Console User's Guide VAX 8800/8700/8550/8500

AA–FH28C–TE

digital

software

CONSOLE USER'S GUIDE
VAX 8800/8700/8550/8500

Order Number: AA-FH28C-TE

This book describes the console operation for the VAX 8800, 8700, 8550, and 8500, including a hardware and software overview, and all system and diagnostic commands.

Software Version: VAX 8800/8700/8550/8500 Console Version 1

digital equipment corporation
maynard, massachusetts

July, 1986

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

This document was prepared using an in-house documentation production system. All page composition and make-up was performed by TEX, the typesetting system developed by Donald E. Knuth at Stanford University. TEX is a registered trademark of the American Mathematical Society.

Contents

FIGURES

TABLES

Preface

Manual Objectives

This manual is a reference and tutorial guide with the following objectives:

- To introduce the VAX 8800, VAX 8700, VAX 8550, and VAX 8500 systems and explain how to use the console to control them
- To outline system control procedures such as those for powering up, initializing, booting, and stopping the systems
- To describe the console and Micromonitor commands and their effect on the VAX systems
- To provide pertinent reference information about the hardware and software components of the VAX and console systems

Intended Audience

This manual is for VAX system managers, operators, and Field Service representatives; that is, those responsible for the control, management, and repair of the computer systems described in this manual.

Structure of This Document

This manual combines the features of a user's guide and a reference manual. Chapters 1 through 4 introduce the product and identify tasks that must be performed on the VAX systems, providing instructions. Chapters 5 and 6 and appendixes A through F are reference material.

The manual consists of the following chapters and appendixes:

Chapter 1, Overview, highlights the manual's major points and lists the steps for operating the console.

Chapter 2, System Hardware, describes the VAX hardware modules, including the console components, power system, and the Environment Monitoring Module.

Chapter 3, Console System Software, describes the VAX console components and their functions. The descriptions include the logical and physical CPUs, the software-implemented processor control panel, modes of console operation, multiple command streams, status monitoring, and miscellaneous software commands.

Chapter 4, Using the Console, explains how to initialize and start the console, start and restart the VAX, move between console and program modes, and power down the system.

Chapter 5, The Console Commands, begins with a description of the scope and structure of the console commands, including definitions of command defaults, names, qualifiers, parameters, options, prompts, and special keys. (The conventions used to explain the commands are included in the Preface.)

Chapter 5 then describes each console command including its parameters, qualifiers, options, and defaults. The commands are ordered alphabetically.

Chapter 6, The Micromonitor Commands, describes the Micromonitor commands (used in the diagnostics software), including parameters, qualifiers, options, and defaults.

Appendix A, Console Command Syntax, lists the commands alphabetically, showing their proper syntaxes and abbreviations (using the conventions described in the Preface).

Appendix B, Numeric and Address Data, lists the numeric literals and symbolic addresses mentioned in Chapters 5 and 6. Brief descriptions of the console State Bit Definitions and the Internal Processor Registers are also included.

Appendix C, Error Messages, lists console error messages with brief descriptions.

Appendix D, Bootstrap Control Flags to VMB, lists the bootstrap control flags with their bit positions, symbolic names, and meanings.

Appendix E, Installing the Console Operating System and Application, provides the installation instructions for the console operating system (P/OS) and the console application.

Appendix F, System Command Files, lists and briefly defines the system command files to initialize, start, halt, restart, and exit the system.

Command Language Conventions

The conventions used to describe the commands in this manual (Chapters 5 and 6) are as follows:

!	In examples, an exclamation point indicates that a comment follows. Characters following the exclamation point are ignored.
( )	Parentheses enclose command options of which one must be used. Command options are usually documented as separate commands in this manual. Do not include the parentheses in the command line.
[ ]	Square brackets enclose command options that are not required. Do not include the brackets in the command line.
|	A vertical bar indicates "or". For instance, :|= means that either the colon or equal sign may be used.
parameter	A parameter follows a command name without brackets or parentheses, indicating it is a required part of the command. For instance, the command syntax: Command-name filename indicates that a file specification is a necessary part of the command.
RETURN	Represents the carriage return key on the keyboard.
CTRL/n	When CTRL/ prefixes a character, press the CTRL key with the character that follows.
TAB	Represents the TAB key on the keyboard.
UPPERCASE	This manual presents commands and qualifiers in uppercase characters. They can be entered in uppercase, lowercase, or a combination.
lowercase	Syntax characters in lowercase indicate generic items to be be entered with the command. For example, "file-name" is the generic term for the name of a file.
bold	In Appendix A, the shortest abbreviations you can use with commands, options, and qualifiers appear in boldface.
red ink	User input appears in red ink.

Associated Documents

- *VAX 8800/8700 Diagnostic User's Guide* (EK–KA88D–UG–001)
- *VAX 8500/8550 Diagnostic User's Guide* (EK–KA85D–UG–002)
- *VAX 8800/8700 System Installation Guide* (EK–8800I–IN–001)
- *VAX 8500/8550 System Installation Guide* (EK–8500I–IN–002)
- *VAX 8800/8700 System Maintenance Guide* (EK–88XV1–MG–001)
- *VAX 8500/8550 System Maintenance Guide* (EK–85XVS–MG–002)
- *VAX 8800/8700 System Hardware User's Guide* (EK–8800H–UG–001)
- *VAX 8500/8500 System Hardware User's Guide* (EK–8500H–UG–002)

Chapter 1

Manual Overview

1.1 Introduction

This manual instructs you in operating the VAX 8800, 8700, 8550, and 8500 systems through their consoles. The VAX 8800 system is a dual-processor (2-CPU) model. The CPUs operate under single copies of the VAX operating system and are controlled by the operator console. The VAX 8700, 8550, and 8500 systems are single-processor (1-CPU) models. The systems accept instructions from the console and uses a software-implemented processor control panel.

This manual describes the console operation for the four indicated VAX models. Most descriptions in the manual apply to all models, so the term VAX is used. If a statement does not apply to all models, the model number is indicated.

1.2 Major Hardware and Software Components

The console is a complete computer system, with its own CPU, storage devices, and operating system. From the console, you can control the major hardware components of the VAX system unit.

The console program is the primary software component and runs as an application under the console's operating system (P/OS). With the console program, you control and manipulate the VAX hardware and software components. These components include the processor control panel (entirely in software), two modes of console operation, three command streams, and status monitoring.

1.3 Using the Console

DIGITAL Field Service installs the VAX system and console. The first time the system is started, some command files need to be modified. For more information, see Section 4.3. After the first start, you turn on the switch to power up and bootstrap the system. Software required to operate the console system, including the console operating system (P/OS) and command files, is installed at the factory.

If the software (or storage device) fails, follow the instructions in Appendix E for reinstalling the console operating system (P/OS) and the console program.

1.4 The Console Commands

You use the console commands more than any other aspect of the VAX console software. The console commands specify and control activity on the VAX system unit. These commands have names, parameters, defaults, qualifiers, and options – all of which require a language and syntax. Several special switches also control the operation of the commands.

1.5 Before You Start

To use the VAX effectively, you should understand the following topics:

- The console hardware
- The VAX power system
- The console program
- The system startup procedure
- The Console Command Language and commands

Chapter 2

System Hardware

This chapter introduces you to the VAX system, including the power system hardware modules, the Environment Monitoring Module, and the console.

2.1 Product Description

The system hardware consists of the processor(s) (two on the VAX 8800 one on the 8700, 8550, and 8500), memories, and I/O unit(s) (two on the VAX 8800, one on the 8700, 8550 and 8500), connected by a common set of backplane wires. I/O consists of the BI (Backplane Interconnect) which allows for NI (Network Interconnect), and CI (Cluster Interconnect) capabillities. These bus architectures, coupled with a 70MB/sec CPU /Memory band width and four I/O channels, provide high I/O flexibility and performance. An Environmental Monitoring Module (EMM) monitors voltage, temperature, and air flow, protecting the CPU. A single console system controls the processor(s). Figures 2–1 and 2–2 are block diagrams of the VAX 8800, 8700, 8550, and 8500 system hardware.

2.2 The Console System

The VAX console is a secondary computer system that allows you to control and monitor the primary VAX systems. The console contains a processor and memory for managing the transactions between you (the operator) and the VAX system, a storage device for microcode, and additional disk space for the console, bootstrap, and diagnostic programs. The console also has terminal interfaces to provide access to both local and remote terminals. A console interface links the console subsystem

Figure 2–1: VAX 8800/8700 System Hardware

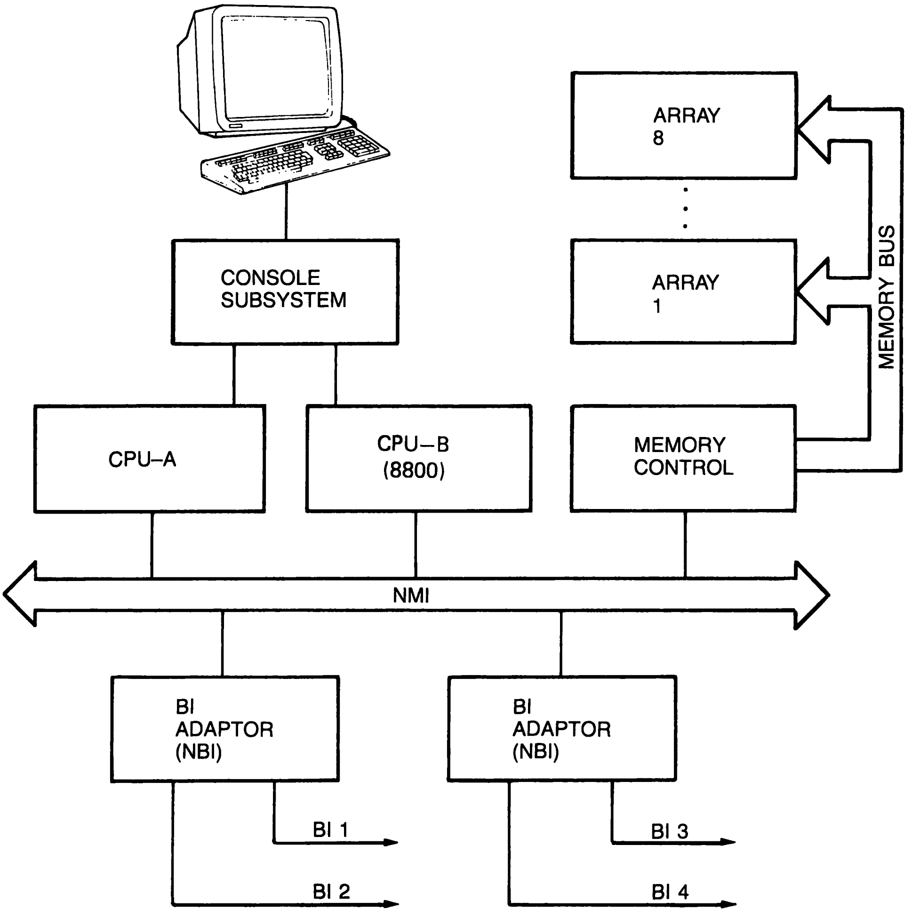
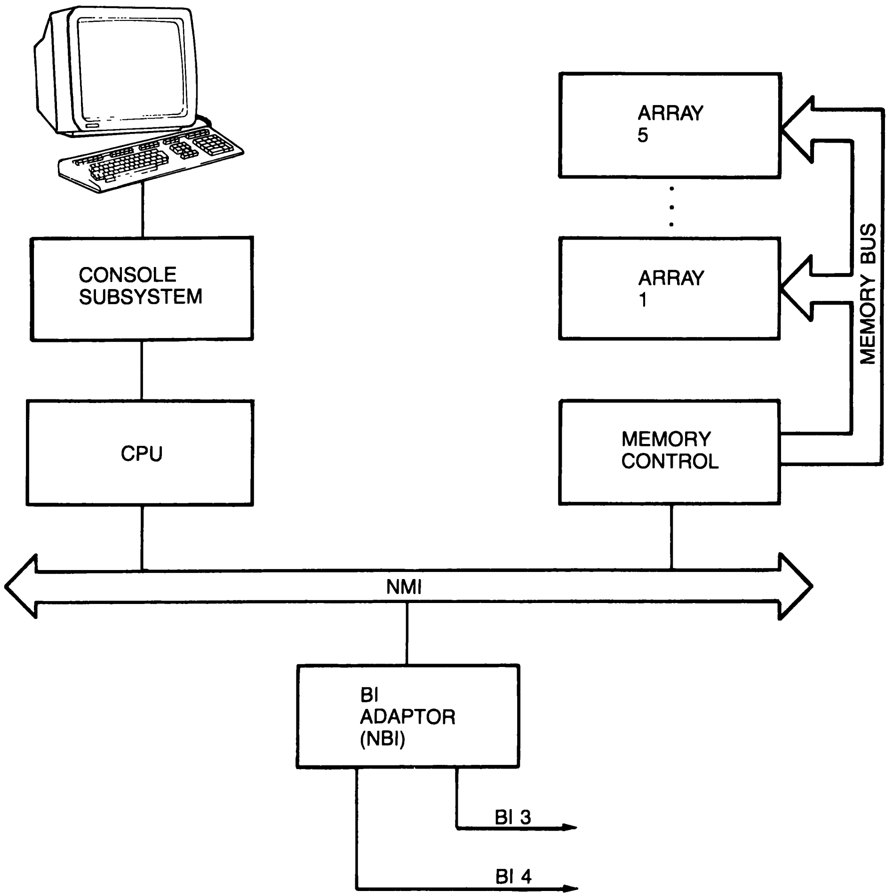


Figure 2-2: VAX 8550/8500 System Hardware



and the VAX processor. The console interface consists of two parts: a Real Time Interface (RTI), which extends the console's I/O capabilities, and a section of logic on the VAX clock module dedicated to supporting console functions. Figure 2–3 is a block diagram of the console subsystem hardware.

On the VAX 8800, two logically independent consoles reside in one physical console, allowing both CPUs to be controlled independently. (Some interdependencies exist, such as certain clock functions.)

You control the VAX CPU(s) at the console with the Console Command Language, described in Chapter 5. These commands provide control over the VAX CPU(s) with output to a CRT screen and an optional printer.

2.2.1 Console Hardware

The console contains 1024 kilobytes (KB) of memory, RX50 diskettes, and a fixed disk. The console comes with a VR201 monochrome monitor and an LK201 keyboard. An optional printer is available.

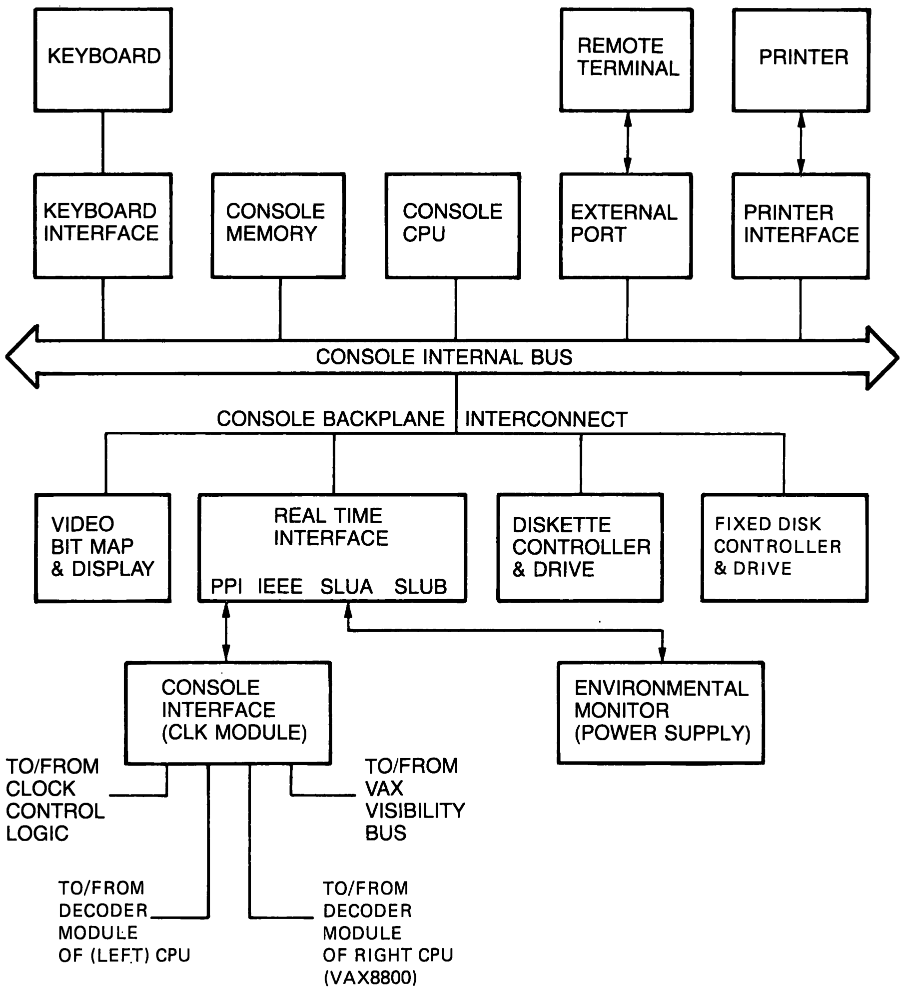
2.2.1.1 Printer Interface

A printer is a console option connected to the console through a serial printer port on the back of the console system unit. The port is capable of performing at programmable rates up to 19.2 Kbaud and uses RS423 signal levels.

2.2.1.2 Console Storage Device

VAX microcode, console programs, and the programs that help to run or diagnose the VAX and console systems are stored on the console's fixed disk. Diskettes can be used as intermediate storage between the VAX and the console. The console has sole write access its own fixed disk, and the VAX operating system has write access to the diskettes.

Figure 2–3: Console Hardware



2.2.1.3 Time of Year Clock

The console has its own clock, which uses an MC146818 CMOS chip with backup power supplied by a rechargeable NiCD battery. The console increments the clock every second. A Valid RAM and Time (VRT) bit indicates the status of the clock. (An asserted VRT bit indicates that the clock has not lost power and that the time and date have been updated properly since the last initialization.) The VAX Time of Day Register is implemented in console software, using the console's Time of Year Clock. The VAX CPU requests Time of Year information from the console when necessary.

2.2.1.4 Real Time Interface

The console communicates with the CPU(s) over a Real Time Interface module (RTI), using the 24-bit Parallel Peripheral Interface bus (PPI) connected to the VAX clock and console interface module located in the CPU backplane. One of the two RS423 serial line interfaces on the RTI is used to communicate with the Environmental Monitoring Module (EMM), to enable the console to monitor and control the VAX power subsystem. The other RS423 interface is not used.

2.2.1.5 External Port

The console system module offers a Communication Port—an asynchronous or bit- and byte-synchronous RS423 port with modem control. This port is used to establish a diagnostic link to the system from a remote location. A 7201 USART is employed, operating at a rate up to 19.2Kbaud asynchronously or 740Kbaud synchronously. This port can be accessed from the back of the console system unit. When a modem is connected to the Communication Port, it operates like a typical VAX remote diagnostic port. Use a BC22F 25-pin connector to connect the modem to the Communication Port.

Use a null modem cable to connect a remote terminal to the Communications Port of the console system unit.

2.2.1.6 Console Interface

The console interface resides on the VAX clock module and shares it with the VAX system clock.

The console interface allows the console to communicate with and control the VAX system and provides the VAX with access to the console subsystem components. The console interface is linked to the rest of the console subsystem by a 24-bit bus from the Programmable Peripheral Interface on the Real Time Interface (RTI).

The VAX side of the console interface contains data, address, and control paths to the Instruction Decoder module of the CPU. For the VAX 8800, a duplicate set of connections is available to the second CPU.

The console controls the system clock. The clock can be stopped and restarted, stepped any number of clock cycles, or stopped at preselected program breakpoints by commands entered at the console terminal.

The console has access to the 16 channels of the VAX 8800 Visibility Bus, which provides access to selected latches of the VAX 8800 system for diagnostics. Each channel is a serial path from one of the 16 modules of the left and right CPUs of the VAX 8800 or from one of the 8 modules of the VAX 8700, 8550, and 8500.

Each VAX CPU has seven Internal Processor Registers on the console interface. Four registers are for passing data to and from the console processor: the Console Receive Data Buffer, Console Receive Control and Status, Console Transmit Data Buffer, and Console Transmit Control and Status. The remaining three registers are used in the Interval Clock to time programmed intervals for the VAX CPU. These are the Interval Count Register, Next Interval Count Register, and the Interval Count Control and Status. One set of the seven IPRs on the interface exist for each CPU.

Partial contents of the System Identification Register are accessed through the console interface. The console takes this data, along with the remainder of the system identification data stored on the console disk, and deposits it in the Slow Data File (SDF) at boot time. As a result, the CPU looks for System Identification Register contents only in the Slow Data File. See the descriptions of the DEPOSIT, EXAMINE, and LOAD commands in Chapter 5 for more information about the Slow Data File.

2.3 Power System

The VAX power system consists of a Modular Power Supply Module (MPS), an Environmental Monitoring Module, a fan, and AC and DC power distribution. The VAX 8800 also has a battery backup unit (BBU).

2.3.1 Environment Monitoring Module

The Environmental Monitoring Module controls the VAX power system and protects the VAX hardware from environmentally induced damage. The EMM senses the state of the system's environment and relays this information to the console. If the temperature exceeds the limits or if the air flow goes below specified limits, the console issues commands to the EMM to turn the power off. When the situation is corrected, the EMM will turn the power back on.

When the system's power-up sequence is initiated, both the console and the EMM undergo self testing, after which the console loads its operational software. As part of the system power-up sequence, the console must initialize each EMM's parameter table and then instruct it to power on the rest of the computer logic. After the power-up sequence, the console requests the EMM to compare the environmental conditions with the nominal values in the initialization sequence and to report any detected discrepancies. The EMM senses the module temperature, inlet air temperature, air flow, and DC voltage levels. Power failure signals are generated in the EMM, and the EMM implements the parallel and serial keying of modules to detect when a module is missing or in the incorrect slot.

