, and the microcode ignores any specified size qualifier. You can write data in 24(decimal) EEPROM locations if the lower key switch on the control panel is in the Update position. The address you specify must be in the range of 0 to 17(hex). If the address is out of range or the lower key switch is not set to Update, the error message ?47 appears on the terminal (see Appendix B). You can examine these EEPROM locations even when the lower key switch is not in the Update position.
/M	64 CPU internal registers in the M chip. The size is longword, and the microcode ignores any specified size qualifier. You can use the E/M command following a double-error halt to read the contents of the stack stored in the M chip registers. You cannot examine M chip register 1F(hex), the Processor Status Longword Temporary Register. If you try, the console terminal will print the error message ?45. You can use the /M qualifier only with the E command.

E.3 EEPROM CUSTOMER OPTIONS

In most cases, data in the EEPROM should be examined and changed by using the EEPROM Utility. However, you can change and examine EEPROM data with D and E commands by using the /E qualifier. Do not use a qualifier other than the /E qualifier to deposit data into the EEPROM. Table E–2 shows the addresses of EEPROM data accessible to the D and E commands.

Table E–2: Accessible EEPROM Addresses

Byte Number	Function	Implementation
0–2	Reserved for DIGITAL	
3	RCX50 self-test disable	Bit <4> (1=disable; 0=enable) Bit <3> must be 1 Bits <7:5;2:2> must be 0
4	Logical console node ID	Bits <7:4> identify the VAXBI node number of the logical console
5	Reserved for DIGITAL	
6	Default console baud rate	Bits <7:0> =30 — 150 baud 31 — 300 baud 32 — 600 baud 33 — 1200 baud 34 — 2400 baud 35 — 4800 baud 36 — 9600 baud 37 — 19200 baud
7*	F chip disable BTB disable Cache disable	Bit <0> (1=disable; 0=enable) Bit <1> (1=disable; 0=enable) Bit <2> (1=disable; 0=enable)
8–17	Reserved for DIGITAL	

* Do not change byte 7; bits <2:0> of byte 7 should always be 0.

E.4 EXAMPLES OF DEPOSIT AND EXAMINE COMMANDS

```
>>>D/L/P 5050 35353535              ! Deposit 35353535 (hex) at
                                    ! physical address 0000 5050.

>>>E/L/P 5050                       ! Examine the longword at address
                                    ! 0000 5050.

      P    00005050 35353535        ! Physical address 0000 5050
                                    ! contains the data 35353535
                                    ! (hex).

>>>E                                ! Examine the next location.

      P    00005054    00FF00FF

>>>D/W/P 5054 0607                  ! Deposit 0607 (hex) at address
                                    ! 0000 5054.

>>>E/B/P                            ! Examine the byte at physical
                                    ! address 0000 5056.

      P    00005056    FF

>>>E/L/V 0204                       ! Examine the longword at virtual
                                    ! address 0000 0204.
      P    00030404    69696969     ! The physical address is
                                    ! 0003 0404.

>>>E/G 2                            ! Examine general purpose
                                    ! register R2.

      G    00000002    00050005
>>>
```

Control Flags for Booting F

You can control various phases of the booting procedure by setting bits in General Purpose Register R5 with the console command B/R5:<data>, where the <data> is in hexadecimal. (See Table F–1.) For example, if you wanted to set bit 4 in R5 when booting, you would type B/R5:10. These bit functions are defined by the VMB primary boot routine and by VMS. The value –1 in R5 is reserved for DIGITAL.

Table F–1: Bit Functions of General Purpose Register 5

Bit Number	Function
0	Conversational boot. At various points in the system boot procedure, the boot code prompts you for data on the console terminal. If bit <4> is also set, the VAX Diagnostic Supervisor should start and prompt you for devices to test.
1	Debug. If this flag is set, VMS maps the code for the XDELTA debugger into the system page tables of the running VMS system.
2	Initial breakpoint. If this flag is set, VMS executes a breakpoint (BPT) instruction immediately after enabling mapping.
3	Secondary boot from boot block. The secondary boot is a single 512-byte block whose logical block number is specified in General Purpose Register R4.
4	Boots the VAX Diagnostic Supervisor. The secondary loader is an image called DIAGBOOT.EXE.
5	Boot breakpoint. This stops the primary and secondary loaders with a breakpoint (BPT) instruction before testing memory.
6	Image header. The transfer address of the secondary loader image comes from the image header for that file. If this flag is not set, control shifts to the first byte of the secondary loader.

(Continued on next page)

Table F–1: Bit Functions of General Purpose Register 5 (Cont.)

Bit Number	Function
7	Memory test inhibit. This function sets a bit in the PFN bit map for each page of memory present, inhibiting the memory test.
8	File name. VMB prompts for the name of a secondary loader.
9	Halt before transfer. VMB executes a HALT instruction before transferring control to the secondary loader.
13	Specifies that a more extensive algorithm be used when testing main memory for uncorrectable hardware (RDS) errors.
15	Used by the VAX Diagnostic Supervisor.
16	Specifies that memory pages with correctable (CRD) errors not be discarded when booting begins. By default, pages with CRD errors are removed from use during the booting memory test.
31:28	Specifies the top level directory number for system disks in multiple systems.