Problems that can cause the EMM to turn off the system power might include regulator voltages out of tolerance, excessive air temperatures in the cabinet, or heat buildup across the logic components, insufficient air flow, and changes in the MODULE OK signals from the regulators.

Because the Modular Power Supply is controlled by the system console, the MPS subsystem acts as an intelligent peripheral, with the EMM as the controller.

Chapter 3

Console System Software

The VAX console software is a set of application tasks built on the console operating system (P/OS). This chapter describes those tasks.

The console provides control over one or two CPUs with the following general functions:

- Control and monitor the VAX power system through the Environmental Monitoring Module
- Load microcode
- Provide a "soft" processor control panel—software that is equivalent to processor control panel switches and lights on previous VAX systems
- Initialize the CPU(s) and major buses
- Provide a means to identify, boot, or restart the VAX 8700/8500/8550 CPU
- Provide a means to identify, boot, or restart the primary and secondary CPUs on the VAX 8800
- Provide console, program, and remote user modes
- Control diagnostic sessions

The console subsystem is the primary interface for service activity and in-depth troubleshooting. The console software application allows the manipulation of registers, control stores, and memory and acts as an operator service terminal for the VAX operating system.

The console program implements the program and console modes, log files, EMM support, the Micromonitor, and remote access. The console functions not implemented by the console program are the logical block server, the file server, spawned microdiagnostics, and drivers. The console program is the only way to power down and power up the VAX, and therefore must always be running.

The console program maintains constant read requests to the EMM, the remote port (when enabled), and the local terminal. In console mode, the local and remote read requests are for entire lines; in program mode, the requests are for single characters. The local and remote ports can be in different modes or the same mode. Therefore, the console's terminal driver uses console mode to implement special characters, such as CTRL/U and the DELETE key. The VAX's terminal driver uses program mode to implement these special characters.

3.1 Logical and Physical CPUs on the VAX 8800

The two VAX 8800 CPUs are known logically as the Primary and the Secondary CPUs. Physically, they are known as Left and Right. Console software maps the logical CPUs, Primary and Secondary, to the physical CPUs, Left and Right, respectively.

3.2 Scope for Commands That Affect More Than One VAX 8800 CPU

Some commands, such as EXAMINE, apply only to one of the VAX 8800 CPUs. Other commands including the initialization commands HALT, INITIALIZE, LOAD/WCS, LOAD/CACHE, LOAD/IBD, LOAD/SDF, SENSE REVISION CPU, START/CONTROLSTORE, TEST, VERIFY MODULE PLACEMENT, and VERIFY REVISION CPU apply to either or both CPUs.

A VAX 8800 CPU reponds to a command when it is both available and selected. The CPU is available if the following conditions are true:

- The CPU has passed the VERIFY MODULE PLACEMENT test.
- The CPU is enabled because it is the primary CPU or because the ENABLE SECONDARY command has been issued successfully.
- The CPU is working (has passed all diagnostics) and thus has not been marked as unavailable.

A VAX 8800 CPU is "selected" by the SET CPU console command. You can vary the scope of a command by using the SET CPU qualifiers BOTH, LEFT, RIGHT, CURRENT_PRIMARY, NEXT_PRIMARY, CURRENT_SECONDARY, and NEXT_SECONDARY.

The CPU is marked "unavailable" if any of the following commands fail:

LOAD/WCS
LOAD/CACHE
LOAD/IBD
LOAD/SDF
VERIFY MODULE PLACEMENT

3.3 The System Clocks

Commands that require clocks to be off in order to execute check the clock state (ON/OFF), and the state of the CPU (RUNNING/HALTED). If the clocks are on and either CPU is running, the console issues a prompt asking if you wish to stop the clocks before continuing. If you reply with a "Y", the clocks are stopped and the command executes. If you reply with anything else, the command terminates without being executed. If the CPU(s) are halted and the clocks are in the wrong state, the console puts the clocks in the correct state and executes the command.

CAUTION

Stopping the clocks while the operating system is running (on any CPU) can cause the operating system to crash.

3.4 Software-Implemented Processor Control Panel

Unlike previous VAX consoles, this console does not have a physical front panel with lights and switches. The typical front panel functions have been implemented in the console software.

Console software implements these "front panel" functions, using the following commands. (The commands are described in Chapter 5.)

- Controlling power to the VAX system:

  ENABLE/DISABLE AUTO POWERON
  POWER ON/OFF/STANDBY

- Enabling/disabling remote port:

  ENABLE/DISABLE REMOTE CONSOLE
  ENABLE/DISABLE REMOTE MODEM

ENABLE/DISABLE REMOTE USER

- Selecting the auto restart and boot functions:

  ENABLE/DISABLE AUTO BOOT
  ENABLE/DISABLE AUTO RESTART

The AUTO_POWERON, AUTO_BOOT, and AUTO_RESTART keyswitches are disabled on new systems.

3.4.1 Software-Implemented Keyswitches

Processor control panel keyswitches are implemented in software. The AUTO_BOOT and AUTO_RESTART commands work together as described in Table 3-1. These "switch settings" are checked in the SYSINIT.COM and RESTAR.COM files to determine whether to restart or boot the system following a power failure recovery or an error halt. The command file checks the AUTO_RESTART "switch setting" first. Depending on its setting or the outcome of that first operation, the AUTO_BOOT "switch setting" may or may not be tested.

Table 3–1: Software-implemented Keyswitches

AUTO_ RESTART	AUTO_ BOOT	RESULT
ENABLED	ENABLED	Attempts restart; if restart fails, reboots CPU
ENABLED	DISABLED	Attempts restart; if restart fails, halts CPU
DISABLED	ENABLED	Attempts reboot; if reboot fails, halts CPU
DISABLED	DISABLED	Halts CPU

Also see Figure 4-1 for more information on the software-implemented keyswitches. The AUTO_POWERON keyswitch determines whether power is applied to the CPU following a CPU power failure recovery.

WARNING

If repair work is necessary on the CPU, you must issue the POWER OFF command and shut off the main power breaker of the CPU. The following safety feature is designed to protect those working on the system hardware:

When you issue the POWER OFF command, the console saves the contents of the AUTO_POWERON "switch setting" and disables it. Then, with any kind of power failure (including turning off the console), the initialization procedure (SYSINIT) detects that AUTO_POWERON is disabled and does not power up the system. That prevents a console-only power failure, for instance, from powering up the entire system.

When the work on the system hardware is finished, turn on the circuit breaker and issue the POWER ON command. The AUTO_POWERON value is restored to its original setting (prior to the POWER OFF command).

If you have enabled the AUTO_POWERON keyswitch and have not issued a POWER OFF command, the console responds to power failures automatically. If the circuit breaker is then inadvertently turned on, the console automatically powers up the system, endangering those working on the system hardware.

3.5 Console Directory Organization

Files used by the console program reside in more than one directory. The directories used are [CONSOLE] and [8800], [8700], [8550], and [8500] (depending upon your system). The [CONSOLE] directory contains all files that are shared by all the VAX models: 8800, 8700, 8550, and 8500. The other directories contain the VAX processor-specific files. For example, there may be files in the [8800] and [8500] directories with the same name but which are actually different. To locate system files, the console first searches in the VAX-specific directory (according to the VAX model the console is connected to). If the desired file is not found, it searches in the [CONSOLE] directory.

From the console user's point of view, this organization has the following impact:

- If you need to locate a file that the console program uses (for example, to edit a command file) and you have an 8800, look for the file in [8800]. If the file is not there, look in [CONSOLE]. If the file is in neither directory, then the file is not usable by the 8800 console.
- If you want to create a new console command file for an 8500, for example, you should put it in the [8500] directory. This assures that the 8500 console program will locate the file and also that it would not be used if the console is later connected to an 8800.

- To run the console program from DCL, type RUN CONTROL if the default directory is [CONSOLE]. If you have changed the default directory, you must type RUN [CONSOLE]CONTROL since the CONTROL program is located in [CONSOLE]. The location of the default directory does not affect the console program.
- Because the search path is hard coded into the console program, you cannot specify a directory as part of a file name used in a console command. For example, the console command @[8800]TESTFILE is not allowed. You must use just @TESTFILE.
- One exception to the directory organization is the file CI780.BIN, which must be in the directory [USERFILES].
- Because of its size, the console log file LOGFILE.DAT is in directory [CONSOLE] instead of in several directories. This file differs between models. Therefore, if you connect your console to a different machine type delete the file [CONSOLE]LOGFILE.DAT before running the CONTROL program. Otherwise you will get an error message.

3.6 Modes of Console Operation

The VAX has two modes of operation: console mode and program mode. Local terminals have the same two modes: console mode and program mode. For remote terminals these modes are called remote console mode and remote user mode.

In console mode (local or remote), the VAX CPU can be either running or halted and the console terminal can execute some or all console commands. In program mode (called user mode for remote terminals), the CPU must be running and the console terminal cannot execute console commands.

3.6.1 Console Mode

In console mode, you can control and monitor the VAX (running or halted) from the console terminal. Though not all commands are available when the VAX is still running (in console mode prior to the HALT command), a set of console commands is available when the CPU is halted. These commands allow you to perform privileged operations, such as depositing to and examining locations in main memory, GPRs, and IPRs; stepping through macrocode or microcode; controlling the system clock; and more. You may also make a list of these console commands to be run as a program from the console processor.

The VAX CPU assists the execution of console command functions that the console cannot perform alone. For example, EXAMINEs and DEPOSITs to main memory or internal registers require CPU intervention because of the access limitations of the console. The CPU must be halted to allow the portion of microcode that is dedicated to assisting the console to run.

The VAX CPU does not respond to interrupts while halted. In order to communicate with the console during a halt, the CPU must poll the Ready and Done bits of the Console Terminal Control and Status Registers. The console processor, however, still employs interrupts in performing its side of the communication process.

3.6.2 Program Mode

While the console can be in console mode when the VAX is running or halted, program mode operates only while the CPU is running. In program mode, the console terminal is like a user terminal, passing input to software in the CPU and displaying the output. Console commands entered at the console terminal are not recognized, as they are no longer sent to the console processor for interpretation. One exception is CTRL/P on the local console terminal which returns the console to console mode. CTRL/P works on both the local and remote terminals to change from program mode to console mode. The one exception is that if REMOTE CONSOLE is disabled, CTRL/P on the remote port is treated like any other program mode character.

3.7 Multiple Command Streams

The local console is always enabled to some degree when the remote terminal is active (so that the remote terminal can be disabled). The commands available at the local terminal depend on the state of the remote port and whether the local terminal is enabled when the remote connection is active. See ENABLE LOCAL CONSOLE in Chapter 5. Commands from the local and remote ports are called the local and remote command streams.

Another feature of the VAX 8800 is that the secondary CPU can reboot independently of any activity in the primary CPU. A third stream of commands, called the reboot stream, can execute independently of commands entered at either the local or remote console. The command stream is also used to implement several miscellaneous software commands, such as Boot This CPU and Boot Other CPU (see Section 3.10).

The console program implements the three independent command streams by having a separate data base for each stream. The console program alternates between the command streams on a line-by-line basis. Thus, if any stream is executing a long command, such as LOAD/WCS, you may notice a delayed response when executing commands on another command stream.

3.8 Status Monitoring

Before executing each console command, the console program checks for the changes described in the following sections. Status changes are displayed on the local terminal and on the remote terminal, if enabled.

3.8.1 Power Change

Unexpected power changes are reported. This situation is usually created by the EMM commanding a TOTAL OFF to the main circuit breaker. (Note that total power offs are normally preceded by a 5-minute warning period.)

3.8.2 Clock Change

If the clock has stopped due to a STOP ON MICROMATCH (see SET SOMM command in Chapter 5), the clock change is reported.

3.8.3 Remote Port Change

If the Communication Port modem detects a carrier change, which indicates that a remote console has been connected or disconnected, the change is reported.

3.8.4 Halt Messages

Prior to executing a new console command, the console displays pending HALT messages.

3.9 Console Messages

The console program generates informational, warning, and error messages, which include a console "beep" and a date-and-time stamp. These message are listed and explained in Appendix C.

Error messages can be printed while a command line is being parsed, while a command is executing, or while status is being monitored.

3.10 Miscellaneous Operating System Commands

The operating system (or macrodiagnostic) can request different console functions, described in the following sections.

3.10.1 Boot This CPU

If issued by the primary VAX 8800 CPU or the 8700, 8550, or 8500 CPU, the command @DEFBOO.COM is entered into the reboot command stream.

If issued by the secondary VAX 8800 CPU, the command @SECBOO.COM is entered into the reboot command stream.

3.10.2 Boot Other CPU (VAX 8800)

If issued by the primary VAX 8800 CPU or the 8700, 8550, or 8500 CPU, the command @SECBOO.COM is entered into the reboot command stream.

CAUTION

> If the @SECBOO.COM command is issued by the VAX 8700, 8500, or 8550 CPU, the results are unpredictable.

If issued by the secondary VAX 8800 CPU, the command @DEFBOO.COM is entered into the reboot command stream.

3.10.3 UNJAM

The UNJAM signal is asserted and deasserted.

3.10.4 Clear Warm Start Flag

The console maintains a "warm start" flag (see glossary) to prevent repeated attempts to restart a failing operating system. If the warm start flag is not set after the operating system halts, a restart is attempted. If the restart is successful, the warm start flag is cleared. If a restart is attempted and the warm start flag is already set, no further attempt is made and all processing stops.

3.10.5 Clear Cold Start Flag

The console maintains a "cold start" flag (see glossary) to prevent repeated attempts to load and start (bootstrap) an operating system. If the cold start flag is not set after a request to boot the operating system, a boot is attempted. If the boot is successful, the the cold start flag is cleared. If a bootstrap is attempted and the cold start flag is already set, no further action is taken to boot.

3.10.6 Get CPU Information

For the VAX 8800, this command retrieves the following information from the console: LEFT CPU AVAILABLE, RIGHT CPU AVAILABLE, SECONDARY ENABLED.

3.10.7 Toggle Next Primary (VAX 8800)

This command changes the VAX 8800 next primary CPU from its current setting to the alternative (from the left CPU to the right or vice versa).

3.10.8 Write Time Of Day Register

The operating system executes a MTPR TXDB command with the command field set to write TODR data.

3.10.9 Read Time Of Day Register

The operating system executes a MTPR TXDB command with the command field set to read TODR data.

3.10.10 Disable Secondary (VAX 8800)

The console disables the VAX 8800 secondary CPU and clears the SECONDARY_ENABLE state bit.

3.11 Log File

Everything that appears on the console terminal in console mode and OPA0: output from the VAX is saved in a log file, a circular buffer that can store 3000 blocks of output. (On the VAX 8800, each CPU contains a separate log file.) In console mode, input and output are logged. In program mode, input and output are not logged (but are in effect, because the characters echoed by the operating system are logged). To avoid logging your program mode session, use the command SET TERMINAL OPAn: (where n = 4 for local console and n = 5 for remote console) prior to issuing the command SET TERMINAL PROGRAM. To log program mode activity again, use the console command SET TERMINAL OPA0:.

The SHOW LOGFILE command provides a window, which can be moved up and down, into the log file. See Chapter 5 for a description of SHOW LOGFILE.

Chapter 4

Using the Console

This chapter describes how to use the system console, including:

- Setting up the console
- Starting the system
- Starting the system the first time
- Powering up and initializing the system
- Recovering from a power failure
- Restarting after a halt
- Moving between console and program modes
- Adding and removing memory
- Installing the console operating system and applications
- Editing boot command procedures
- Using reserved file names

4.1 Setting Up the Console

Before the VAX system or console can be used, the hardware and software must be installed and tested. Contact your DIGITAL representative to arrange for installation.

4.2 Starting the System

The first time you start up the system following hardware/software installation, follow the start-up procedure described in the next section. After the first time, start the system by pressing the "On" (1) switch on the console's system unit. The console executes the power-up and system-initialization sequence (described in Section 4.4). The first image on your screen is the DIGITAL logo, followed by the VAX (8800, 8700, 8550, or 8500) banner. For instance:

```
                    VAX 8800

  Copyright @ 1986 Digital Equipment Corporation

                    Version 1
                   8-JAN-1986
```

Depending on the outcome of system initialization, the system enters program mode or remains in console mode. If the system enters program mode, the system initialization and boot/restart is successful. If the system stays in console mode, the system initialization failed or the AUTO_BOOT and/or AUTO_RESTART switches are disabled. The successful completion of the START command in the boot file and restart file constitutes successful system initialization.

4.3 Starting the System the First Time

The first time you start up the system, the system manager must install the VAX operating system. Instructions for installing the operating system are in the User's Guide for your system. After the VAX operating system has been installed, modify the boot command files to reflect your system's configuration. You can also set the software-implemented keyswitches and EMM parameter files. The software-implemented switches (described in Section 3.4.1) are disabled, and the system initialization procedure does not complete the first time it powers up. Follow these steps to set up the console to boot the VAX operating system. This procedure assumes that the console software is installed.

1. Turn the console off and then on to automatically execute the SYSINIT.COM file. The first time SYSINIT.COM runs, it exits to console mode (> > > prompt) when it attempts to power up the CPU(s). That happens because the AUTO_POWERON keyswitch is disabled on new systems.

2. Once in console mode, you can enable the keyswitches AUTO_POWERON, AUTO_BOOT, and AUTO_RESTART with the ENABLE command (see Chapter 5).
3. Now exit console mode by entering EXIT. The following message appears:

```
SYSTEM INITIALIZATION NOT COMPLETED
```

   The following files are used to initialize the EMM parameter tables.

   - PL1REG.DAT–Defines voltage limits for MPS regulators.
   - PL2REG.DAT–Defines temperature limits and timeout intervals.
   - DMMREGON.DAT–Mask of monitoring functions to perform POWER ON.
   - DMMREGSB.DAT–Mask of monitoring functions to perform POWER STANDBY.
   - DMMREGOF.DAT–Mask of monitoring functions to perform POWER OFF.

   The VAX is shipped with the following files:

   - PL1REG.DAT
   - PL1REG.850
   - PL2REG.DAT
   - DMMREGON.DAT (2 BI CONFIGURATION)
   - DMMREGON.1BI (1 BI CONFIGURATION)
   - DMMREGSB.DAT
   - DMMREGOF.DAT

4. On the VAX 8800 system with one BI power supply, copy the file DMMREGON.1BI to DMMREGON.DAT. (On the VAX 8700, 8550 or 8500, skip this step.)
5. Edit the boot files for your hardware configuration. See Section 4.10.
6. Enter the P/OS command:

```
$ RUN CONTROL
```

   You see the following warning message:

```
PREVIOUS INITIALIZATION FAILED
```

7. At the console mode prompt, enter the following console commands:

```
>>> POWER ON
>>> @SYSINIT
```

The SYSINIT.COM file executes and automatically creates a memory configuration file, MEMCONFIG.DAT, if one is not found. Also, depending on the keyswitch settings, SYSINIT.COM attempts to boot the VAX.

4.4 Powering Up and Initializing the System

When the VAX is turned on (by turning on the console), power is applied to the power supply control logic, to the Environmental Monitoring Module (EMM), and then to the console. Both the EMM and console administer self-tests to verify their operations as part of their power-up procedures. The EMM also creates an environmental parameter list in its RAM and begins performing a limited number of monitoring tasks, using the list to determine what conditions are considered tolerable.

Once the console and EMM systems have been verified, the console executes SYSINIT.COM. If a system initialization is already in progress, the console executes a POWER OFF command and (provided the power is on) enters console mode. (System initialization is defined as execution of SYSINIT.COM through the successful execution of a START command.) You should then enter the following commands to restart system initialization:

```
>>> POWER ON
>>> @SYSINIT
```

The system initialization procedure then attempts to establish communications with the EMM. When successful, a more extensive list of environmental parameters are loaded into the EMM RAM, providing the EMM with monitoring capabilities specific to the VAX system. If the VAX power is off, the console attempts to turn on the VAX system power according to the programmed sequence in the command file SYSINIT.COM. If the AUTO_POWERON keyswitch is enabled, the power is turned on. If the AUTO_POWERON keyswitch is disabled, SYSINIT exits.

If SYSINIT exits, enter the following commands to continue execution of the system initialization procedures.

```
$POWER ON
$@SYSINIT
```

VAX modules in the CPU backplane are keyed electronically and with the Visibility Bus (VBUS) to insure the modules are installed correctly. Electronic keying uses the microcode and ensures the four modules (CPU, Clock, NBIA and MCL) are in the correct slots. If they are in the wrong slots, the EMM does not operate and the POWER ON command has no effect. VBUS keying ensures the eight CPU modules are in the correct slots. If a module is in the wrong slot, in, the CPU will not operate even with the power on.

The console also tests its interface to the CPU at power-up time. Major data, address, and control paths on the console interface may be looped back to the console processor. These paths can be checked by writing to the interface and reading back the data. The console has access to hardware and software revision-level data from the Visibility Bus, backplane, and console fixed disk. The console collects that data and determines the compatibility of the hardware and software components during the initialization process.

VAX firmware is loaded into the Control Store and RAMs from the console's fixed disk. The console loads the Control Store, Decoder RAMs, and Cache control store. The remainder of the VAX RAMs loaded from the console receive their data with the assistance of microcode running in the CPU. On the VAX 8800, an unbuffered path on the interface to each of the CPUs allows the control store of each CPU to be loaded independently.

The microcode then begins to execute, thus starting the CPU.

Originating on the console are the UNJAM and CPU INIT initialization signals to the Memory Interconnect (NMI) and CPU. The console controls the assertion or deassertion of these signals during initialization and while the machine is running. These commands initialize the CPU to a known state and establish a stable state on the NMI.

The system initialization procedure checks the AUTO_RESTART keyswitch. If it is enabled, the system attempts to restart the VAX operating system. If the AUTO_RESTART switch is disabled or the restart fails, the AUTO_BOOT keyswitch is checked. If it is enabled, a boot is attempted. If the boot fails or the AUTO_BOOT keyswitch is disabled, the system remains halted and in console mode. See Figure 4–1 for a flowchart of this boot/restart procedure.

4.5 Recovering from a Power Failure

After a power failure, the console determines whether the console or the CPU failed.

For a console-only power failure (the console power switch is pushed to the "Off" position or the console is unplugged), the console program restores the console state, which is saved in a log file. The information includes the last known operating state of the CPU(s) (halted or running) and the console mode (console or program). The console interface hardware registers are restored, and the console returns to its last mode.

If the console is recovering from a CPU power failure, the console forces the execution of the SYSINIT.COM file.

4.6 Restarting After a Halt

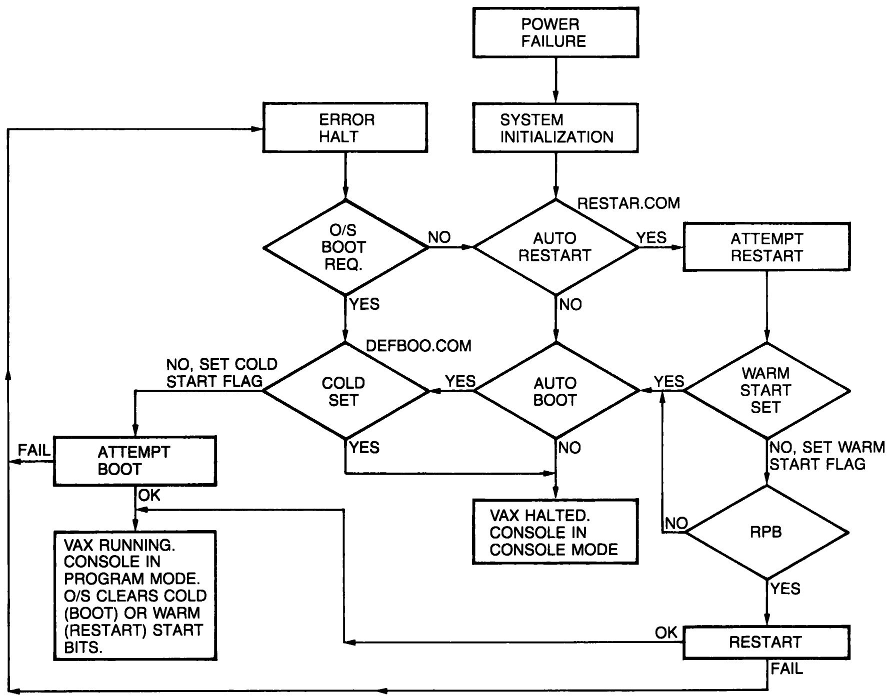
If the operating system requests a reboot following a system halt, the console executes DEFBOO.COM, regardless of the keyswitch settings.

If the operating system does not request a reboot after a system halt, the console executes RESTAR.COM, which does the following:

- Checks the AUTO_RESTART keyswitch. If it is not enabled, the console checks the AUTO_BOOT keyswitch. If it is enabled, the primary CPU is booted (DEFBOO.COM is executed).
- If neither AUTO_RESTART nor AUTO_BOOT is enabled, the console remains halted by exiting RESTAR.COM.
- If AUTO_RESTART is enabled, the restart command procedure (RESTAR.COM) is executed.
- If a restart parameter block is located, RESTAR.COM completes.
- If a restart parameter block is not located and AUTO_BOOT is enabled, a reboot is attempted by executing DEFBOO.COM.
- If the restart parameter block is not located and the AUTO_BOOT keyswitch is disabled, RESTAR.COM exits and leaves the system halted.

Figure 4–1 diagrams these procedures.

Figure 4–1: System Power Failure Recovery or Error Halt Recovery Flow



4.7 Moving Between Console and Program Modes

You can use the VAX console as a console terminal in console mode or as a user terminal interfacing with the VAX operating system in program mode. After the system successfully initializes, the console moves into program mode, from which it can interact with the VAX operating system. To move the console from program mode to console mode, type CTRL/P.

To return to program mode, enter the SET TERMINAL PROGRAM command.

After CTRL/P, you can enter the HALT command to halt the processor and run Console Support Microcode. To move from the halted console mode back to program mode, type CONTINUE.

If you type EXIT to go from console mode to P/OS, type RUN CONTROL to reenter console mode.

NOTE

After entering CTRL/P or SET TERMINAL PROGRAM, wait for the console prompt or the VAX operating system prompt >>> before typing. That is because the type-ahead capability is disabled while the processor changes modes. For instance, if you type three commands, such as:

```
CTRL/P
>>> HALT
>>> SHOW CPU
```

before the console changes from program mode to console mode, the commands do not work as expected, because the console is not in a state to execute the last two commands.

The console ID determines if characters echoed in program mode are logged. The default console ID is OPA0:. All OPA0: data is logged. To change the console ID, use the command SET TERMINAL OPA4 for the local console, or use SET TERMINAL OPA5 for the remote console. Enter the command prior to issuing the SET TERMINAL PROGRAM command.

4.8 Adding and Removing Memory

To add or remove memory from your VAX system, use the following steps:

1. Create a new MEMCONFIG.DAT file with the following steps:

   In console mode, enter the following commands:

   ```
   >>> POWER OFF
   >>> EXIT
   ```

   You are now in P/OS. Change the memory boards.

2. Enter the following command to remove old copies of the memory configuration table:

   ```
   $ DELETE MEMCONFIG.DAT;*
   ```

3. Run the console software, either by using the Main Menu application CONSOLE/PRODCL or the following P/OS command:

   ```
   $ RUN CONTROL
   ```

   You are now in console mode.

4. Enter the following commands:

   ```
   $ POWER ON
   $ @SYSINIT
   ```

The system initialization procedure creates a new MEMCONFIG.DAT file, using the new memory configuration.

4.9 Installing the Console Operating System and Applications

The console operating system (P/OS) and console applications (PRO/DCL, PRO/Communications) are installed on the console's fixed disk at the factory. Under most circumstances, you do not need to reinstall the operating system or applications. However, if you need to install any of them (for example, if you have hardware problems or if there is a console software update), refer to Appendix E. Generally, your service representative can determine whether to reinstall the console operating system and/or any of the applications.

4.10 BOOT Command Procedures

Your VAX system is delivered with a set of templates for boot files, conversational boot files, and files for stopping in the primary bootstrap program (VMB) for debug purposes. These templates are set up for and named after the following devices:

- BCI750
- BDA50
- UDA50

The following naming conventions are used for the templates:

- dddBOO.COM - Boot command procedures
- dddGEN.COM - Conversational boot command procedures
- dddXDT.COM - XDELTA boot command procedures

where: ddd = BCI or BDA or UDA

When your system is installed, these command files contain instructions to deposit zeros in R1, R2, and R3. However, the instructions are commented out so that the command procedure fails if an attempt is made to use it before it is updated. These DEPOSIT commands must be changed to conform to your system's configuration. Changing the deposit to R3 is optional; leaving it commented out lets you provide the unit number in the BOOT command. That makes booting more flexible. The deposit to R5 under certain circumstances must be modified. This command can be commented out if you wish to supply R5 data in the BOOT command.

Use the following chart to determine which bits must be set in each register. The syntax <nn:nn> indicates the bits to modify. For instance, <07:00> means bits 0 through 7. Bits are numbered from left to right in the longword, starting at 0. MBZ means Must Be Zero. The configuration information can be obtained from DIGITAL Field Service.

R0 Boot device type code, hexadecimal radix

<07:00>

17 UDA-50
32 HSC on CI
33 BDA-50
64 Console block storage device

<31:08> 0

R1 Boot device's bus address, hexadecimal radix

<31:06> MBZ
<05:04> indicates which BI, 0-3
<03:00> BI node number of adapter, 0-F

R2 Controller, port number, hexadecimal radix

<31:24> Controller letter designator of boot device (optional).[1] To specify a controller letter of C, 43 (hex), deposit the value 03 in bits <31:24> of R2. In general, subtract %X40 from the hex value of the ASCII controller letter.

UNIBUS:

<23:18> MBZ
<17:00> UNIBUS address of the device's Control and Status Register

CI:

<31:16> MBZ
<15:08> HSC port number (station address) of second HSC if dual ported
<07:00> HSC port number (station address)

BDA:[2]

R3 Boot device unit number. Uses %D notation, decimal radix, for readability

This value can be included in the boot file if it is unlikely to change. If you leave this DEPOSIT command commented out, you can use the BOOT dddnnn command and replace nnn with the unit number from which you wish to boot (BDA, UDA or BCI).

---

1 If your system has similar devices on a BI that use the same device driver, you must ensure that the controller letter is specified in R2 if you wish to boot from any device other than the first one on the BI. If, for instance, you have two BDA devices on the same BI and you wish to boot from the BDA at the higher BI node number, then R2 must contain the controller letter of the BDA in the high-order byte. The value placed in this byte overrides the VMS initialization time assignment of controller letters.

2 A BDA device requires no inputs to R2 other than the optional controller letter designator.

To boot from a Shadow set, the contents of R3 are as follows:

<31:24> 80 Shadow set indicator
<23:16> Shadow unit DUSnn
<15:0> Device in the set to boot

R4 Logical block number to boot from if bit 3 is set in R5, hex

R5 Software boot control flags (see Appendix D), hex

<31:28> Top-level directory number for system disks with multiple systems. To boot stand-alone backup from the system disk, the stand-alone backup files are located in an alternate root. This root number "E" must be deposited into R5.

In addition to the boot files described, you receive a template for DEFBOO.COM, the default boot file. That file is executed when a BOOT command is entered without a device name or when you enable AUTO_BOOT, and the system reboots due to a power failure or error halt.

DEFBOO.COM consists of a single command, a BOOT command. You must modify the device name to refer to the command file you have set up to boot your system. For instance, if you wish to boot from a device on the BCI750 with a unit number of 2, you would modify the DEFBOO.COM BOOT command from:

> > > *BOOT dddnnn*

to:

> > > *BOOT BCI2*

When DEFBOO.COM is executed, it starts the execution of BCIBOO.COM and deposits a 2 in R3. Successful completion depends on the prior editing of BCIBOO.COM. This assumes that you commented out the R3 deposit in BCIBOO.COM and that you have set up R1, R2, and, in this case, R5 (to specify for the system boot) correctly.

4.10.1 Editing a Boot File

To edit the boot files, you must be at the PRO/DCL command level and in the proper directory. (See Section 3.5 for a description of the system directories.) If you are in console command mode, type EXIT to enter PRO/DCL. The $ prompt will appear.

You can make copies of the templates so that they will be available later. To do this, enter the following command, replacing ddd with a valid device mnemonic as specified above for each file you wish to save:

$ COPY dddBOO.COM dddBOO.SAV

After you have "saved" the templates, you can edit the command files. To do this, use the EDT editor provided with P/OS.

$ RUN EDT
EDT> dddBOO.COM

The P/OS EDT editor automatically enters command mode, where you have both keypads (main and auxiliary) available for editing purposes. Refer to Figures 4–2 and 4–3 for illustrations of the LK201 keyboards. This editor is a subset of the VMS EDT editor. Refer to the VMS EDT editor documentation for more information. To obtain on-line help for EDT, press the PF2 key after you have started the editor.

4.10.1.1 Keyboard Editing Functions

The following basic keyboard editing functions allow you to edit the command files. Except for CTRL/Z and the ⌫ key, the keys are on the two auxiliary keypads located to the right of your keyboard.

Table 4–1: Keyboard Editing Functions

Key	Function
[↓]	Moves the cursor down one line
[↑]	Moves the cursor up one line
[→]	Moves the cursor one character to the right
[←]	Moves one character to the left
[,]	Deletes the character under the cursor

Table 4–1 (Cont.): Keyboard Editing Functions

Key	Function
<X]	Deletes the character to the left of the cursor
[PF1] [,]	Restores the last character deleted with [,]
[-]	Deletes a word (place the cursor on the leftmost character)
[PF1] [-]	Restores the last word deleted with [-]
[PF4]	Deletes a line
[PF1] [PF4]	Restores the last line deleted with [PF4]

- Insert character(s) at the location of the cursor by typing the desired character(s) on the main keyboard keys.
- Press CTRL/Z to exit the editing command mode. An asterisk "*" appears at the bottom left corner of the screen.

  Type EXIT to exit EDT and make a new file containing your changes.

  Type QUIT at the asterisk if you wish to leave the edit session without creating a new file that contains the changes.

A typical editing session requires using the arrow keys to move the cursor between lines (vertically) and along lines (horizontally). The delete character keys are useful for replacing the data portion of the DEPOSIT command.

Following are descriptions of more advanced keypad keys:

Table 4–2: Advanced Keypad Functions

Key	Function
[4]	Sets the direction of cursor movement to forward (toward the end of the file).
[5]	Sets the direction of cursor movement to backward (toward the beginning of the file).
[0]	Moves the cursor one line in the previously set direction and positions the cursor at the left margin.

The following is an editing sequence that uses the rightmost keypad to delete a character:

1. Press 4 to set the direction forward.
2. Press 0 as many times as it takes to move to the line to be changed.
3. Press 1 as many times as it takes to move to the word to be changed.
4. Press the right arrow as many times as it takes to move to the character to be changed.
5. Press the comma (,) to delete the character.
6. You may then type the character on the main keyboard you wish to add and make any other necessary changes.
7. Finish the editing session by pressing CTRL/Z. At the "*" prompt, type EXIT.
8. When you have finished editing the boot files, enter:

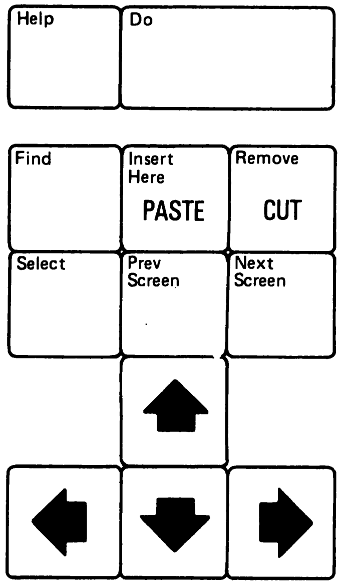
   $ RUN CONTROL

   to enter console mode. Type EXIT to return to the P/OS Main Menu.

If you are editing the files for the first-time start-up procedure, continue with the instructions in Section 4.3.

Figure 4–2: Keypad Editing Keys: LK201 Keyboard

Figure 4–3: Additional Keypad Editing Keys: LK201 Keyboard



4.11 Reserved File Names

The following file names are reserved for the system and are listed alphabetically with brief definitions. Do not rename these files or use these names for your files.

BCIBOO.COM	CI750 boot command procedure
BCIGEN.COM	CI750 conversational boot command procedure
BCIXDT.COM	CI750 debug boot command procedure

BDABOO.COM	BDA50 boot command procedure
BDAGEN.COM	BDA50 conversational boot command procedure
BDAXDT.COM	UDA50 debug boot command procedure
CI780.BIN	CI microcode version 7.0
CI780V50.BIN	CI microcode version 5.0
CI780V60.BIN	CI microcode version 6.0
CCODE.BIN	Cache control store contents
CONTROL.TSK	Main console control program
CRASH.COM	Command procedure used to crash the VAX to get a dump
CSBOO.COM	Stand-alone backup boot command procedure
CSGEN.COM	Stand-alone backup conversational boot command procedure
CSXDT.COM	Stand-alone backup debug boot command procedure
SCIBOO.COM	Diagnostic Supervisor BCI750 boot command procedure
SDABOO.COM	Diagnostic Supervisor BDA50 boot command procedure
DEFBOO.COM	Default boot command procedure
DIABOO.COM	Diagnostic Supervisor boot command procedure from console disk
DMMREGOF.DAT	Default Mode Mask when power is OFF
DMMREGON.DAT	Default Mode Mask when power is ON
DMMREGON.1BI	Default Mode Mask for a 1BI system when power is ON
DMMREGSB.DAT	Default Mode Mask when power is in STANDBY
DRAM.BIN	Ibox decoder RAM contents
SUABOO.COM	Diagnostic Supervisor UDA50 boot command procedure
EXIT.COM	Command procedure used to exit another command procedure
EZKPA.TSK	Power-up microdiagnostics
EZKPB.BIN	Power-up microdiagnostics
EZKPB.WC1	Power-up microdiagnostics

EZ%%.BIN	Licensed microdiagnostics - optional
EZK%%.TSK	Licensed microdiagnostics - optional
EZK%%.WC1	Licensed microdiagnostics - optional
*.HLP	Help files
LBS.TSK	Logical Block Server program
LOGFILE.DAT	Console logfile (journal file)
MEMCONFIG.DAT	Memory array card to physical address space
MICEXIT.COM	Return to console mode from MICMON mode
NMIRESET.COM	Executed after NMI Reset signal
PL1REG.DAT	Max/min voltage levels for monitoring
PL2REG.DAT	Automatic Shutdown parameters (red/yellow temperature limits)
RESTAR.COM	Restart command procedure
REVHIST.DAT	Definition of compatible revs mapping
RT.TSK	RTI device driver
SDFDEF.BIN	Slow Data File, Constants
SECBOO.COM	Secondary boot command procedure
SYSINIT.COM	System initialization and startup command procedure
UDABOO.COM	UDA50 boot command procedure
UDAGEN.COM	UDA50 conversational boot command procedure
UDAXDT.COM	UDA50 debug boot command procedure
UCODE.BIN	ISP WCS contents
VMB.EXE	Primary bootstrap program
XKDRV.TSK	COMM port device driver

Chapter 5

Console Commands

The Console Command Language (CCL) is the set of commands you use to control and monitor the VAX from the console. This chapter describes each command, including its structure, syntax, parameters, qualifiers, options, and restrictions. Examples of each command are also provided.

5.1 Commands

A command is usually a verb that describes the action the console program is to take. For example, typing DISABLE PRINTER stops all console output to the printer.

Commands must be contained on one line and can be a maximum of 120 characters. Adjacent spaces and tabs are equivalent to a single space or tab.

5.1.1 Parameters

Parameters are always preceded by a space. They describe the object of the command, which can be a file, an address, a bit offset, a command, a count, or some data. For instance, the following MICROSTEP command parameter specifies the number of microcycles to step:

>>> *MICROSTEP 6*

If a command parameter is a file name, the syntax is always in Files–11 format:

filename.extension;version

where:

filename = any name up to 9 characters
extension = a 3-letter extension to the file name
version = number assigned by the Files–11 file system

Allowable characters are A through Z and 0 to 9.

Device names are not allowed. Any file name specified must reside on the fixed disk, DW1:. Files may reside in the following directories:

VAX 8800 - [CONSOLE] or [8800]
VAX 8700 - [CONSOLE] or [8700]
VAX 8550 - [CONSOLE] or [8550]
VAX 8500 - [CONSOLE] or [8500]

You cannot specify a directory as part of a file name.

5.1.2 Options

An option is a keyword used with a command. For example, AUTO RESTART and AUTO BOOT are options used with the ENABLE command. You are usually required to specify one option.

5.1.3 Qualifiers

A qualifier is a command argument preceded by a slash (/) that causes the command to perform a more specific action. For example, the /NEXT=n qualifier used with the DEPOSIT command causes the command to repeat 10 times at successively higher addresses:

```
DEPOSIT/NEXT=10 address data
```

Qualifiers are used with commands and parameters. You may leave a space between the command (or parameter) and the qualifier but not between the slash (/) and the qualifier.

5.1.4 Defaults

A default is a value automatically used by the system when you do not enter another value. Commands have defaults that take effect when the console application is installed. Defaults are either reinitialized each time the console is powered up or retained across power failures. Some defaults are initialized each time a new command is invoked. Defaults and qualifiers for commands are included in their descriptions.

5.1.5 Command Abbreviations

Commands and qualifiers can be abbreviated as long as the abbreviation contains enough characters to identify it uniquely. The briefest command form appears in Appendix A in bold lettering.

5.1.6 Command Terminator

Any command can be terminated before execution with CTRL/U.

5.2 Syntax Error Reporting

The console reports syntax errors by displaying the command line just read (either from the terminal or command file) and the words "INVALID COMMAND SYNTAX". Error messages are described in Appendix C.

5.3 The Command Language

The following tables describe how symbols, prompts, and special keys are used in the command language.

5.3.1 Symbols

Table 5–1: Command Language Symbols

Symbol	Definition
/	A slash indicates a command qualifier. No space is allowed between the slash and the qualifier.
+	A plus sign denotes a default address, which is the last address plus the current data length in bytes.
-	A minus sign denotes a location just before (one less than) the address previously referenced.
@	When used in place of an address or data argument, the at sign indicates that the data returned from the last reference to an address should be used as the current address or data. The at sign is also used to execute command procedures. When used as a data argument, the data of the previous EXAMINE or DEPOSIT command is copied into the specified address. (See descriptions of the EXAMINE and DEPOSIT commands for more information.)
* or .	When used as an address argument, the asterisk or period indicates that the last address referenced is to be used again.
,	The comma separates items in a list.
= or :	The equal sign or colon separates a qualifier from an associated argument. For example, /NEXT=n or /NEXT:n may be used.

5.3.2 Prompts

The following are the command prompts for various modes:

Table 5–2: Command Language Prompts

Prompt	Definition
> > >	Console command input prompt
MIC>	Micromonitor input prompt
< < <	Link mode input prompt (see LINK and PERFORM commands)

5.3.3 Special Keys

This section describes keyboard option keys and indicator LEDs for the VAX Console Command Language.

Table 5–3: Command Language Special Keys

Key	Definition
CTRL/C	Aborts activity in progress.
CTRL/S	Stops the screen from scrolling. See CTRL/Q.
CTRL/Q	Allows the screen to resume scrolling. See CTRL/S.
CTRL/U	Aborts input without executing it. On the remote port, CTRL/U must be followed by Return in order to obtain a new prompt.
CTRL/Z	Terminates and executes the command line. (Functionally equivalent to pressing Return.)
<X]	Deletes the character to the left of the cursor and moves the cursor one space to the left.
Hold Screen	Freezes a screen display so you can read it. Press the key again to release the hold. When Hold Screen is in effect, its indicator light (top of the keyboard) stays on.
Print Screen	The Print Screen key causes the current image on the screen to be printed. Certain types of graphics images cannot be printed with Print Screen. Print Screen also executes a Hold Screen (see Caution below). Print Screen is disabled when the printer has been enabled with the ENABLE PRINTER command.
Return	The Return key terminates and executes a command line.
Set-Up	Pressing Set-Up unlocks a locked keyboard. If nothing happens when you type on the keyboard, check to see if the Wait light at the top of the keyboard is lit. If the Wait light stays on for more than 15 seconds, the keyboard is probably locked. You may be able to unlock it by pressing the Set-Up key once. Some applications intentionally lock the keyboard for brief periods, and you need to wait. Pressing Set-Up may affect the program running on the VAX.