G VAX Disk Formatters

If a disk is corrupted, you can reformat it by using a specific program contained in the VAX Disk Formatters diskette. The diskette contains the following formatters:

- EVRAC.EXE—Reformats disks in the RK06, RK07, RP05, RP06, RP07, RM05, and RX02 disk drives.
- EVRLB.EXE—Reformats disks in the RA60, RA80, and RA81 disk drives.
- EVRMC.EXE—Reformats disks in the RC25 disk drive.

To obtain help about running each formatter, follow these steps:

1. Turn the upper key switch to the Enable position. Turn the lower key switch to the Halt position.
2. Insert the Diag Super + Auto diskette into console drive 1.
3. Type B/R5:10 in response to the console mode prompt >>> to boot the VAX Diagnostic Supervisor. The DS prompt appears.
4. Remove the Diag Super + Auto diskette and insert the VAX Disk Formatters diskette into console drive 1.
5. In response to the DS> prompt, type HELP followed by the name of the formatter. For example, type HELP EVRLB to obtain help about EVRLB.EXE.

Glossary

Adapter

VAXBI option that allows communication between the VAXBI bus and other buses or devices.

Attached Processor

The second processor of a multiprocessing system. The primary processor is in the right-most VAXBI slot; the attached processor can be in any other slot.

Boot (bootstrap)

To bring software into system memory.

Boot code

A program that lets the processor boot the operating system from a specific device.

Bus

A set of electrical conductors that carry signals to various components of the computer.

Checksum

A number generated from an algorithm performed on data. A checksum is used to verify that data has not been changed.

CIBCI

The adapter that connects the CIBCI cabinet and the CI bus to the VAXBI bus.

CI adapter

An adapter that connects the CI bus to the VAXBI bus.

Cold start

An attempt by the processor to boot a fresh copy of the operating system.

Console mode

A mode of operation in which the console terminal communicates directly with a processor.

Control store

A combination of microcode in the read-only memory and patches in the random-access memory.

Controller

A device that controls the flow of data between the processor and stored data.

CPU

Central processor unit.

Device type

A 2-letter code that represents a particular type of device. For example, CS is the console RX50 drive, and DU is either a RA60 or a RA81 disk drive.

DWBUA

UNIBUS adapter.

EEPROM

Electrically Erasable Programmable Read-Only Memory.

HSC controller

File server for VAXclusters. An HSC controller lets systems in a VAXcluster share disks.

KA820 processor

The processor used by the VAX 8200 and VAX 8300 computers.

KDB50

The adapter for MSCP (mass storage control protocol) devices.

Microcode

A program fixed permanently in read-only memory.

Module (VAXBI)

A printed circuit card that fits into one of the multiple slots on the VAXBI bus.

Multiprocessing

The combination of two or more processors working together to increase processing speed and capacity for compute-intensive applications.

Node (VAXBI)

A VAXBI interface that occupies one of 16 logical locations on the VAXBI bus. A node exists for each VAXBI option. If the option is a bus adapter, the node includes the bus and all devices attached to the bus, in addition to the adapter itself.

Option (VAXBI)

A DIGITAL product, such as a processor, a bus adapter, memory, or a disk controller, on the VAXBI bus. An option consists of one or more VAXBI modules.

Patch

Code that replaces an instruction in the microcode. The processor loads patches from the EEPROM into the microcode's associated random-access memory.

RAM

Random-access memory.

ROM

Read-only memory.

UNIBUS

A high-speed bus, linking I/O devices to a UNIBUS adapter.

Unit number

The number assigned to a device identifying the location of the device on a node. In the device designation ddnu, for example, u is the unit number. DIGITAL Field Service affixes unit numbers to the fronts of devices when they install the VAX 8200/8300 system.

VAXBI bus

The 12-slot or 24-slot system bus used by the VAX 8200/8300 computer.

VAX Diagnostic Supervisor

Software that loads and runs diagnostic and utility programs.

Warm start

An attempt by the CPU to restart the operating system at the point where it stopped.

Index

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA and Puerto Rico call 800–258–1710

In Canada call 800–267–6146

In New Hampshire Alaska or Hawaii call 603–884–6660

ELECTRONIC ORDERS (U.S. ONLY)

Dial 800–DEC–DEMO with any VT100 or VT200 compatible terminal and a 1200 baud modem. If you need assistance, call 800–DEC–INFO.

DIRECT MAIL ORDERS (U.S. and Puerto Rico*)

DIGITAL EQUIPMENT CORPORATION
P.O. Box CS2008
Nashua, New Hampshire 03061

DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.
940 Belfast Road
Ottawa, Ontario, Canada K1G 4C2
Attn: A&SG Business Manager

INTERNATIONAL

DIGITAL
EQUIPMENT CORPORATION
A&SG Business Manager
c/o Digital's local subsidiary
or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Northboro, Massachusetts 01532

*Any prepaid order from Puerto Rico must be placed with the Local Digital Subsidiary:
809–754–7575

VAX 8200/8300
Owner's Manual
AZ-GN4AC-TE

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent.

__ Assembly language programmer
__ Higher-level language programmer
__ Occasional programmer (experienced)
__ User with little programming experience
__ Student programmer
__ Other (please specify) ____________

Name ____________ Date ____________

Organization ____________ Telephone ____________

Street ____________

City ____________ State ____ Zip Code or Country ____________

Do Not Tear — Fold Here and Tape

digital

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
CORPORATE USER PUBLICATIONS
MLO5–5/E45
146 MAIN STREET
MAYNARD, MA 01754–2571

Do Not Tear — Fold Here

Cut Along Dotted Line