CAUTION

Hold Screen causes program mode output to be lost. To prevent this, use CTRL/S to hold the screen and CTRL/Q to release it.

5.4 Console Command Descriptions

The remainder of this chapter describes each console command in detail. The commands are ordered alphabetically. Parameters, options, and qualifiers associated with the command are described. Micromonitor commands, which enable you to use the diagnostics program, are described in Chapter 6.

On the VAX 8800, some commands apply only to the CPU whose output is on display, while others affect both CPUs. The scope of multi-CPU and single-CPU commands is specified by SET CPU, which is described in this chapter. The description of each command indicates the CPU to which the command applies. See Section 3.2 for more information.

Multiple CPU commands (those referencing Primary, Secondary, Left, and Right CPUs) do not apply to the VAX 8700, 8550, or 8500.

5.5 Help

Information about each command can be accessed on-line by entering the following at the console (> > >) prompt:

> > > *HELP command-name*

To get a listing of HELP topics, just enter HELP.

@

Executes console commands contained in a command procedure.

Format **@file-name**

Description

The command procedure specified by the parameter file name is opened. Records in the file are processed as sequential console commands. When @ is invoked from a command file, the original command file is closed and the new (specified) file opened. When the new file terminates, processing stops. (Execution does not return to the previous file.) This command allows chaining of one command procedure to another, but not nesting of command procedures inside each other.

If VERIFY mode has been set, the console displays each line as it is read from the command file, as well as the response to each command as it is executed. If VERIFY mode has not been set, lines from the file are not echoed. (See SET VERIFY and SET NOVERIFY command descriptions.)

To terminate execution of a command file, press CTRL/C. Control returns to the terminal.

Command Parameter

file-name

The file name specifies the command file to be opened. The file can be any valid FILES-11 file specification that identifies a sequential ASCII file. If the file extension is not supplied, .COM is assumed.

Examples

1.
```
>>> SET VERIFY
>>> @SYSINIT
```

2.
```
>>> If not $AUTO_POWERON then @EXIT
>>> POWER ON
.
.
.
```

3.
```
>>> @EXAMINE.COM
  EXAMINE/P/L  20080000/N=20
  EXAMINE/G    9
```

BOOT

Executes a command procedure that starts the VAX operating system software.

Format

BOOT *[device-name]*

Command Qualifiers	**Defaults**
/R5 : data	None

Description

BOOT executes the command file device-nameBOO.COM. If no device name is specified, the file DEFBOO.COM is used.

Command Parameter

device-name
Specifies the name of the device from which the operating system is to be loaded and run. If no device name is specified, the default device is booted as specified in the default boot command procedure DEFBOO.COM. Device-name uses the syntax dddnnn where:

ddd is a 3-letter device mnemonic (valid devices are BCI, BDA, UDA).

nnn is an optional 1- to 3-digit unit number.

The unit number is picked up from the parameter device-name by the console and deposited into R3 prior to execution of the boot file device-nameBOO.COM. Any deposit to R3 contained in the boot file overrides nnn in device-name. Therefore, the DEPOSIT R3 command in the boot command file can be commented out if greater flexibility is desired in choosing a unit to boot.

Command Qualifiers

/R5 : data
Deposits the specified numeric literal in R5 before the command file is processed. This mechanism passes information to the primary bootstrap program (VMB). (See Appendix D for valid data.) If the /R5 : data qualifier is used, the DEPOSIT R5 command in the boot command file must be removed or commented out.

Examples

1. `>>> BOOT BCI2`

   Executes BCIBOO and boots from unit 2.

2. `>>> B/R5:1`

   Executes DEFBOO.COM and stops in SYSBOOT.

CLEAR

Restores settings made by other commands to their former state.

Format

CLEAR *(option)*

Description

The CLEAR command options are described as individual commands.

Command Options

ACCUMULATOR

RESTART_FLAGS

SCREEN

SOMM

TOMM

CLEAR ACCUMULATOR

Clears the accumulator set by the PROBE VBUS command. The accumulator can be displayed with the SHOW ACCUMULATOR command.

Format

CLEAR ACCUMULATOR

Examples

1. `>>> CL A`

2. `>>> CLEAR ACCUMULATOR`

CLEAR RESTART_FLAGS

Clears both the WARM_RESTART and COLD_RESTART flags.

Format

CLEAR RESTART_FLAGS

Description

The COLD_RESTART flag is set by the FIND/MEM command, which is used in the BOOT command procedure. The WARM_RESTART flag is set by the FIND/RPB command, which is used in the RESTAR.COM command procedure. The VAX operating system clears the COLD_RESTART and WARM_RESTART flags following a successful boot or restart. If the boot or restart fails, these flags remain set to prevent automatic boot and restart procedures from looping indefinitely. The CLEAR RESTART_FLAGS command can be used to manually clear these flags so that a subsequent BOOT command will execute. It is used as the final command in the SYSINIT.COM procedure.

Restriction

Use caution when including CLEAR RESTART_FLAGS inside command files. Under certain error conditions, the command files may execute indefinitely.

Examples

1. `>>> CLEAR RESTART_FLAGS`

CLEAR SCREEN

Resets the VAX console terminal from console mode.

Format

CLEAR SCREEN

Examples

1. ```
   >>> CL SC
   ```

2. ```
   >>> CLEAR SCREEN
   ```

CLEAR SOMM

Clears the "Stop on Micromatch" flag.

Format

CLEAR SOMM

Description

See the description of the SET SOMM command.

Examples

1. `>>> CLEAR SOMM`

2. `>>> CL S`

CLEAR TOMM

Clears the "Trap on Micromatch" flag.

Format

CLEAR TOMM

Description

See the description of the SET TOMM command.

Examples

1. `>>> CLEAR TOMM`

2. `>>> CL T`

CONTINUE

Resumes execution of a halted VAX instruction program.

Format

CONTINUE

Description

CONTINUE starts macrocode execution at the address in the macro program counter (PC). Processor initialization is not performed, and the CPU's state is changed from Halted to Running. The console changes from console mode to program mode. If output has been halted by CTRL/P or the HALT command, the CONTINUE command sends CTRL/Q (XON) to the VAX on OPA0:, OPA4:, or OPA5: (depending on the most recent SET TERMINAL PROGRAM command), and output resumes.

CONTINUE has a different meaning in Micromonitor mode. The Micromonitor commands are described in Chapter 6.

Examples

1. `>>> C`

2. `>>> CONTINUE`

DEPOSIT

Replaces the contents of a specified location or locations.

Format

DEPOSIT *address-identifier deposit-data*

Command Qualifiers	Defaults
/NEXT	1
address-space-qualifiers	(PHYSICAL)
/CACHE	
/CONTROLSTORE or /WCS	
/GPR	
/IPR	
/PHYSICAL	
/SDF	
/TB	
/TEMP	
/VIRTUAL	
size-qualifiers	(LONGWORD)
/BYTE	
/LONGWORD	
/QUADWORD	
/WORD	

Description

The DEPOSIT command can replace the contents of a location(s) in memory, CPU registers, cache, translation buffer, control stores, IPRs, GPRs, or I/O space.

Defaults

If no data size or address space qualifiers are specified, the defaults are established by the most recent EXAMINE, DEPOSIT, or SET DEFAULT command. (Note that the EXAMINE command does not change the default address space when examining the PLS or the UPC.)

Initially, the default is LONGWORD data size, PHYSICAL address space, address identifier 0, and radix hexadecimal.

Restrictions

- The CPU must be halted.
- After the WCS is loaded (see the /WCS qualifier description), you cannot deposit (or examine) the Slow Data File (SDF).
- You cannot use the /GPR or /IPR qualifier when specifying a symbolic address.
- The size qualifiers /BYTE, /WORD, and /QUADWORD are valid only with physical and virtual address spaces. However, if the size qualifiers are used with any other address spaces, the qualifiers are ignored and /LONGWORD is used instead. The default size changes to /LONGWORD.

Command Parameters

address-identifier
Specifies the starting numeric or symbolic address of a location or series of locations whose contents are to be replaced with deposit-data. The numeric address is a numeric literal. (See Appendix B.)

deposit-data
A numeric literal that defines the data to be deposited into the specified location(s). If deposit-data is specified as @, the last data examined is used as the deposit data. (See the description of the EXAMINE command.) The DEPOSIT command writes the data in consecutive locations, beginning with the address specified. (See Appendix B.)

Command Qualifiers

/NEXT
Causes the DEPOSIT command to be repeated n times, with successively higher addresses. The amount of address increment/decrement depends on the size qualifier (see below). For GPRs and IPRs, the address increments by one. The same data is deposited into each location.

The DEPOSIT command continues either until the command has been repeated n times or until address overflow occurs. Address overflow occurs when the GPR number exceeds 15(decimal) or the IPR number exceeds 255(decimal).

address-space-qualifiers
The following qualifiers may be used on any DEPOSIT address to override the default address space setting.

/CACHE
Cache data store.

/CONTROLSTORE or /WCS
The address is to be interpreted as a Writable Control Store (WCS) address.

/GPR
/GPR indicates that the address is to be interpreted as the number of a VAX General Purpose Register (GPR). The GPR number must be between 0 and 15(decimal). If /NEXT=n is specified and the GPR number exceeds 15, the command terminates with an address overflow error.

/IPR
/IPR specifies that the address is to be interpreted as the number of an Internal Processor Register (IPR). The IPR number must be in the range 0 to 255 (decimal). If the /NEXT=n qualifier is specified, and the IPR number exceeds 255, the command terminates with an address overflow error.

/PHYSICAL
The address space is to be in physical memory or physical I/O space.

/SDF
Slow data file.

/TB
Translation buffer.

/TEMP
IPRs known only to microcode, not macrocode. See Appendix B. Only numeric literals are valid with the /TEMP qualifier.

/VIRTUAL
Interprets the address as a virtual address. Access and protection checking occurs. If access is not allowed, the console issues an error message. If memory management is not enabled, virtual addresses are equal to physical addresses.

size-qualifier
Use the following qualifiers to specify data size. The initial default data size is LONGWORD. The qualifiers /BYTE, /WORD, and /QUADWORD are valid only on /PHYSICAL and /VIRTUAL address spaces. If /B, /W, and /Q size qualifiers are used with any other address spaces, they are treated as /LONGWORD and the default is changed to LONGWORD.

/BYTE
References a 1-byte quantity.

/WORD
References a 2-byte quantity.

/LONGWORD
References a 4-byte quantity.

/QUADWORD
References an 8-byte quantity.

Examples

1. `>>> DEPOSIT/PHYSICAL 51B8 50001D0`

   Deposits the value 50001D0 at physical address 51B8.

2. `>>> D/BYTE B197 15`

   Deposits a byte of data, changes default size to byte.

3. `>>> D/GPR 0  0`

   Deposits 0 to GPR 0.

4. `>>> D/IPR 85 1`

   Deposits a 1 to IPR 85.

5. `>>> DEP/TEMP 4E 0`

   Deposits the value to the temp location 4E.

6. `>>> D/VIRTUAL/LONGWORD 61B7 70002D1`

7.
```
>>> DEP/T 4E 0
>>> EX/L/P 20080000/n:8
P 20080000 0000B810
P 20080004 0000E084
.
.
.
P 20080020 00000000
>>> DEP + C084
```

   Sets the defaults to LONGWORD and PHYSICAL then examine 9 longwords.

   Deposits values into 20080000 and the next longword.

8. `>>> DEP UPC 3710`

   Deposits the value 3710 in the micro-PC.

9. `>>> DEP PSL 1F0000`

   Deposits the value 1F0000 in the PSL.

10. `>>> DEP @ @`

    Deposits the data from the last EX/DEP at address 1F0000. Demonstrates the use of the @ symbol.

11.
```
>>> EX 1 F0000
 P 001F0000 001F0000
```

DISABLE

Prevents specified console options from operating.

Format

DISABLE *(option)*

Description

Values established for console options by the DISABLE command are retained across power failures, restarts, and reboots. DISABLE negates the effect of the ENABLE command (described in this chapter). The DISABLE command options are described as individual commands.

Command Options

AUTO BOOT

AUTO POWERON

AUTO RESTART

LOCAL CONSOLE

PRINTER

REMOTE CONSOLE

REMOTE MODEM

DISABLE

REMOTE MONITORING

REMOTE USER

SECONDARY (VAX 8800)

DISABLE AUTO BOOT

Suppresses automatic booting following recovery from a VAX power failure or error halt. A restart may be attempted, depending on the AUTO RESTART setting.

Format

DISABLE AUTO BOOT

Examples

1. `>>> DIS A B`

2. `>>> DIS AUTO BOOT`

DISABLE AUTO POWERON

Suppresses automatic application of logic power following recovery from a VAX power failure. The VAX logic remains off under these circumstances.

Format

DISABLE AUTO POWERON

Examples

1. `>>> DIS A P`

2. `>>> DISABLE AUTO POWERON`

DISABLE AUTO RESTART

Suppresses automatic restart of the operating system following recovery from a VAX power failure or error halt.

Format

DISABLE AUTO RESTART

Restrictions

If AUTO BOOT is enabled, a boot is attempted. If AUTO BOOT is disabled, the VAX remains halted.

Examples

1. `>>> DIS A R`

2. `>>> DISABLE AUTO RESTART`

DISABLE LOCAL CONSOLE

Prevents commands that change the state of the VAX from being processed by the local console terminal during a remote diagnostic session.

Format

DISABLE LOCAL CONSOLE

Description

DISABLE LOCAL CONSOLE can be entered by either the remote or local console operator. It negates the effect of the ENABLE LOCAL CONSOLE command. The command is typically used after the local operator is finished with state-changing commands.

DISABLE LOCAL CONSOLE issues an informational message on the local and remote terminals.

This command is retained across power failures.

Default

The local console is initially enabled.

Examples

1. `>>>DI L C`

2. `>>>DISABLE LOCAL CONSOLE`

DISABLE PRINTER

Stops console output to the printer.

Format

DISABLE PRINTER

Restriction

This command is rejected if entered from the remote console.

Examples

1. `>>> DIS P`

2. `>>> DISABLE PRINTER`

DISABLE REMOTE CONSOLE

Disables the remote port as a remote console.

Format

DISABLE REMOTE CONSOLE

Description

The following occurs if the remote port is in console mode when DISABLE REMOTE CONSOLE is issued:

- The warning message "SHUTTING DOWN REMOTE PORT" is printed on the remote port.
- The phone line is disconnected and reconnected.
- The remote port is forced to program I/O mode, and a CTRL/P character received from the remote port is not processed by the VAX operating system.

The remote port can still function as a user terminal (depending on the state of REMOTE USER).

Examples

1. `>>> DIS R C`

2. `>>> DISABLE REMOTE CONSOLE`

DISABLE REMOTE MODEM

Sets the Communications Port to NO MODEM. This is required to hard-wire a RS232 line to the Communications Port.

Format

DISABLE REMOTE MODEM

Examples

1. `>>> DIS R M`

2. `>>> DISABLE REMOTE MODEM`

DISABLE REMOTE MONITORING

Turns off the monitoring of remote port activity on the local terminal.

Format

DISABLE REMOTE MONITORING

Restriction

This command is rejected if entered from the remote console.

Examples

1. `>>> DIS R MON`

2. `>>> DISABLE REMOTE MONITORING`

DISABLE REMOTE USER

Prevents access to the remote port.

Format

DISABLE REMOTE USER

Description

When this command completes, the remote port does not respond to any character sequences. This is the most secure state of the remote port.

If the remote port is connected, the warning message "SHUTTING DOWN REMOTE PORT" is printed on the remote port and the phone line is hung up. If the remote console is enabled, it is disabled by this command.

Examples

1. `>>> DIS R U`

2. `>>> DISABLE REMOTE USER`

DISABLE SECONDARY

Prevents the secondary CPU from booting. This command applies only to the VAX 8800.

Format

DISABLE SECONDARY

Description

When the secondary CPU is disabled, encoded messages from the primary CPU to boot the secondary CPU produce no response.

The secondary CPU remains disabled across power failures, reboots, and restarts until enabled with ENABLE SECONDARY console command.

Examples

1. `>>>DIS S`

2. `>>>DISABLE SECONDARY`

ENABLE

Enables console options to become operational.

Format

ENABLE *(option)*

Description

Console parameters established by the ENABLE command are retained across power failures, restarts, and reboots. The ENABLE command negates the effect of the DISABLE command settings (described in this chapter). The ENABLE command options are described as individual commands.

Command Options

AUTO BOOT

AUTO POWERON

AUTO RESTART

LOCAL CONSOLE

PRINTER

REMOTE CONSOLE

REMOTE MODEM

ENABLE

REMOTE MONITORING

REMOTE USER

SECONDARY (VAX 8800)

ENABLE AUTO BOOT

Allows the VAX to boot automatically following system initialization, a power failure, or an error halt. This command applies if the restart attempt fails or AUTO RESTART is disabled.

Format

ENABLE AUTO BOOT

Description

ENABLE AUTO BOOT sets the state bit AUTO_BOOT to true. As an example, when this state bit is true, the following console command executes the command procedure DEFBOO.COM to reboot the system following recovery from a power failure or error halt:

```
>>> IF AUTO_BOOT THEN @DEFBOO
```

- If AUTO RESTART is enabled (see description of the ENABLE /DISABLE AUTO RESTART command), a restart is attempted.
- If AUTO RESTART fails and AUTO BOOT is enabled, a reboot is attempted.
- If AUTO RESTART is not enabled but AUTO BOOT is, then a reboot is attempted.
- If the reboot fails, the VAX remains halted.

See Figure 4–1 for more information.

AUTO_BOOT is disabled on new systems.

Examples

1.
```
>>> EN A B
```

2.
```
>>> ENABLE AUTO BOOT
```

ENABLE AUTO POWERON

Allows logic power to be applied automatically following recovery from a VAX power failure.

Format

ENABLE AUTO POWERON

Description

ENABLE AUTO POWERON sets the state bit AUTO_POWERON to true. As an example, when this state bit is true, the following console commands execute the POWERON command rather than EXIT and leaves the system in a powered-off state.

```
>>> IF NOT AUTO_POWERON THEN @EXIT
>>> POWER ON
```

AUTO_POWERON is initially disabled on new systems.

Examples

1. `>>> EN A P`

2. `>>> ENABLE AUTO POWERON`

ENABLE AUTO RESTART

Allows the VAX to be automatically restarted following a power failure or error halt.

Format

ENABLE AUTO RESTART

Description

ENABLE AUTO RESTART sets the state bit AUTO_RESTART to true. As an example, when this state bit is true, the following console command executes the command procedure RESTAR.COM to recover from the power failure or error halt:

```
>>> IF AUTO_RESTART THEN @RESTAR
```

See Figure 4–1 for more information.

AUTO RESTART is initially disabled on new systems.

Restrictions

If the AUTO RESTART attempt fails (or AUTO RESTART is disabled), an AUTO BOOT is attempted, provided AUTO BOOT is enabled (see AUTO BOOT).

Examples

1. `>>>EN A R`

2. `>>>ENABLE AUTO RESTART`

ENABLE LOCAL CONSOLE

Allows commands that change the state of the VAX to be processed by the local terminal, whether or not a remote diagnostic session is in progress.

Format

ENABLE LOCAL CONSOLE

Description

Commands that change the state of the VAX hardware are normally disallowed on the local terminal while the remote terminal is active. That prevents the local operator from changing the hardware state while a remote operator is diagnosing the system. The remote operator issues ENABLE LOCAL CONSOLE in order for the local operator to issue command(s) that change the VAX hardware state.

ENABLE LOCAL CONSOLE issues an informational message on the local and remote terminals.

ENABLE LOCAL CONSOLE is retained across power failures.

Default

The local console is initially enabled.

Restriction

ENABLE LOCAL CONSOLE is not accepted from the local terminal. An error message is printed. Note, however, that the DISABLE REMOTE USER and DISABLE REMOTE CONSOLE commands implicitly set ENABLE LOCAL CONSOLE.

Examples

1. `>>> EN L C`

2. `>>> ENABLE LOCAL CONSOLE`

ENABLE PRINTER

Permits character output being written to the log file to also be printed on the printer.

Format **ENABLE PRINTER**

Description

Causes console mode output from both the local and remote terminals, as well as OPA0: output from the CPU(s) to be printed.

ENABLE PRINTER allows the hard-copy output of local (OPA0) console events and VMS messages to be sent to the operator's console (OPA0).

Restriction

If the printer is off line for 10 seconds, output is lost.

Examples

1. `>>> EN P`

2. `>>> ENABLE PRINTER`

ENABLE REMOTE CONSOLE

Allows the remote port to be used as a remote console.

Format

ENABLE REMOTE CONSOLE

Description

When REMOTE CONSOLE is enabled, the remote port is also enabled as a remote user terminal. Many commands from the local port are then disabled, such as those that affect the CPU clocks, halted CPUs, and so on.

Restriction

When the remote port is connected through a modem, most local commands are disabled. Local users may experience a delay in command execution.

Examples

1. `>>> EN R C`

2. `>>> ENABLE REMOTE CONSOLE`

ENABLE REMOTE MODEM

Sets the Communications Port characteristics for a modem.

Format

ENABLE REMOTE MODEM

Description

The Communications Port can be used for remote diagnosis by DIGITAL's Diagnostic Center. For this service, a modem is connected to the console's Communications Port and a phone line. The remote modem is enabled, and a remote field service center can dial into the console and run diagnostics.

Examples

1. `>>> EN R M`

2. `>>> ENABLE REMOTE MODEM`

ENABLE REMOTE MONITORING

Causes characters printed on the remote port to be monitored locally.

Format

ENABLE REMOTE MONITORING

Description

The local terminal screen displays all characters being displayed on the remote port. The remote port may be acting as a remote console or remote user terminal.

Examples

1. `>>> EN R MON`

2. `>>> ENABLE REMOTE MONITORING`

ENABLE REMOTE USER

Allows the remote port to communicate with the VAX as a user terminal.

Format

ENABLE REMOTE USER

Description

The ENABLE REMOTE USER command entered at the local console terminal enables a user at the remote console terminal to interact with the VAX operating system as though from a user terminal. The remote port may not be used as a console if the remote console is disabled.

Examples

1. `>>>EN R U`

2. `>>>ENABLE REMOTE USER`

ENABLE SECONDARY

Enables the secondary CPU to be booted. This command applies to the VAX 8800 only.

Format

ENABLE SECONDARY

Description

ENABLE SECONDARY allows the console to act on encoded messages from the VAX to boot the secondary CPU.

The secondary CPU is enabled by default and can be disabled with the DISABLE SECONDARY command.

Examples

1. `>>>EN S`

2. `>>>ENABLE SECONDARY`

EXAMINE

Displays the contents of a specified address.

Format

EXAMINE *address-identifier*

Command Qualifiers	**Defaults**
/NEXT=value	1
address-space-qualifiers	PHYSICAL
/GPR	
/IPR	
/PHYSICAL	
/SDF	
/TEMP	
/VIRTUAL	
size-qualifiers	LONGWORD
/BYTE	
/LONGWORD	
/QUADWORD	
/WORD	

Description

When the EXAMINE command is executed, it displays the specified address and its contents as follows:

address-space-indicator address contents-of-address

The address-space-indicator displays the following:

- G - General Processor Register
- I - Internal Processor Register
- M - Machine-dependent space (PSL,SDF,TEMP)

- P - Physical memory: When examining virtual memory, the address space and address in the response are translated to physical addresses. The EXAMINE command normally changes the default address space, except when examining the PSL or UPC.

The address is displayed in hexadecimal, right justified, and zero filled to the actual address magnitude. The contents-of-address is displayed in hexadecimal, regardless of default settings.

Defaults

The address-space and size qualifiers default to the values or radixes used in the previous EXAMINE, DEPOSIT, or SET DEFAULT command. Initially, the default is LONGWORD data size, PHYSICAL address space, address identifier 0, and radix hexadecimal. The size qualifiers /BYTE, /WORD, and /QUADWORD used with any address-space qualifier other than /PHYSICAL or /VIRTUAL reset the size-qualifier to LONGWORD.

Restrictions

- The CPU must be halted.
- Do not use the /GPR or /IPR qualifiers when using a symbolic register address. (See Appendix B for the symbolic addresses.)
- After the WCS is loaded, you cannot examine (or deposit) the Slow Data File (SDF).
- The /BYTE, /QUADWORD, or /WORD sizes are valid only with the /PHYSICAL and /VIRTUAL address-space-qualifiers.

Command Parameters

address-identifier
Specifies the starting numeric or symbolic address of a location or series of locations whose contents you want to examine. If no address identifier is used, "+" is assumed. (See Section 5.3.1 for a description of command language symbols.) The numeric address is a numeric literal. (See Appendix B.)

Command Qualifiers

/NEXT=n
Causes the EXAMINE command to be repeated n times at successively higher addresses. The amount of address increment/decrement depends on the size qualifier (see below). For GPRs and IPRs, the address increments by one.

The EXAMINE command continues until the command has been repeated n times or until one of the following access errors occurs:

- The GPR number exceeds 15(decimal).
- The IPR number exceeds 255(decimal).

address-space-qualifiers
The following qualifiers may be used on any EXAMINE address to override the default address space setting.

/GPR
/GPR indicates that the address is to be interpreted as the number of a VAX General Purpose Register (GPR). The GPR number must be between 0 and 15 (decimal). If /NEXT=n is specified and the GPR number exceeds 15 (decimal), then the command terminates with an address overflow error.

/IPR
/IPR specifies that the address is to be interpreted as the number of an Internal Processor Register (IPR). The IPR number must be between 0 and 255 (decimal). If the /NEXT=n qualifier is specified and the IPR number exceeds 255 (decimal), the command terminates with an address overflow error.

/PHYSICAL
Places the address space in physical memory or physical I/O space.

/SDF
Examines the Slow Data File.

/TEMP
Slow Data File locations known only to microcode, not macrocode. The TEMPS are listed. Only numeric literals can be used with the /TEMP qualifier.

/VIRTUAL
Interprets the address as a virtual address. Access and protection checking occurs. If access is not allowed, the console issues an error message. If memory management is not enabled, virtual addresses equal physical addresses.

size-qualifiers
The following qualifiers may be used on any EXAMINE address. The initial default data size is LONGWORD.

/BYTE
References a 1-byte quantity.

/WORD
References a 2-byte quantity.

/LONGWORD
References a 4-byte quantity.

/QUADWORD
References an 8-byte quantity.

xamples

1.
```
>>> EX/P 0
P 00000000 EFDE4800
```
   Examines physical address 0, default size=long.

2.
```
>>> EX/B B197
```
   Examines a byte at physical location B197.

3. `>>>EX/G 9`

   Examines GPR 9, default size changed to LONGWORD.

4. `>>>EX/I 133`

   Examines IPR 133 hexadecimal.

5. ```
   >>>EX/L/P  20080000/n:8
   P 20080000     0000B810
   P 20080004     0000E084
   .
   .
   .
   P 20080020      00000000
   ```

   Examines nine longwords, starting at a physical address.

6. `>>>EX NMIFSR`

   Examines the IPR NMIFSR, the size defaults to LONGWORD.

7. `>>>EX/S`

   Examines the Slow Data File location last-address + 4 (+ is assumed), default size is LONGWORD.

8. ```
   >>>EX/T 19/n=3
   M 00000019 006600F6
   M 0000001A 00000000
   M 0000001B 30BFFD97
   M 0000001C 00000000
   ```

   Examines four TEMPS, starting with 19.

9. `>>>EX/V 35DE/n=16`

   Examines 17 longwords, starting at virtual address 35DE.

10. `>>>EX/W 51C7`

    Examines a word at virtual address 51C7.

11. `>>>EX/B`

    Examines a byte at the last address +1 (+ is assumed) or 51C8.

12. `>>>EX PC/N:10`

    Examines the PC and the next 10 longwords (IPRs reset default size).

13. `>>>EX PSL`

    Examines the PSL.

EXIT

Terminates the console program and returns control to the console operating system (P/OS).

Format

EXIT

Description

The console program exits to the console operating system, enabling the operating system to function. To run the console program again, select the CONSOLE/DCL entry from the VAX CONSOLE menu or use the RUN command from DCL and RUN the CONTROL program.

Exiting from Micromonitor mode is described in Chapter 6.

Restriction

EXIT is not permitted from the remote port.

Examples

1. `>>>EXI`

2. `>>>EXIT`

FIND

Initiates a memory search required to boot or restart an operating system.

Format

FIND

Command Qualifiers	**Defaults**
/MEMORY	None
/RPB	None

Description

Use the FIND command in the boot and restart command procedures to either search for space for the primary bootstrap program or to find a restart parameter block. FIND also controls multiple boots or restarts.

Default

If no switch is specified with the FIND command, /RPB is the default.

Restriction

The CPU must be halted.

Command Qualifiers

/MEMORY

Requests a search of main memory starting at address 0, looking for a page-aligned 64KB block of good memory into which the primary bootstrap program (VMB) can be loaded. The stack pointer is loaded at starting address + 512, where VMB is loaded and started. To prevent the console from repeatedly trying to bootstrap the system, the FIND

/MEMORY command sets the COLD_RESTART flag. If another FIND /MEMORY command executes before the RESTART command completes or boots the system, the second command fails, because the COLD_RESTART flag is still set. When the operating system successfully boots, it sends a message to the console to clear the COLD_RESTART flag. To clear the COLD_RESTART flag and boot manually, use the CLEAR RESTART_FLAGS command.

The FIND/MEMORY command is in the boot command procedures DEFBOO.COM and device-nameBOO.COM.

/RPB

Requests a search of physical memory for a valid restart parameter block, which is used to restart the system after a power failure or error halt. The stack pointer is loaded with the RPB (starting address + 512) and the PC is loaded with the contents of RPB (starting address + 4). The search leaves the contents of memory unchanged. To prevent the console from repeatedly trying to restart the system, the FIND/RPB command sets the WARM_RESTART flag. If another FIND/RPB command is executed before the RESTART command completes or the system successfully restarts, the second FIND/RPB command fails because the WARM_RESTART flag is still set. When the operating system successfully restarts, it sends a message to the console to clear the WARM_RESTART flag. To clear the WARM_RESTART flag and restart the system manually, use the command CLEAR RESTART_FLAGS.

The FIND/RPB command is in the command procedure RESTAR.COM, which is executed automatically after a power failure or error halt if AUTO_RESTART is enabled.

Examples

1. `>>> F/M`

2. `>>> F/R`

3. `>>> FIND/MEMORY`

4. `>>> FIND/RPB`

HALT

Halts activity in selected and available CPU(s). Instead of executing the next macroinstruction, the CPU(s) halts.

Format

HALT

Description

HALT stops macrocode execution in selected and available CPUs, beginning at the next macroinstruction. I/O operations already in progress are unaffected. For a description of "Selected" and "Available," see descriptions of the SET CPU and VERIFY MODULE PLACEMENT commands, as well as Section 3.2.

Restriction

If the processor is already halted when the HALT command is issued, a warning message is displayed. If the CPU is malfunctioning, it may never try to execute the next macroinstruction, and therefore never halt.

Examples

1. `>>>H`

2. >>> HALT

When the CPU halts, a message similar to the following appears on the console screen:

```
?00  LEFT CPU  -  CPU HALTED
PC = 80001A21
```

This message provides the 2-digit halt code indicating the operator-requested HALT and, on the VAX 8800, shows which CPU is halted (left or right).

HELP

Provides on-line information about console commands.

Format

HELP *[help-topic]*

Description

HELP describes the specified command, including syntax, qualifiers, parameters, and options.

If HELP is issued with no qualifiers, a list of HELP topics is displayed on the terminal.

Restriction

The HELP command cannot be executed in a command procedure.

Command Parameter

help-topic
A topic that the help utility describes. HELP opens and prints the contents of the file named help-topic.HLP. HELP topics follow:

@	EXAMINE	LOOP*	SENSE
BOOT	EXIT	MICROSTEP	SET
CLEAR	FIND	NEXT	SHOW
CONTINUE	HALT	PERFORM	START
DEPOSIT	IF	POWER	TEST
DIAGNOSE*	INITIALIZE	PROBE	UNJAM
DISABLE	LINK	REPEAT	VERIFY
ENABLE	LOAD	SELECT*	WAIT

* Command valid in Micromonitor mode only.

If the help file contains more than 22 lines, the prompt "Press RETURN for more ..." appears at the bottom of the screen. To see the next screen, type RETURN. To stop printing the help file, enter CTRL/C before pressing RETURN.

Examples

1. `>>>HELP MICROSTEP`

2. `>>>HE M`

3. `>>>HELP INITIALIZE`

4. `>>>HE I`

5. `>>>HELP ENABLE`

6. `>>>HE EN`

IF

Tests the value of a specified state bit and executes the console command following the THEN statement if the state bit is true (or false if NOT is specified).

Format

IF [NOT] *state-bit-name* ***THEN*** *console-command*

Description

The IF command conditionally executes console commands. The conditional execution is based on one of the console state bits. Each state bit has a mnemonic, all of which are defined in Appendix B. The state bits are set by console commands or console software.

The success or failure of the console previous command is reflected in the state bit mnemonic $STATUS. (The IF command itself does not affect $STATUS.) If the console-command is an "@" command, another stream of console commands follows. The use of "@" commands and $STATUS and other state bits allows complicated sequences, such as the initialization sequence, to be expressed in command files.

Command Parameters

state-bit-name
Each state bit is expressed by a mnemonic.

console-command
Any valid console command

Examples

1. `>>>IF NOT AUTO_POWERON THEN @EXIT`

   See ENABLE AUTO POWERON.

2. ```
   >>>IF NOT POWER_STATUS T POWER ON
   ```

   Checks the power. If it is not on, issues the POWER ON command.

3. ```
   >>>IF AUTO_RESTART T @RESTART
   ```

   Enables AUTO RESTART command.

4. ```
   >>>IF AUTO_BOOT THEN @DEFBOO
   ```

   Enables the AUTO BOOT command.

5. ```
   >>>LOAD/WCS UCODE.BIN
   >>>IF NOT $STATUS THEN @EXIT
   ```

   Checks outcome of previous command and branches to EXIT.COM if previous command fails.

INITIALIZE

Sets selected and available CPU(s) or memory to an initial state.

Format

INITIALIZE

Command Qualifiers	**Defaults**
/CREATE_CONFIGURATION_FILE	None
/MEMORY	

Description

INITIALIZE is used to bring the hardware to a known consistent state. It causes the console support microcode to execute and enables the micromachine. The INITIALIZE command without qualifiers copies the value of NEXT_PRIMARY to CURRENT_PRIMARY.

For a VAX 8800, INITIALIZE applies to selected and available CPU(s). The CPU clocks are left on. Traps, stalls, and the Nautilus Memory Interconnect (NMI) microsequencer are enabled.

The INITIALIZE command can halt a CPU under certain conditions. If it does, a warning appears.

Restrictions

The CPU(s) must be halted.

To insure that the revision Internal Processor Registers (IPRs) are properly filled in by INITIALIZE, the following commands must be executed prior to INITIALIZE:

- SENSE REV CPU

  Reads and stores revisions of CPU modules, CLK, and CPU backplane.

- VERIFY REVISION CONSOLE

  Determines the top revision of the console and displays incompatibilities among the console program, RTI driver, P/OS, and EMM firmware, as well as necessary corrections.

- VERIFY REVISION CPU

  Determines the top revision level of each CPU. Any incompatibilities among the CPU backplane, the CPU modules, and the clock module, as well as any necessary corrections, are displayed. This option applies to available and selected CPU(s).

- VERIFY REVISION MICROCODE

  Microcode must be loaded for microcode revisions to be retrieved and stored.

- LOAD/WCS

  Data is loaded to the WCS.

- LOAD/IBDECODER

  Data is loaded to the instruction stream Decoder RAMs.

- LOAD/CACHECONTROL

  Data is loaded to the cache control store.

- LOAD/SDF

  Data is loaded to the Slow Data File, which requires that the CPU(s) be halted.

Command Qualifiers

/CREATE_CONFIGURATION_FILE

Creates a MEMCONFIG.DAT file, which contains the interleaved map of the physical address space to physical memory. This file is based on the memory arrays present when the command is executed. This qualifier must be used prior to the /MEMORY qualifier in order to create an initial (or new) MEMCONFIG.DAT file. That is done during the first startup procedure (see Section 4.3) and when you add or remove memory arrays (see Section 4.8).

/MEMORY

This qualifier causes the console to load the MEMCONFIG.DAT file into the memory controller's decode RAM. If the memory controller's cold bit is set, indicating that the memory is invalid, physical memory is initialized to zero with good parity. If the console cannot find the MEMCONFIG.DAT file, it creates a new one and notifies you with a message.

NOTE

If memory is added or removed, the command INITIALIZE /CREATE_CONFIGURATION_FILE and INITIALIZE /MEMORY must be executed before you attempt to reboot your system.

Examples

1. `>>> INIT`

2. `>>> I`

3. `>>> I/MEMORY`

4. `>>> I/CREATE_CONFIGURATION_FILE`

LINK

Creates a temporary indirect command file.

Format

LINK

Description

LINK is used for repeatedly executing a series of console commands. After building the temporary command procedure LINK.COM, execute it with the PERFORM command. (The @LINK command is equivalent to this.) The temporary command procedure remains until the next LINK command is executed, replacing the previous command procedure with the new commands.

When LINK is entered, the console deletes the previous LINK command procedure and prompts with the link-mode prompt (< < <) . Console commands are saved in the temporary file LINK.COM (on the console's hard disk) until CTRL/C is entered. Renaming this file saves it. It can also be edited with an editor running under P/OS.

Restriction

The PERFORM command is invalid inside a file created by the LINK command.

Examples

1.
```
>>> LI
<<<EX 20080000
<<<dep 20080000  0000E81C
<<<CTRL/C Return)
```

   A LINK.COM is created which calls for EXAMINE and DEPOSIT at the indicated addresses. CTRL/C terminates input.

2.
```
>>> LINK
<<<P 0/n:100
<<<P 0/n:100 0
<<<CTRL/C Return)
>>> PERFORM
```

A LINK.COM file is created, using the POWER OFF command. Then the PERFORM command is issued, causing the LINK.COM file to execute.

3.
```
>>> LINK
<<<SHO CPU
<<<SHO STA
<<<CTRL/C Return)
>>> PERFORM
```

A LINK.COM file is created, using the SHOW commands. Then the PERFORM command is issued to cause the LINK.COM file to execute.

LOAD

Loads data into memory or a control store.

Format

LOAD *file-name | /DATA=numeric-literal*

Command Qualifiers	**Defaults**
/CACHECONTROL	
/CONTROLSTORE or WCS	
/DATA=numeric-literal	
/IBDECODER	
/MAINMEMORY	
/SDF	
/START=numeric-address	
/UWCS	

Description

The LOAD command is used by the system initialization routine to load the Writeable Control Store (WCS), Instruction Stream Decode Rams (IBDECODER), Slow Data File (SDF), and CACHECONTROL store.

NOTE

Failure of the LOAD command can mark a CPU unavailable.

Defaults

If no qualifiers are specified, the command defaults to loading main memory (/MAINMEMORY), starting at physical address 0 (/START=0).

Restrictions

The LOAD data comes either from a file or the command line (expressed as a numeric literal–see Appendix B. Supply either a file name or the /DATA qualifier. (See Command Qualifiers.)

The CPU must be halted and the CPU clocks running. (See the SET CLOCK command or SHOW STATUS.) To load main memory or the Slow Data File (see /SDF), the CPU must be halted.

Files of control-store data must be in the correct format for the particular control-store specified in the LOAD command. If the Writeable Control Store (/WCS), Instruction Stream Decoder Rams (/IBDECODER), Cache Control Store (CACHECONTROL), or Slow Data File (/SDF) is loaded from a file, the header block of the file is checked for the following compatibilities:

> The file is the same type of control store to which it is to be loaded.
>
> The revision level of the file is the same as the previous file's level loaded into that control store since the last time the console was powered on.

The /DATA qualifier and file name parameter cannot be used at the same time.

Command Parameter

file-name
Name of the file containing the data to be loaded. It can be any valid file name, including device and directory. The console device names are DW1 (fixed disk) and DZ1 and DZ2 (diskette drives 1 and 2).

Command Qualifiers

/CACHECONTROL
The destination is the cache control store. LOAD/CACHECONTROL applies to selected and available CPU(s). If a parity error is detected, the CPU is marked unavailable.

/CONTROLSTORE or /WCS

The destination is the WCS. LOAD/WCS applies to selected and available CPU(s). If a parity error is detected, the CPU is marked unavailable.

/DATA=numeric-literal

A single element of the control store or memory is loaded. You enter data on the console terminal and load the data into the destination indicated by the qualifier designating the address space and the /START qualifier indicating the address in that address space. (See Appendix B.)

If loading main memory, the data is assumed to be a longword. If loading a control store, the data is assumed to be the correct length for that control store. Maximum sizes are as follows:

IBDECODER (or DRAM): 24 bits
CACHECONTROL (or CCS): 32 bits
SDF: 32 bits
MAINMEMORY: 32 bits

/IBDECODER

The destination is the instruction stream Decoder RAMs. LOAD /IBDECODER applies to selected and available CPU(s).

/MAINMEMORY

The data's destination is main memory. This qualifier requires the CPU(s) to be halted.

/SDF

The data's destination is the Slow Data File, which requires that the CPU(s) be halted. LOAD/SDF applies to selected and available CPU(s).

/START=numeric-address

The data is loaded, starting at the address specified. The address is expressed as a numeric literal (see Appendix B).

If the /START qualifier is omitted, data is loaded at address=0 of the load destination. /START is ignored if loading microcode or SDF (valid when loading memory) from a file, because the address(es) are embedded in the file.

/UWCS

You use the /UWCS qualifier to load your own microcode into the User Writable Control Store (UWCS). You can do this only if SDF has been loaded. The system initialization procedure SYSINIT.COM provides the commands to enable you to do this (though they are commented out). The VAX Extended Function, XFC, instruction provides the mechanism for software to request services of the nonstandard microcode loaded into the UWCS.

Examples

1. >>> `LOAD/MAIN/START=0 CDRV0103.EXE`

   Load CDRV0103.EXE file into main memory starting at address 0.

2. >>> `L/MAIN/ST=0 EGYPTD.EXE`

   Load EGYPTD.EXE file into main memory starting at address 0.

3. >>> `LOAD/M/S=4530/D=00000000`

   Load D=00000000 into main memory starting at address 4530.

4. >>> `L TESTPGM.EXE`

   Load TESTPGM.EXE file; by default to main memory at address 0.

5. (Contents and machine response of the load portion of SYSINIT.COM)

```
>>>L/W UCODE.BIN

          LOAD DONE, 16384 WORDS LOADED, REV=n.


>>>IF NOT $STATUS THEN @EXIT
>>>L/I DRAM.BIN

          LOAD DONE, 1195 WORDS LOADED, REV=n.


>>>IF NOT $STATUS THEN @EXIT
>>>L/C CCODE.BIN

          LOAD DONE, 256 WORDS LOADED, REV=n.


>>>IF NOT $STATUS THEN @EXIT
>>>L/M MCODE.BIN

          LOAD DONE, 26 BLOCKS LOADED, REV=n
```

MICROSTEP

Steps the CPU master clock a specified number of cycles.

Format

MICROSTEP *count*

Description

After stepping the specified number of microcycles, the console displays the micro-PCs of available CPU(s) and enters "space bar step mode". On the VAX 8800, subsequent pressing of the space bar causes another microcycle to be executed on both CPU(s). Space bar step mode is exited by pressing any key other than Spacebar or Return. The micro-PCs of the available CPU(s) are displayed.

The micro-PCs are displayed in hexadecimal.

NOTE

The micro-PCs on the VBUS are those of the trap silo, not the actual micro-PCs. Therefore, single stepping through a trap routine or decoder cycles does not change the micro-PCs.

Restrictions

The CPU clock(s) must be stopped. (See the SHOW STATUS and/or SET CLOCK commands.) The console will automatically stop the clock(s) unless a CPU is running.

If MICROSTEP is executed from a command file, "space bar step mode" (see Microstep command) is not entered. Control returns to the command file after the initial number of microinstructions have executed.

Command Parameter

count
Number of microcycles to step, expressed in a numeric literal. (See Appendix B.) If **count** is not specified, 1 is assumed. The maximum number of cycles that can be stepped at one time is 255.

Examples

1. `>>> MICROSTEP 6`

```
LEFT  upc1:3419 upc2:3418 upc3:3417    RIGHT upc1:3419 upc2:3418 upc3:3417
LEFT  upc1:3410 upc2:3419 upc3:3418    RIGHT upc1:3410 upc2:3419 upc3:3418
LEFT  upc1:3411 upc2:3410 upc3:3419    RIGHT upc1:3411 upc2:3410 upc3:3419
LEFT  upc1:3412 upc2:3411 upc3:3410    RIGHT upc1:3412 upc2:3411 upc3:3410
LEFT  upc1:3413 upc2:3412 upc3:3411    RIGHT upc1:3413 upc2:3412 upc3:3411
LEFT  upc1:3414 upc2:3413 upc3:3412    RIGHT upc1:3414 upc2:3413 upc3:3412
LEFT  upc1:3415 upc2:3414 upc3:3413    RIGHT upc1:3415 upc2:3414 upc3:3413 H

>>>
```

VAX 8800 example. Shows upc locations for left/right trap silos.

NEXT

Executes the specified number of macroinstructions.

Format

NEXT *count*

Description

After the last macroinstruction is executed, the program counter (PC) is displayed and the console enters space bar step mode. The next macroinstruction is executed, and the subsequent PC is displayed each time the space bar is pressed. Typing any other character causes an exit from space bar step mode.

The PC is displayed in hexadecimal.

Restriction

If NEXT is executed from a command file, space bar step mode is not entered. Control returns to the command file after the initial number of macroinstructions execute.

Command Parameter

count

The number of macroinstructions to be executed is expressed in a numeric literal. (See Appendix B.) If **count** is not specified, the default is one macroinstruction. The maximum value of **count** is 32767.

Examples

1. ```
   >>> N 2
   LEFT CPU - PC = 00000009
   LEFT CPU - PC = 00000010
   ```

   Executes next two macroinstructions on VAX 8800's left CPU.

2. ```
   >>> N 2
   CPU - PC = 00000009
   CPU - PC = 00000010
   ```

   Executes next two macroinstructions on single-CPU model (8700/8550 /8500).

PERFORM

Executes the temporary command procedure created by the LINK command. PERFORM is identical to the command @LINK.

Format

PERFORM

Description

The temporary command procedure LINK.COM can be PERFORMed as many times as desired until the next LINK command is entered. See the @ command for a description of how command procedures are processed.

Restriction

The PERFORM command may not be included in a LINK.COM file created by the LINK command.

Examples

1.
```
>>> LINK
<<<EXAMINE/L/P 20080000
<<<EXAMINE 56B4
<<<EXAMINE 200A0000/N=5
>>>CTRL/C Return)
```

   Shows LINK.COM file which contains three EXAMINE commands.

2.
```
>>>PERFORM
P 20080000    00000000
P 000056B4    00001234
P 200A0000    00000001
P 200A0004    00000002
P 200A0008    00000003
P 200A000C    00000004
P 200A0010    00000005
P 200A0014    00000006
>>>
```

Executes LINK.COM file in Example 1 (PERFORM command is equivalent to @LINK). This example shows the addresses being examined. (Note that the third EXAMINE command in Example 1 has a /NEXT=5 qualifier.)

POWER

Changes the state of the VAX power system by issuing commands to the Environmental Monitoring Module (EMM). The power can be ON, OFF, or STANDBY.

Format

POWER *(option)*

Description

One of the following options must be used with the POWER command: OFF, ON, STANDBY. The options are described as individual commands.

WARNING

If it is necessary to work on the CPU, you must issue the POWER OFF command and shut off the main power breaker of the CPU. The following safety feature is designed to protect those working on the system hardware:

When you issue the POWER OFF command, the console saves the contents of the AUTO_POWERON keyswitch and disables it. Then with any kind of power failure (including turning off the console), the initialization procedure (SYSINIT) detects that the AUTO_POWERON keyswitch is disabled and does not power up the system. That prevents a console-only power failure, for instance, from powering up the entire system.

When the work on the system hardware is finished, turn on the circuit breaker and issue the POWER ON command. The AUTO_POWERON keyswitch value is restored to its setting prior to the POWER OFF command.

If you have enabled the AUTO_POWERON keyswitch and have not issued a POWER OFF command, the console responds to power failures automatically. If the circuit breaker is then inadvertently turned on, the console automatically powers up the system, endangering those working on the system hardware.

Restriction

POWER commands cannot be executed from the remote port. An error message is printed.

POWER OFF

Turns off all system regulators.

Format

POWER OFF

Description

POWER OFF turns off all regulators in the reverse order they are turned on (see POWER ON).

The POWER OFF command saves the current AUTO_POWERON state bit setting and disables AUTO_POWERON.

Restriction

POWER commands cannot be executed from the remote port.

Examples

1. `>>> P O`

2. `>>> POWER OFF`

POWER ON

Turns on all system regulators and enables the CPU(s) to function.

Format

POWER ON

Description

POWER ON turns on all the regulators, deasserts the DCLO signals, deasserts the ACLO signals, enables battery backup (VAX 8800), enables EMM default mode monitoring, tests the RTI, initializes the shadow copies of the clockboard registers, unlocks the CSEQ module, clears CPU_INIT to the CPU(s), initializes sensed revision tables, and loads the counts used in loading the control stores. The regulators are turned on in the following order:

- VAX 8800: J, B, C, F, H, E, and D
- VAX 8700/8550/8500: J, B, E, H, and D

The POWER ON command enables the saved AUTO_POWERON state. This command may take as long as 10 seconds to complete. It may be aborted with CTRL/C before execution.

Restrictions

POWER ON cannot be executed until the VERIFY EMM command has been successfully executed.

POWER ON

POWER commands cannot be executed from the remote port. An error message is printed.

Examples

1. `>>>POWER ON`

POWER STANDBY

Powers down system to a STANDBY state. All regulators except B and J are turned off.

Format

POWER STANDBY

Description

POWER STANDBY turns off the system regulators in the same order as the POWER OFF command, stopping at regulator B. Regulator B supplies power to memory, and regulator J supplies additional power to B. In STANDBY mode, memory contents are preserved.

Restrictions

When power is off, the POWER STANDBY command cannot be executed.

POWER commands cannot be executed from the remote port.

Examples

1. `>>>P S`

2. `>>>POWER STANDBY`

PROBE VBUS

PROBE VBUS specifies which bit of the accumulator receives which Visibility Bus data bit. The accumulator is a 32-bit quantity that can be cleared or shown (see the descriptions of the CLEAR and SHOW commands). PROBE is useful for formatting VBUS data.

Format

PROBE VBUS *vbus-address accumulator-bit*

Description

The Visibility Bus (VBUS) displays signals for diagnostics and debugging. The VBUS address is split into two fields. The low-order 12 bits specify a multiplexer address in each of the eight CPU modules. The high-order hexadecimal digit selects one of the eight CPU modules from which to read the VBUS data. This command is used by Field Service for maintenance purposes.

```
                       vbus_address (hex)
                    +------------------------+
     hex digit--->|  3   |  2   |  1   |  0  |
                    +------------------------+
                           |
     module selector       | 12 bit address
                           | sent to VBUS
     0 SEQ    4 ADP        |
     1 WCS    5 SLC0       |
     2 DEC    6 SLC1       |
     3 CCS    7 SHR        |
```

Refer to the *System Maintenance Guide* for more information about the addresses that can be probed.

Restriction

The CPU master clock(s) must be stopped. (See the SHOW STATUS and /or SET CLOCK commands.)

Command Parameters

vbus-address
The starting VBUS address at which the probe command reads, expressed as a numeric literal. (See Appendix B.) Selects the module and the signal in the module.

accumulator-bit
A number between 0 and 31 that specifies which bit of the accumulator to OR with the VBUS data bit. The results replace the accumulator bit.

Examples

1.
```
>>>CLEAR ACCUMULATOR
>>>PROBE VBUS %X1234 00
```

   %X1234 is the address of VBUS bit %X234 on the WCS module and is put in accumulator bit 0.

2.
```
>>>PR VBUS %X1235 01
```

   %X1235 is the address of VBUS bit %X235 on the WCS module and is put in accumulator bit 1.

REPEAT

Causes a console command to execute continuously.

Format

REPEAT *(command)*

Description

REPEAT mode forces the continuous execution of the console command. The REPEAT command can be terminated with CTRL/C.

Restrictions

The following commands cannot be used with REPEAT: @, ASSIGN, DISABLE, ENABLE, EXIT, HALT, HELP, LINK, NEXT, POWER, PERFORM, and REPEAT.

If the REPEAT command is used in a command procedure, two CTRL/C sequences must be used to stop execution of the command file. The first CTRL/C stops the REPEAT command, and the second stops the command procedure.

Command Parameter

(command)
The REPEAT command can be used with any of the following console commands:

BOOT	INITIALIZE	SHOW
CLEAR	LOAD	START
CONTINUE	LOOP	TEST
DEPOSIT	MICROSTEP	UNJAM
DIAGNOSE	PROBE	VERIFY
EXAMINE	SELECT	WAIT
FIND	SENSE	
IF	SET	

Examples

1. `>>>REPEAT INITIALIZE`

   Repeats the INITIALIZE command.

2. `>>>R E 30`

   Repeats the command to EXAMINE physical address 30.

3. `>>>R E PC`

   Repeats the command to EXAMINE the program counter.

SENSE REVISION

Reads the revision(s) of the specified component group and stores them in a table in console memory. The data is used by the VERIFY REVISION commands.

Format

SENSE [REVISION] *(option)*

Description

SENSE REVISION affects all selected and available CPU(s). After using SENSE REVISION to read the revisions, they can be displayed with the SHOW REVISION SENSED command. SENSE REVISION is necessary prior to using the VERIFY REVISION commands.

Microcode revisions are implicitly sensed by the LOAD command.

The SENSE REVISION command requires one of the options listed below. The options are described as individual commands.

Command Options

CONSOLE

CPU(S)

MCL

NBI

SENSE REVISION CONSOLE

Reads the revision of the console application software, Real Time Interface (RTI) driver, Environmental Monitoring Monitor (EMM) firmware, and the console operating system (P/OS).

Format

SENSE [REVISION] CONSOLE

Examples

1. `>>>SEN CO`

2. `>>>SEN R CO`

3. `>>>SENSE CONSOLE`

SENSE REVISION CPU(S)

Reads the CPU backplane and module revisions for selected and available CPU(s).

Format

SENSE [REVISION] CPU(S)

Restrictions

Module revisions are accessed through the VBUS. Thus, CPU master clock(s) must be stopped to execute this command (see the SHOW STATUS and/or SET CLOCK commands). Module keying must be successfully verified first, using the VERIFY MODULE PLACEMENT command.

Examples

1. `>>>SENSE REVISION CPU(S)`

2. `>>>SEN C`

3. `>>>SENSE CPU(S)`

4. `>>>SEN R C`

SENSE REVISION MCL

Reads the revision of the memory controller.

Format

SENSE [REVISION] MCL

Restriction

The CPU must be halted, and the CPU clock must be running. (See the SET CLOCK command or SHOW STATUS.)

Examples

1. `>>>SENSE REVISION MCL`

2. `>>>SEN M`

3. `>>>SEN R M`

SENSE REVISION NBI

Reads the revisions of the NBIA and NBIB modules.

Format

SENSE [REVISION] NBI

Restriction

The CPU must be halted and the CPU clock must be running. (See the SET CLOCK command or SHOW STATUS.)

Examples

1. `>>>SENSE R NBI`

2. `>>>SEN N`

3. `>>>SENSE NBI`

SET

Establishes or changes settings for specified options.

Format

SET *(option)*

Description

The SET command requires one option. The options are described as separate commands.

Command Options

CLOCK

CPU (VAX 8800)

DEFAULT

EMM

MARGINS

NEXT_PRIMARY (VAX 8800)

NOVERIFY

RELOCATION

SOMM

SYNC

TERMINAL

TOMM

VERIFY

SET CLOCK

Turns on or off the CPU master clock or sets the CPU clock period.

Format

SET CLOCK *(option)*

Description

The SET CLOCK command affects clock settings for all CPU(s). Settings are retained across power failures, reboots, and restarts.

Only stallable clocks are affected by the SET CLOCK command. Free running clocks are not affected.

Command Options

FAST
SET CLOCK FAST sets the clock to a period of 43 nanoseconds.

NORMAL
SET CLOCK NORMAL sets the clock to a period of 45 nanoseconds.

OFF
SET CLOCK OFF stops the clock.

ON
SET CLOCK ON starts the clock running at the cycle time previously set.

SLOW
SET CLOCK SLOW sets the clock to a period of 48 nanoseconds.

Examples

1. `>>> SE CL ON`

   Turns clock on.

2. `>>>SET CLOCK SLOW`

   Sets clock period to 48 nanoseconds.

3. `>>>SE CL F`

   Sets clock period to 43 nanoseconds.

SET CPU

Logically connects the console terminal to the specified CPU(s). By selecting the CPU, this command affects the scope of subsequent commands. This command applies only to the VAX 8800.

Format

SET CPU *(option)*

Description

SET CPU changes the selected CPUs and the impact of commands whose scope is for all available and selected CPUs. When SET CPU is specified, commands affect the selected CPU(s). When SET CPU BOTH is specified, single-CPU commands apply to the current primary CPU.

The SET CPU command requires one of the options listed below.

Option

BOTH
SET CPU BOTH selects both CPUs. When SET CPU BOTH is set, single-CPU commands apply to the current primary CPU. If it is not available, a warning message is printed, but the terminal is connected to the CPU for diagnostics.

CURRENT_PRIMARY
SET CPU CURRENT_PRIMARY logically connects the terminal to the CPU designated as primary. The logical primary CPU is translated to a physical CPU (LEFT or RIGHT), and that CPU is selected.

CURRENT_SECONDARY
SET CPU CURRENT_SECONDARY logically connects the terminal to the CPU designated as the secondary. The logical secondary CPU is translated to a physical CPU (LEFT or RIGHT), and that CPU is selected.

LEFT

SET CPU LEFT logically connects the terminal to the CPU located on the left side of the backplane (when viewed from the front of the machine).

NEXT_PRIMARY

SET CPU NEXT_PRIMARY logically connects the terminal to the primary CPU, specified by the SET NEXT_PRIMARY command. (See the description of the SET NEXT_PRIMARY command.)

NEXT_SECONDARY

SET CPU NEXT_SECONDARY logically connects the terminal to the primary CPU, implied by the SET NEXT_PRIMARY command. (See the description of the SET NEXT_SECONDARY command.)

RIGHT

SET CPU RIGHT logically connects the terminal to the CPU on the right side of the backplane (when viewed from the front of the machine).

Examples

1. `>>>SE C B`

   Selects both CPUs.

2. `>>>SET CPU LEFT`

   Selects the left CPU.

3. `>>>SE C CURRENT_P`

   Connects the console terminal to the primary CPU.

4. `>>>SE C NEXT_S`

   Connects the console terminal to primary CPU as specified by the SET NEXT_PRIMARY command.

SET DEFAULT

Establishes an address space, size, and radix for use by other console commands.

Format

SET DEFAULT *[address-space, size, radix]*

Description

SET DEFAULT changes either the default address space or default size for the EXAMINE and DEPOSIT commands or the default radix in which numeric input is interpreted.

Command Parameters

The following table lists the kinds of defaults and the values for each one.

address-space	size	radix
CACHE	BYTE	BINARY
GPR	LONGWORD	DECIMAL
IPR	QUADWORD	HEXADECIMAL
PHYSICAL	WORD	OCTAL
SDF		
TB		
TEMP		
WCS		

Restrictions

One option from each category may be entered. If more than one option from a category is entered, only the last option is used. For example, VIRTUAL, QUADWORD, OCTAL is accepted. However, if PHYSICAL, VIRTUAL, QUADWORD, OCTAL is entered, PHYSICAL is discarded, and the remaining options are accepted.

Examples

1. `>>>SE D P,L,H`

   Sets address-space PHYSICAL, size LONGWORD, and radix HEXIDECIMAL as defaults.

2. `>>>SET DEFAULT Q,V,D`

   Sets size QUADWORD and radix DECIMAL as defaults.

SET EMM

Loads parameters into the Environmental Monitoring Module (EMM).

Format

SET EMM

Description

On power up, the EMM's default mode parameters are loaded into a RAM from a ROM in the EMM. The console, after its power-up sequence, loads CPU-specific parameters into the RAM. Internal errors in the EMM cause RAM contents to be destroyed. In this event, the console automatically reloads the parameters.

Examples

1. `>>>SE E`

2. `>>>SET EMM`

SET MARGINS

Changes the values against which the EMM tests for voltages out of tolerance.

Format

SET MARGINS *(option) [regulator-list]*

Description

The EMM monitors the environmental conditions of the CPU(s) against the nominal (normal) values supplied in the EMM initialization sequence. However, the EMM uses a different set of parameters when monitoring a "margined" regulator.

The SET MARGINS command requires one of its options.

Command Options

HIGH

The SET MARGINS HIGH command instructs the EMM to allow a 5% increase over the normal values.

LOW

The SET MARGINS LOW command instructs the EMM to allow a 5% decrease in the normal values.

NORMAL

The SET MARGINS NORMAL command returns the parameter values to those set at initialization.

Command Parameters

regulator-list

Specifies one of the following regulators for which to set the margins:

VAX 8800: B, C, D, E, F, or H

VAX 8700/8550/8500: B, D, E, or H

The default is to set all of the regulators to the specified option value.

Examples

1. `>>> SE M N`

   Returns parameter values to those at initialization.

2. `>>> SET MARGINS LOW`

   Allows a 5% decrease in values.

3. `>>> SE M HIGH`

   Allows a 5% increase in values.

4. `>>> SET MARGINS N E,D,H`

SET NEXT_PRIMARY

Specifies which CPU is the primary and which is the secondary when the system reboots or restarts. This command applies only to the VAX 8800.

Format

SET NEXT_PRIMARY *(option)*

Description

At the next reboot or restart, the CPU specified by SET NEXT_PRIMARY becomes the current primary CPU (copied by the INITIALIZE command). The other CPU becomes the secondary.

The initial NEXT_PRIMARY is the left CPU; the initial secondary is the right CPU.

The NEXT_PRIMARY command requires one option.

Option

LEFT
Specifies the left CPU as the next primary (and the right as the secondary).

RIGHT
Specifies the right CPU as the next primary (and the left as the secondary).

Examples

1. `>>>:SE N RIGHT`

2. `>>> SET NEXT_PRIMARY LEFT`

SET NOVERIFY

Disables commands in command files from being echoed on the terminal.

Format

SET NOVERIFY

Description

Allows command files to execute without each command line being displayed on the terminal. Echoing can be restored with SET VERIFY.

Examples

1. `>>>SE NO`

2. `>>>SET NOVERIFY`

SET RELOCATION

Places a value into the address relocation register.

Format

SET RELOCATION *base-address*

Description

The value established by the SET RELOCATION command is added to the effective physical or virtual address for the EXAMINE and DEPOSIT commands.

The relocation register value can be displayed with the SHOW DEFAULTS command. The value is initialized to zero.

Restriction

The relocation register is not preserved across console power failures or restarts.

Command Parameter

base-address
SET RELOCATION base-address causes the base address value to be added to the address value used by the EXAMINE and DEPOSIT commands. Base-address is a numeric literal. (See Appendix B.)

Examples

The following examples show how the SET RELOCATION command affects the EXAMINE command:

1.
```
>>>SE R 1000
>>>EXAM/P 0
   P 00001000 12345678
```

2.
```
>>>SET RELO 200
>>>EXAM/P 0
   P 00000200 12345678
```

SET SOMM

SET STOP ON MICROMATCH command. Loads the micro-PC into the Micromatch register.

Format

SET SOMM *address*

Description

The SET SOMM command loads the micro-PC into the Micromatch register. When the micro-PC matches the value of the Micromatch register, a sync pulse is generated, and the clock stops on the rising edge of the next A clock.

Command Parameter

address
A valid micro-PC in the range 0-%X3FFF.

Examples

1. `>>> SE S 328C`

2. `>>> SET SOMM 32C1`

SET SYNC

Loads the micro-PC into the Micromatch register.

Format

SET SYNC *address*

Description

SET SYNC clears Stop On Micromatch (SOMM) and Trap On Micromatch (TOMM). When the micro-PC matches the value of the Micromatch register, a sync pulse is generated.

Command Parameter

address
A valid micro-PC in the range 0-%X3FFF.

Examples

1. `>>>SE SY 0`

2. `>>>SET SYNC 1000`

SET TERMINAL

Changes the console or remote console from console I/O mode to program I/O mode or modifies the console's ID.

Format

SET TERMINAL *(option)*

Description

The SET TERMINAL command affects how program mode characters are transmitted to the VAX and how program mode activity is logged in the console log files. When you enter program mode, the last OPA setting is in effect.

SET TERMINAL has the options listed below.

Command Options

PROGRAM

The SET TERMINAL PROGRAM command causes the console to enter program I/O mode. Subsequent characters typed on the terminal are transmitted to the VAX connected to the console. If the CPU is halted, the command is rejected and an error message is printed.

OPA0, OPA4, OPA5

SET TERMINAL OPAn specifies the ID bits to be supplied with program mode data sent from the console to the CPU.

If OPA0 is set, input and output from the terminal is logged in the log file. For more information on log files, see Section 3.10.

To avoid logging program mode data in the console log file, set the ID bit to either OPA4 or OPA5. OPA4 is normally associated with the local console, and OPA5 with the remote console. Change the ID before issuing the SET TERMINAL PROGRAM command.

Examples

1. ```
   >>>SE T P
   ```

   Enter Program mode. Program mode activity will be logged.

2. ```
   >>>SET TERMINAL OPA4
   >>>SET T P
   ```

   Reset the console ID bits, then enter Program mode. Program mode activity will not be logged.

SET TOMM

SET TRAP ON MICROMATCH command. Loads the micro-PC into the Micromatch register.

Format

SET TOMM *upc-address*

Description

The SET TOMM command loads the specified micro-PC into the Micromatch register. When the micro-PC matches the value of the Micromatch register, a sync pulse is generated, and a microtrap occurs.

Command Parameters

upc-address
A valid micro-PC in the range 0-%X3FFF.

Examples

1. `>>> SE TO 200`

2. `>>> SET TOMM 1000`

SET VERIFY

Allows the contents of command procedures to be displayed during execution.

Format

SET VERIFY

Description

The SET VERIFY command causes input from command procedures to be echoed on the console and logged in the log file. This command is useful for debugging newly created command procedures.

Echoing can be turned off by the SET NOVERIFY command.

Examples

1. `>>>SE V`

2. `>>>SET VERIFY`

SHOW

Displays information about the VAX or the console.

Format

SHOW *(option)*

Description

The SHOW command requires one of the options listed below. They are described as separate commands.

Command Options

ACCUMULATOR

CPU

DEFAULTS

LOGFILE

POWER

REVISION

STATUS

TIME

SHOW ACCUMULATOR

Displays the contents of the probe accumulator.

Format

SHOW ACCUMULATOR

Description

After the PROBE command, the accumulator contains digits in the long-word according to the mask provided in the PROBE command. If the PROBE command has not yet been executed, the SHOW ACCUMULATOR command gives the following response:

%X00000000

Examples

1. `>>>SH A`

2. `>>>SHOW ACCUMULATOR`

SHOW CPU

Displays operational information about the CPU(s).

Format **SHOW CPU**

Description

SHOW CPU displays the following information:

- CPU INDICATOR (LEFT/RIGHT) (VAX 8800)

  The CPU to which single-CPU commands apply is marked by a carat (^) on the left margin of the screen. The CPU with the carat is always selected. If SET CPU BOTH is in effect (VAX 8800), the primary CPU has the carat, and both CPUs are selected. The selected CPU is affected by those commands that affect all selected CPUs. If a CPU is not selected, nothing is displayed.

- STATE

  RUNNING, HALTED, POWER OFF, UNAVAILABLE

- UMATCH

  The status of the UMATCH can be SYNC (a result of SET SYNC), SOMM (a result of SET SOMM), or TOMM (SET TOMM). The Micromatch address is also shown.

- FLAGS/STATUS (VAX 8800)

  For a VAX 8800, FLAGS/STATUS tells which CPU is available and selected, the current primary, the next primary, and the secondary. (If the secondary is disabled, SECONDARY_DISABLED appears.)

- The display also shows the settings of the software-implemented keyswitches (AUTO_POWERON, AUTO_RESTART, and AUTO_BOOT).

Examples

1. >>> SH C

```
  CPU   STATE    UMATCH     FLAGS/STATUS -- 8-JAN-86 16:01:39
>LEFT  HALTED   SYNC %X316F AVAILABLE SELECTED(DUAL) PRIMARY NEXT_PRIMARY
 RIGHT HALTED   SYNC %X0000 AVAILABLE SELECTED(DUAL) SECONDARY
Automatic Poweron:ENABLED; Restart:DISABLED; Boot:DISABLED
```

VAX 8800 example shows both CPUs halted, both CPUs selected and available. Right CPU is primary; a sync pulse occurs when upc=%X316F. Left CPU is secondary. AUTO POWER ON is enabled, RESTART and BOOT disabled.

2. >>> SH C

```
STATE    UMATCH     FLAGS/STATUS -- 11-JAN-86 14:10:59
HALTED   SYNC %X0000 AVAILABLE
Automatic Poweron:ENABLED; Restart:DISABLED; Boot:DISABLED
```

VAX 8700/8550/8500 example shows AUTO POWER ON enabled, RESTART and BOOT disabled.

SHOW DEFAULT

Displays the defaults established by the SET DEFAULT and SET RELOCATION commands. Otherwise, the defaults result from the last EXAMINE or DEPOSIT command.

Format

SHOW DEFAULT

Description

The SHOW DEFAULT command displays the following information:

- The address (in hexadecimal) used to calculate the next EXAMINE or DEPOSIT address.
- The default EXAMINE/DEPOSIT address space.
- The default size of EXAMINEs or DEPOSITs to physical or virtual memory.
- The relocation automatically applied to physical and virtual addresses (in hexadecimal).
- The radix in which input is interpreted. The radix can be overridden with a radix prefix.
- The state of the VERIFY flag.

Defaults are specified by the SET DEFAULT, SET RELOCATION, EXAMINE, and DEPOSIT commands. Default settings are retained across power failures and restarts of the console software.

Examples

1. >>>SH D

```
EXAMINE/DEPOSIT%X0000000E /GPR /LONGWORD        RELOCATION=%X00000000
DEFAULT RADIX             = Hexadecimal
VERIFY FLAG               = ON
```

VAX 8800 machine response shows current defaults for EXAMINE and DEPOSIT. Relocation default is established by the SET RELOCATION command. The Verify Flag is set by the SET VERIFY command.

SHOW LOGFILE

Displays a log file containing console output.

Format

SHOW LOGFILE

Description

Console output is saved in a journal, or log file. On the VAX 8800, a separate log file exists for each CPU.

Console mode input and output and OPA0: output from the VAX are saved in the log file. In program mode, input on the console and output from the VAX are not logged if the console ID is set to OPA4 or OPA5.

The SHOW LOGFILE command displays the log file. After the log file appears, subcommands scroll it up or down on the screen.

Subcommands

After displaying the log file, use the following function keys to scroll through it:

Key	Function
↑	Scrolls up one line.
↓	Scrolls down one line.
Prev Screen	(Also ESC[5.) Scrolls up 23 lines.
Next Screen	(Also ESC[5.) Scrolls down 23 lines.
8	Scrolls 23 lines in direction of last command.
CTRL/W	Repaints current display.

Use CTRL/C to exit the log file and return to console mode.

SHOW LOGFILE and its subcommands are not logged in the log file.

Restrictions

SHOW LOGFILE searches for a line feed <LF> to delimit records. If you scroll to an area that that has no line feeds, several seconds may elapse until the next <LF> is found. Two examples are MONITOR PROCESSES/TOPCPU and SHOW CLUSTER/CONTINUOUS. If the log file wraps around into the output of one of those commands, SHOW LOG may take a few moments to respond. Also, in such cases, the screen may produce unexpected results.

Examples

1. `>>> SH L`

2. `>>> SHO LOG`

3. `>>> SHOW LOGFILE`

SHOW POWER

Displays the current state of the sensors that the Environmental Monitoring Module can read.

Format

SHOW POWER

Description

Temperatures are shown in degrees Celsius, voltages in volts. You can change the voltage displayed, using the SET MARGINS command.

Examples

1. >>>SH P

```
POWER SYSTEM STATUS: 8-JAN-86 16:02:19
MODULE:      B      C      D      E      F      H      L+     L-
 VOLTS:   5.02   5.02  -1.99  -5.19   5.08   5.02  11.81 -12.02
 RANGE:     ok     ok     ok     ok     ok     ok     ok     ok
MARGIN:    nom    nom    nom    nom    nom    nom

 THERM:     T1     T2     T3     T4    D21    D31    D41
  TEMP:  22.99  26.99  26.32  23.99
  ZONE:    nom    nom    nom    nom    nom    nom    nom

MODULE:   B    C    D    E    F    H    J    L    LAC  KEY
STATUS:   ok   ok   ok   ok   ok   ok   ok   ok   ok   ok

SIGNAL:  PERR PENA DCLO ACLO AFF2 CBF  BBF  AFF1 55D  ASD  DME  PSW
 VALUE:  lo   hi   lo   lo   lo   lo   hi   lo   lo   lo   hi   lo
```

VAX 8800 machine response. VAX 8700, 8550 and 8500 machine response is similar, except the C and E modules are omitted.

2. >>>SHOW POWER

SHOW REVISION

Displays revisions of VAX modules.

Format

SHOW REVISION *(option)*

Description

The SHOW REVISION command displays current or historical revision data in a hierarchical fashion. Each system component has a revision level. Low-level revisions are combined to form a revision level for a group. Group-level revisions form a kernel (top-level) revision.

Revisions are used to insure that hardware, software, and their combinations are valid.

Default

If no option is specified, the default is SHOW REVISION SENSED ALL.

Command Options

HISTORY
[ALL, CONSOLE, CPU, KERNEL, MCL, MICROCODE, NBI]

SHOW REVISION HISTORY displays the contents of the revision history file REVHIST.DAT. The revision history file defines top revisions and compatibility between revisions and top revisions. The display shows the revision range of subordinate components that make up a top revision. The revision of the history file is displayed as *File rev n*.

The display is segmented into sections that describe the range of top revisions for each top revision group. The top revision groups are CONSOLE, CPU, KERNEL, MCL, MICROCODE, and NBI. Under each segment, the values of the top revisions for a given group are displayed across the page and labeled *Rev n*, where *n* is the value of the top revision. Under each top revision, a column of subordinate components is listed, with the minimum and maximum required component revision for that top revision (labeled *min* and *max*). The top revision sections also comprise the following options of the SHOW REVISION HISTORY command. If none of these options are used, the default is SHOW REVISION HISTORY ALL.

- ALL–displays the revision history for hardware and software components.
- CONSOLE–displays console revision information.
- CPU(S)–displays revision information about the CPU(s).
- KERNEL–displays the revision history of top-level revisions for components, including the CPU, console, Memory Controller (MCL), microcode, and NBI.
- MCL–displays revision information about the Memory Controller.
- MICROCODE–displays revision history about the microcode.
- NBI–displays revision history about the NBI.

SENSED
[ALL, CONSOLE, CPU, KERNEL, MCL, MICROCODE, NBI]

SHOW REVISION SENSED displays the revisions of modules sensed with the SENSE command, the kernel software components, and the top revisions that have been calculated by the VERIFY REVISION command. SHOW REVISION SENSED does not actually cause the revisions to be read or calculated. The command accepts qualifiers that limit the display to a specified revision group.

Items in the Component column identify the component or top revision. A top revision is calculated by the console from the revisions of one or more individual components. Indentation of a component name indicates that it is subordinate to a top revision. Top revisions are calculated when the VERIFY REVISION command is executed.

For the VAX 8800, the value of top revisions and components are displayed under the columns labeled Left CPU, Right CPU, and Common. (The VAX 8700, 8550, and 8500 columns are labeled CPU and Common.) Some components, such as CPU logic modules, are unique to a CPU. Since these components and top revisions might have components with different revisions in each CPU, the components are displayed under the Left CPU or Right CPU columns. Components that are not unique to either CPU, such as console software, are displayed in the Common column.

A revision for an individual component is displayed as a number or a letter. If a component is "unavailable", that component has not yet been sensed.

Top revisions are expressed as a number, flagged as unavailable, or undefined <n> . When a number is displayed, the top revision has been calculated through the associated VERIFY REVISION command. The information is stored in the revision history file. The flag "unavailable" indicates that the VERIFY REVISION command has not yet been successfully executed to determine a top revision. The flag "undefined <n> " indicates that the revisions of the subordinate components do not match any top revision described in the revision history file. The <n> is the closest top revision in the revision history file, based on the sensed revisions of the subordinate components.

The *Comments* column displays information about the component revision status. "Not calculated" or "Calculated" refers to whether the VERIFY REVISION has been successfully executed to determine a top revision. "CPUs mismatched" on a VAX 8800 is a warning that differences occur in top revisions of the CPUs, which violates a VMS compatibility rule. "Not sensed" indicates the revision is unavailable, because the SENSE command for that component was not executed or failed to read the component's revision.

SHOW REVISION SENSED has the options listed below. The default is ALL.

- ALL - displays sensed revisions for all components.
- KERNEL - displays the revisions of all top-level components groups sensed with the SENSE command. These component groups include console, CPU, MCL, microcode, and NBI.
- CONSOLE - displays revisions of the console sensed with the SENSE command and/or calculated by the VERIFY REVISION command.
- CPU(S) - displays revisions of CPU(s) sensed with the SENSE command.
- MCL - displays revisions of the memory controller sensed by the SENSE command and/or calculated by the VERIFY REVISION command.
- MICROCODE - displays revisions of the microcode that were remembered when the microcode files were loaded.
- NBI - displays revisions of the NBI sensed by the SENSE command and/or calculated by the VERIFY REVISION command.

SHOW REVISION

Examples

The following examples for the 8800, 8700, and 8550 are shown with sample machine responses. Machine response for the 8500 is the same, except the DEC component is F1015 instead of F1007.

1. \>>>**SHOW REVISION HISTORY ALL**

```
                REVISION HISTORY <file rev 3>
                KERNEL Top Revisions
Component        Rev 10  Rev 11  Rev 12
-------------   |min,max|min,max|min,max
CPU Top Rev     |008,008|008,008|009,009
MCL Top Rev     |006,007|006,007|008,008
NBI Top Rev     |003,003|003,003|004,004
Console Top     |006,006|007,007|007,007
Microcode Top   |006,006|007,007|007,007

                CPU Top Revisions
Component        Rev 7   Rev 8   Rev 9
-------------   |min,max|min,max|min,max
F1009 (WCS)     | X , X | X , X | X , X
F1008 (SEQ)     | A , A | A , A | A , A
F1007 (DEC)     | A , A | A , A | A , A
F1006 (CCS)     | J , K | F , K | F , K
F1005 (ADP)     | D , E | D , E | D , E
F1004 (SLCO)    | A , A | A , A | A , A
F1003 (SLC1)    | A , A | A , A | A , A
F1002 (SHR)     | A , A | A , A | A , A
F1010 (CLK)     | B , D | B , D | B , D
CPU Backplane   | D , D | D , D | D , D

                MCL Top Revisions
Component        Rev 7   Rev 8
-------------   |min,max|min,max
MCL hardware    | F , H | J , J

                NBI Top Revisions
Component        Rev 2   Rev 3   Rev 4
-------------   |min,max|min,max|min,max
NBIAO           | B , B | A , B | C , D
NBIAO-NBIBO     | C , C | A , C | A , C
NBIAO-NBIB1     | C , C | A , C | A , C
NBIA1           | B , B | A , B | C , D
NBIA1-NBIBO     | C , C | A , C | A , C
NBIA1-NBIB1     | C , C | A , C | A , C
```

```
                CONSOLE Top Revisions
Component        Rev 5   Rev 6   Rev 7
-------------   |min,max|min,max|min,max
Console prog    |036,036|003,003|001,001
RTI driver      |004,004|004,004|005,005
P/OS software   |002,002|002,002|002,002
EMM firmware    |067,067|068,068|068,068

                MICROCODE Top Revisions
Component        Rev 5   Rev 6   Rev 7
-------------   |min,max|min,max|min,max
WCS microcode   |199,199|216,216|220,220
SDF microcode   |186,186|186,186|186,186
IBD microcode   |142,142|216,216|216,216
CCS microcode   |010,010|010,010|010,010
```

VAX 8800 sample machine response. VAX 8700/8550/8500 machine response contains one CPU column and omits the NBIA0 information.

2. >>> **SHOW REVISIONS SENSED ALL**

```
                SENSED REVISIONS
Component         Left cpu      Common        Right cpu       Comments
-------------     -----------   -----------   -----------     ---------------
KERNEL Top      =               unavailable                   Not calculated
CPU Top Rev     = closest to 9                closest to 9    CPUS revs match
 F1009 (WCS)    =  S                           S
 F1008 (SEQ)    =  X                           X
 F1007 (DEC)    =  X                           X
 F1006 (CCS)    =  D                           D
 F1005 (ADP)    =  C                           B
 F1004 (SLC0)   =  X                           X
 F1003 (SLC1)   =  X                           X
 F1002 (SHR)    =  X                           X
 F1010 (CLK)    =                X
 CPU Backplane  =                D
MCL Top Rev     =               unavailable                   Not calculated
 MCL hardware   =               unavailable                   Not sensed
NBI Top Rev     =               unavailable                   Not calculated
 NBIA0          =               unavailable                   Not sensed
 NBIA0-NBIB0    =               unavailable                   Not sensed
 NBIA0-NBIB1    =               unavailable                   Not sensed
 NBIA1          =               unavailable                   Not sensed
 NBIA1-NBIB0    =               unavailable                   Not sensed
 NBIA1-NBIB1    =               unavailable                   Not sensed
Console Top     =               unavailable                   Not calculated
```

```
 Console prog  =              001
 RTI driver    =              004
 P/OS software =              002
 EMM firmware  =              067
Microcode Top  = unavailable                 unavailable
 WCS microcode = 227                         227
 SDF microcode = 186                         186
 IBD microcode = 216                         216
 CCS microcode = 010                         010
 UWCS ucode    = unavailable                 unavailable     Not sensed
```

VAX 8800 sample machine response. VAX 8700/8550/8500 response is similar, except there is just one CPU column, and omitting the NBIA0 information.

SHOW STATUS

Displays information about the console state.

Format

SHOW STATUS

Description

SHOW STATUS displays some of the same information about the CPU(s) as SHOW CPU. In addition, the following information is displayed:

- The state of the power system (ON, OFF, STANDBY).
- The state of the battery backup (MISSING/PRESENT, ENABLED /DISABLED).
- The state of the CPU clock (ON, OFF) and the clock period. The clock period is not usually an integral number of nanoseconds.
- The state of the remote port, which has the following possibilites:

  Remote Console: Enabled/Disabled
  Remote User: Enabled/Disabled
  Remote Monitoring: Enabled/Disabled
  Remote Modem: Enabled/Disabled
  Remote Carrier: Present/Absent

- The state of the printer, as specified with ENABLE and DISABLE PRINTER. Valid states are:

  Enabled on-line
  Enabled off-line
  Disabled on-line
  Disabled off-line

  The printer does not print, even if it is enabled, if it has transmitted XOFF. (If that happens, the Ready light on the printer should go off).

- Program Mode settings: this shows the IDs specified with the SET TERMINAL command for the local and remote ports.

Examples

1. >>>**SH S**

```
  CPU   STATE    UMATCH     FLAGS/STATUS -- 8-JAN-86 16:01:39
>LEFT  HALTED   SYNC %X316F AVAILABLE SELECTED(DUAL) PRIMARY NEXT_PRIMARY
 RIGHT HALTED   SYNC %X0000 AVAILABLE SELECTED(DUAL) SECONDARY
Automatic Poweron:ENABLED; Restart:DISABLED; Boot:DISABLED
Power:ON.  BBU:DISABLED,MISSING.  Clock:ON; CLOCK PERIOD = %D44.94 ns
Printer ENABLED, ONLINE.  Remote carrier:ABSENT.
Remote user:DISABLED; console:DISABLED; modem:DISABLED; monitoring:DISABLED
Program Mode settings:  LOCAL=OPA0      REMOTE=OPA5
```

VAX 8800 machine response shows status of VAX CPUs; also shows console as disabled as well as status of printer, remote carrier, remote user, modem, remote monitoring.

2. >>>**SH S**

```
STATE    UMATCH     FLAGS/STATUS -- 8-JAN-86 16:01:39
HALTED   SYNC %X0000 AVAILABLE
Power:ON.  Clock:ON; CLOCK PERIOD = %D44.94 ns
Printer ENABLED, ONLINE.  Remote carrier:ABSENT.
Remote user:DISABLED; console:DISABLED; modem:DISABLED; monitoring:DISABLED
```

VAX 8700/8550/8500 machine response shows status of VAX CPU; also shows console as disabled as well as status of printer, remote carrier, remote user, modem, remote monitoring.

SHOW TIME

Displays the current Console Operating System (P/OS) date and time.

Format

SHOW TIME

Examples

1.
```
>>>SH T
10-DEC-1985 15:39:16
```

2.
```
>>>SHOW TIME
10-DEC-1985 15:39:16
```

START

Starts execution of either macrocode or microcode.

Format

START *address*

Command Qualifiers	**Defaults**
/CONTROLSTORE or /WCS	None

Description

START applies to selected and available CPU(s).

If the /CONTROLSTORE or /WCS qualifier is not entered, START executes macrocode. If the /CONTROLSTORE or /WCS qualifier is provided, START begins execution of microcode.

Restriction

The clocks must be stopped prior to issuing this command. (See the SHOW STATUS and/or SET CLOCK commands.)

Command Parameter

address

Specifies the address at which execution begins. The address is expressed as a numeric literal, the special address symbol @, or the symbolic name CSM_INIT (see Appendix B). When the address is specified as a numeric literal, the macro-PC is set to the specified address, and a CONTINUE command is automatically executed.

When the @ symbol is used as the starting address, the data pointed to by the last EXAMINE or DEPOSIT command is taken as the address.

If no address is specified, a CONTINUE command is executed, and execution begins at the current macro-PC.

Command Qualifiers

/CONTROLSTORE or /WCS

Execution begins at the Writable Control Store (WCS). The micro-PC is loaded into the micro-PC register, and microcode execution is initiated by starting the clock. The starting address CSM_INIT is used with the /CONTROLSTORE or /WCS qualifiers. Starting the microcode at this address initializes the micromachine to a known state.

Examples

1. ```
   >>> S 1600
   ```

   Starts execution at address 1600.

2. ```
   >>> START 1000
   ```

   Starts execution at address 1000.

3. ```
   >>> SET CLOCK OFF
   >>> S/C CSM-INIT
   ```

   Turns off the CPU clock and starts execution at the CSM_INIT address.

4. ```
   >>> EX SP
   >>> START
   ```

TEST

Initiates the microdiagnostic test sequence and changes the mode to Micromonitor.

Format

TEST

Command Qualifiers	**Defaults**
/COMMANDS	None

Description

The TEST command (issued without the qualifier) causes a series of diagnostic tests to proceed automatically under the control of the Micromonitor. The Micromonitor commands are described in Chapter 6. Diagnostics are included if you have a DIGITAL Field Service contract or a Diagnostic License. If diagnostics are not available, an error message appears.

Micromonitor commands ignore the "selected" and "available" attributes and apply to the CPU(s).

The Micromonitor command EXIT returns control to the console prompt level >>> .

Command Qualifiers

/COMMANDS

Starts the Micromonitor, which displays the MIC> prompt and waits for a Micromonitor command (see Chapter 6).

Examples

1.
```
>>>T/C
MIC>
```

2.
```
>>>TEST/COMMANDS
MIC> DIAGNOSE
MIC> EXIT
```

UNJAM

Produces a stable state on the Nautilus Memory Interconnect (NMI).

Format

UNJAM

Description

UNJAM clears all NMI sequences without affecting the NMI error conditions.

CAUTION

The UNJAM command can cause data in transit to be lost.

Restriction

The clocks must be running.

Examples

1. `>>>U`

2. `>>>UNJAM`

VERIFY

Tests communications with the EMM, communication over the RTI, positioning of modules, and revision levels.

Format

VERIFY *(option)*

Description

The VERIFY command requires one of the options listed in the following VERIFY commands.

Command Options

EMM

MODULE PLACEMENT

REVISION

RTI

VERIFY EMM

Executes EMM loopback and confidence tests and obtains the EMM revision level.

Format

VERIFY EMM

Description

VERIFY EMM is the only way to mark the Environmental Monitoring Module as available. The EMM can be marked unavailable from a communication failure. If the EMM is not available, the power-up sequence fails. After power-up completes successfully, an EMM failure inhibits console interaction with the EMM.

Examples

1. `>>>V E`

2. `>>>VERIFY EMM`

VERIFY MODULE PLACEMENT

Verifies that each CPU module is in the correct slot and marks the CPU(s) available.

Format

VERIFY MODULE PLACEMENT

Description

VERIFY MODULE PLACEMENT marks a CPU as available. (Several other console commands that detect hardware errors can mark a CPU as unavailable.) On the VAX 8800, both CPUs are checked. Each CPU module is verified as being in the correct slot. Note that the CPU-available states are preserved across console-only power failures.

Execute this command prior to the SENSE REVISION CPU command.

Restriction

The CPU clocks must be stopped. (See the SHOW STATUS and/or SET CLOCK commands.)

Examples

1. `>>> V M P`

2. `>>> VERIFY MODULE PLACEMENT`

VERIFY REVISION

Determines the revision group's top revision and checks the compatibility of components in the group.

Format

VERIFY REVISION *(option)*

Description

The revision-control commands automate some revision management tasks, prevent problem propagation, and provide remote access to revision data. VERIFY REVISION checks for compatibility, determines top revision levels, reports incompatibilities, and reports possible corrections. Execute the SENSE REVISION command before executing a VERIFY REVISION command. Microcode is implicitly sensed during load procedures.

VERIFY REVISION requires one of the options listed below.

Command Options

CONSOLE
Determines the top revision of the console and displays incompatibilities among the console program, RTI driver, P/OS, and EMM firmware, as well as necessary corrections.

CPU
Determines the top revision level of the CPU(s). Any incompatibilities among the CPU backplane, the CPU modules, and the clock module, as well as any necessary corrections, are displayed. This option applies to available and selected CPU(s).

KERNEL
Determines the kernel revision level of the machine (top revision level) by checking the compatibility of CPU, MCL, NBI, CONSOLE, and MICROCODE top revisions. Use this command after the other VERIFY commands.

MCL
Determines the top revision level of the memory controller.

MICROCODE
Determines the top revision level of the microcode. Incompatibilities among the WCS, SDF, IBDECODER, and CCS microcode, as well as necessary corrections, are displayed.

NBI
Determines the top revision of the NBIA and NBIB modules. Incompatibilite and necessary corrections are displayed.

Examples

1. `>>>V R C`

   Determines the top revision level of the CPU(s), displaying incompatibilities.

2. `>>>VERIFY REVISION CPU`

   See Example 1.

3. `>>>V R MI`

   Determines the top revision level of the microcode, displaying any incompatibilities.

4. `>>>V R CO`

   Determines the top revision level of the console, displaying any incompatibilities.

VERIFY RTI

Executes Real Time Interface module loopback tests.

Format

VERIFY RTI

Examples

1. `>>>V R`

2. `>>>VERIFY RTI`

WAIT

Causes the console to wait for selected and available CPU(s) to halt. Used in command procedures after the HALT command.

Format

WAIT

Examples

1. `>>>W`

2. `>>>WAIT`

Chapter 6

Micromonitor Commands

This chapter describes the Micromonitor commands that are available in Micromonitor mode for the VAX 8800, 8700, 8550, and 8500 diagnostics program. Commands are issued in Micromonitor mode, which you enter by using the TEST/COMMANDS command. The user prompt is:

MIC>

The Micromonitor commands are valid if you have a DIGITAL Field Service contract or a DIGITAL VAX Diagnostic License for one of the models. (See the *Diagnostic User's Guide* for more information.) If the Micromonitor commands are not valid, they produce error messages.

The microdiagnostics test the following components:

- CPU(s)
- CPU clock/console interface
- Nautilus Memory Interconnect (NMI)
- Memory
- Nautilus Backplane Interconnect (NBI)

Power-up microdiagnostics are included in the SYSINIT.COM file.

CLEAR FLAGS

Clears flags that have been set with the SET FLAGS command (see SET FLAGS).

Format

CLEAR FLAGS *[flag[,flag,...]]*

Description

The CLEAR FLAGS command has the options listed below. If no options are used, all Micromonitor flags are cleared. If both QUICK_VERIFY and EXHAUSTIVE are entered, the default is QUICK_VERIFY.

Command Options

DEBUG_MESSAGES
Clears the flag that prints debug messages embedded in a microdiagnostic.

EXHAUSTIVE
Clears the flag that runs all tests.

HALT
Clears the flag that causes the Micromonitor to halt and issue a prompt when an error occurs during the execution of diagnostics. The operator can then issue Micromonitor commands. (Default = set.)

QUICK_VERIFY
Clears the flag that causes short, fast tests to run.

TRACE
CLEAR FLAGS TRACE has two options: SECTIONS and TESTS. CLEAR FLAGS TRACE SECTIONS clears the flag that causes the section name and revision level to be displayed as each new microdiagnostic section is executed. CLEAR FLAGS TRACE TEST clears the flag that causes the test name and a short description of each test to be displayed as each new test is being executed. If neither option is provided, both flags are cleared.

WCS_STEP
Clears the flag that enables you to use the MICROSTEP command to step through the WCS microdiagnostic.

Examples

1. `MIC> CL FLAGS TR S`

   Clears the section/revision level flag.

2. `MIC> CLEAR FLAGS TRACE TEST`

   Clears the test name flag.

3. `MIC> CL F H`

   Clears the halt flag.

4. `MIC> CLEAR FLAGS EXHAUSTIVE`

   Clears all test flags.

CONTINUE

Continues the execution of a microdiagnostic that failed or was halted by CTRL/C or an error.

Format

CONTINUE

Examples

1. `MIC> C`

2. `MIC> CONTINUE`

DIAGNOSE

Executes specified diagnostics.

Format

DIAGNOSE

Command Qualifiers	**Defaults**
/CONTINUE	None
/PASS:count	
/SECTION:section1[:section2]	
/TEST:test1[:test2]	

Description

The standard set of microdiagnostics can be run by executing the DIAGNOSE command with no qualifiers. Specific tests can be selected, using the SECTION and TEST qualifiers.

An error results unless you have the DIGITAL Field Service contract or an appropriate Diagnostic License.

Reboot the system after the DIAGNOSE command executes.

Command Qualifiers

/CONTINUE

The qualifiers can be used together as follows:

- /C/S:section

  This combination starts at test 1 of the specifed section and executes all tests in the section, as well as the remaining sections.

- /C/S:section/T:test

  These qualifiers direct the Micromonitor to start at the specified test in the specified section and to execute the remaining tests in that section, as well as all tests in the remaining sections.

- /C/T:test

  This combination starts at the specified test in the current section and executes the remaining test in that section. The /CONTINUE and /PASS qualifiers are mutually exclusive.

/PASS: count

Executes the microdiagnostics a specified number of times. A pass count of zero causes infinite passes. The maximum value that can be specified is 32767.

/SECTION:section1:[section2:]

Causes execution of all tests in the first section and in all subsequent sections until the second section specified is reached. If only one section name is specified, tests in that section are executed. Valid section names are EZKPA and EZKPB. (These are the power-up tests and are included with all systems.)

A section is a collection of microdiagnostic tests.

/TEST:test1:[test2]

Executes microdiagnostics, starting with the first test specified and continuing to the second test specified (test1 and test2 are numeric literals). If only one test number is specified, only that test is executed. If the test number is not in the section, an error results.

To test a series of tests sequentially, specify the first test number, followed by the number of the last test to be executed.

Examples

1. `MIC> DI`

2. `MIC> DIAGNOSE/S:EZKPA`

   Executes all tests in the section EZKPA.

3. `MIC> DI/T:1:5`

   Executes tests 1–5 in the current section.

4. `MIC> DIAGNOSE/P:2/S:EZKPA/T:2`

   Executes test 2 in section EZKPA twice.

EXIT

Returns the console program from the Micromonitor to the Console Command Language.

Format

EXIT

Examples

1. `MIC> EXI`

2. `MIC> EXIT`

LOOP

When an error occurs, the Micromonitor loops on the error.

Format

LOOP

Restrictions

LOOP is only valid immediately after an error has occurred. Otherwise, an error message is returned by the Micromonitor, and no action is taken.

Examples

1. `MIC> LOOP`

SELECT

Specifies which memory arrays to test in the memory diagnostics.

Format

SELECT *ARRAY=(array[,array,...] | ALL | NONE)*

Description

The SELECT command requires one of options listed below.

Command Options

ARRAY=(array [,array ,...])
Selects the specified memory arrays for testing in the memory diagnostics.

Array=ALL
Selects all memory arrays (0–7 for the VAX 8800, 0–4 for the VAX 8700 /8550/8500).

Array=NONE
Selects no memory arrays. Tests only the memory controller.

Examples

1. `MIC> SEL ARRAY=ALL`

   Selects all memory arrays for testing.

2. `MIC> SEL A = 5,2`

   Selects memory array 5.2 for testing.

3. `MIC> SEL ARRAY = NONE`

   Tests controller but not memory arrays.

SET FLAGS

Sets Micromonitor flags used to control execution of microdiagnostics.

Format

SET FLAGS *[DEFAULT | flag[,flag,...]]*

Description

The SET FLAGS command has the options described below. If no option is specified, SET FLAGS DEFAULT is used.

Command Options

DEBUG_MESSAGES

Prints debug messages that are embedded in a microdiagnostic, for instance, turning on clocks, turning on stalls, and locating RAMs.

DEFAULT

Sets the defaults for the SET FLAG options as follows:

DEBUG_MESSAGES	Disabled
EXHAUSTIVE	Disabled
HALT	Enabled
QUICK_VERIFY	Disabled
SBE_THRESHOLD	512
TRACE SECTIONS	Enabled
TRACE TESTS	Enabled
WCS_STEP	Disabled

EXHAUSTIVE

Sets the flag to run all diagnostic tests.

HALT
Causes the Micromonitor to halt and issue a prompt when an error occurs during the execution of diagnostics. The operator can then issue Micromonitor commands. (Default = set.)

QUICK_VERIFY
Executes a set of short, fast tests.

SBE_THRESHOLD=maximum tolerance
Parameter is used by memory diagnostics. SET FLAGS SBE_THRESHOLD specifies the maximum number of single bit memory errors "tolerated" before an error is reported. (Default=1.) To clear this flag, set SBE_THRESHOLD to 0. The maximum value is 32767.

TRACE [SECTION, TEST]
SET FLAGS TRACE SECTION displays the section name and revision level as each new microdiagnostic section is executed. SET FLAGS TRACE TEST displays the name and a short description of each test. If neither SECTION nor TEST is specified with the TRACE option, the default is both.

WCS_STEP
Allows you to use the MICROSTEP command to step through the WCS microdiagnostic.

Examples

1. `MIC> SE F T S`

   Enables TRACE SECTION flag.

2. `MIC> SET FLAGS TRACE TEST`

   Displays the name and provides short description of each test.

3. `MIC> SE F H`

   Enables the HALT flag.

4. `MIC> SET FLAGS EXHAUSTIVE`

5. `MIC> SE F S=2`

   Sets the maximum number of tolerable single-bit errors to 2.

SHOW MICMON

Shows the state of all Micromonitor flags.

Format

SHOW MICMON

Description

Besides showing the state of Micromonitor flags, SHOW MICMON shows how RAMs are loaded for diagnostics. It also shows the currently loaded section and its revision. If no section is loaded, the command reports "None". See the *Diagnostics User's Guide* for more information.

Examples

1.
```
MIC> SH M

Micromonitor Section:      NONE.
Micromonitor flags set:    Trace Section, Trace Test, Halt
Memory arrays selected:    0,1,2,3,4,5,6,7
Memory SBE threshold:      %D512
Rams loaded in LEFT CPU:   None.
Rams loaded in Right CPU:  None.
```

   VAX 8800 machine response shows state of flags, arrays selected, and RAMs loaded in both CPUs.

2. MIC> **SHOW MICMON**

```
Micromonitor Section:     NONE.
Micromonitor flags set:   Trace Section, Trace Test, Halt
Memory arrays selected:   0,1,2,3,4
Memory SBE threshold:     %D512
Rams loaded in CPU:       None.
```

VAX 8700/8550/8500 machine response shows state of flags, arrays selected, and RAMs loaded in the CPU.

VERIFY MODULE

Runs all pertinent diagnostics for the specified module.

Format

VERIFY MODULE *module-name or (option)*

Description

The VERIFY MODULE command is implemented as a command procedure. See the *Diagnostic User's Guide* for more information.

Restrictions

VERIFY MODULE must be the last command in the file.

Command Options

module-name
VERIFY MODULE can be used to test the following CPU modules: ADP, CCS, CLK, DEC, EMM, MAR, MCL, RTI, SEQ, SHR, SLC0, SLC1, and WCS.

Examples

1. `MIC> V M DEC`

   Tests DEC module.

2. `MIC> VERIFY MODULE CCS`

   Tests CCS module.

Appendix A

Console Command Syntax

This appendix displays the syntax of each console command, including all parameters, qualifiers, and options. The commands are listed alphabetically and use the conventions described in the Preface.

@file-name

BOOT [device-name]

[/R5:data]

CLEAR

ACCUMULATOR
RESTART_FLAGS
SCREEN
SOMM
TOMM

CONTINUE

DEPOSIT address-identifier deposit-data

[address-space-qualifier]
[size-qualifier]
[/**N**EXT=value]

address-space-qualifiers	size-qualifiers	address-identifier
/**CA**CHE	/**B**YTE	GPR mnemonic
/**C**ONTROLSTORE	/**L**ONGWORD	
/**G**PR	/**Q**UADWORD	IPR mnemonic
/**I**PR	/**W**ORD	numeric literal
/**P**HYSICAL		PSL
/**S**DF		UPC
/**TB**		+
/**T**EMP		-
/**V**IRTUAL		.
/**WCS**		@
		*

DISABLE

- **A**UTO **B**OOT
- **A**UTO **P**OWERON
- **A**UTO **R**ESTART
- **L**OCAL **C**ONSOLE
- **P**RINTER
- **R**EMOTE **C**ONSOLE
- **R**EMOTE **M**ODEM
- **R**EMOTE **MON**ITORING
- **R**EMOTE **U**SER
- **S**ECONDARY (VAX 8800)

ENABLE

- **A**UTO **B**OOT
- **A**UTO **P**OWERON
- **A**UTO **R**ESTART
- **L**OCAL **C**ONSOLE
- **P**RINTER
- **R**EMOTE **C**ONSOLE
- **R**EMOTE **M**ODEM
- **R**EMOTE **MON**ITORING
- **R**EMOTE **U**SER

SECONDARY

EXAMINE address-identifier

/[NEXT]
[address-space-qualifier]
[size-qualifier]

address-space-qualifiers	size-qualifiers	address-identifier
/GPR	/BYTE	GPR mnemonic
/IPR	/LONGWORD	IPR mnemonic
/PHYSICAL	/QUADWORD	numeric literal
/SDF	/WORD	PSL
/TEMP		UPC
/VIRTUAL		+
		-
		.
		*
		@

EXIT

FIND

/MEMORY
/RPB

HALT

HELP [help-topic]

IF [NOT] state-bit-name THEN console-command

INITIALIZE

/CREATE_CONFIGURATION_FILE
/MEMORY

LINK

LOAD (file-name or /DATA=numeric-literal)

/**CA**CHECONTROL
/**C**ONTROLSTORE or /**W**CS
/**I**BDECODER
/**M**AINMEMORY
/**S**DF
/**ST**ART=numeric-address
/**U**WCS

MICROSTEP [step-count]

NEXT [next-count]

PERFORM

POWER

OFF
ON
STANDBY

PROBE [VBUS] vbus-address accumulator-bit

REPEAT command

BOOT	MICROSTEP
CLEAR	PROBE
CONTINUE	SELECT
DEPOSIT	SENSE
DIAGNOSE	SET
EXAMINE	SHOW
FIND	START
IF	TEST
INITIALIZE	UNJAM
LOAD	VERIFY
LOOP	WAIT

SENSE [REVISION]

- **CO**NSOLE
- **C**PU(S)
- **M**CL
- **N**BI

SET

- **CL**OCK
  - **F**AST
  - **N**ORMAL
  - **OF**F
  - **ON**
  - **S**LOW
- CPU (VAX 8800)
  - **B**OTH
  - **CURRENT_P**RIMARY
  - **CURRENT_S**ECONDARY
  - **L**EFT
  - **NEXT_P**RIMARY
  - **NEXT_S**ECONDARY
  - **R**IGHT
- **D**EFAULT [address-space, size, radix]
- **E**MM
- **MA**RGINS (option) [regulator-list]
  - **H**IGH
  - **L**OW
  - **N**ORMAL
- NEXT_PRIMARY (option)–VAX 8800
  - **L**EFT
  - **R**IGHT
- **NO**VERIFY
- **R**ELOCATION base-address
- **S**OMM upc-address
- **SY**NC upc-address

TERMINAL (option)

PROGRAM
OPA0:
OPA4:
OPA5:

TOMM upc-address
VERIFY

SHOW (option)

ACCUMULATOR
CPU
DEFAULTS
LOGFILE
MICMON
POWER
REVISION (option)

HISTORY
SENSED

ALL
CONSOLE
CPU(S)
KERNEL
MCL
MICROCODE
NBI

STATUS
TIME

START address

/**C**ONTROLSTORE or /**W**CS

TEST

/**C**OMMANDS

UNJAM

VERIFY (option)

EMM
MODULE **P**LACEMENT

REVISION

- **CO**NSOLE
- **C**PU
- **K**ERNEL
- **M**CL
- **MI**CROCODE
- **N**BI

RTI

WAIT

MICROMONITOR COMMANDS

CLEAR FLAGS [flag[,flag,...]] (default: clear all flags)

DEBUG_MESSAGES
EXHAUSTIVE
HALT
QUICK_VERIFY
TRACE

SECTIONS
TESTS

WCS_STEP]

CONTINUE

DIAGNOSE

/CONTINUE
/PASS
/SECTION:section1[:section2]
/TEST1[:test2]

EXIT

LOOP

SELECT (ARRAYS = (array(#)[,array(#),...] | ALL | NONE)

SET FLAGS [DEFAULT | flag[,[flag,...]]

DEBUG_MESSAGES
EXHAUSTIVE
HALT
QUICK_VERIFY
SBE_THRESHOLD=numeric-literal
TRACE

SECTIONS
TESTS

WCS_STEP

SHOW MICMON

VERIFY MODULE module-name

Appendix B

Numeric and Address Data

This appendix provides the numeric literals and symbolic addresses referred to in Chapters 5 and 6. The state bit definitions and the Internal Processor Registers descriptions are also listed.

B.1 Numeric Literals

A numeric literal can be specified in any base by using a radix specifier. The default radix is set with the SET DEFAULT command and displayed with the SHOW DEFAULTS command.

Table B–1: Radix Specifiers

Radix Specifiers	Definition
%O→	Numeric literal interpreted and verified as OCTAL.
%X→	Numeric literal interpreted and verified as HEXADECIMAL.
%D→	Numeric literal interpreted and verified as DECIMAL.
%B→	Numeric literal interpreted and verified as BINARY.

The maximum size of a numeric literal depends on its context. For example, micro-PC addresses may not exceed 14 bits, physical and virtual addresses may not exceed 32 bits, and the size of the EXAMINE or DEPOSIT commands data may not exceed the size qualifier.

The default radix is initially hexadecimal. Using a radix specifier changes the default radix.

B.2 Symbolic Addresses

The console supports a set of symbolic addresses to facilitate use of the EXAMINE and DEPOSIT commands. The symbolic addresses are expressed as mnemonics, symbolic IPR names, and special addresses. When symbolic IPR and GPR names are included in a command, the command qualifiers /IPR and /GPR must be omitted. When a mnemonic or symbolic address is specified, it is translated into a numeric address value, an implicit size, and an implicit address space.

Table B–2: Mnemonic Addresses

Mnemonic Addresses	Definitions
PSL	Processor status longword
R0-R11	General registers
AP	Argument pointer
FP	Frame pointer
SP	Stack pointer
PC	Program counter
UPC	Micro-PC

B.2.1 Internal Processor Registers (IPRs)

The internal processor registers for the VAX 8800, 8700, 8550, and 8500 are listed below with brief descriptions. Numeric addresses are in hexadecimal.

Table B–3: Internal Processor Registers

Mnemonic	Numeric Address	Description
ASTLVL	13	AST level register
CBER*	A0	C-Box error register
COR	85	Cache on Register: 1=ON, 0=OFF
EBER*	B0	E-Box error register
ESP	01	Executive stack pointer
IBER*	C0	I-Box error register
ICCS	18	Interval clock control and status reg
ICR	1A	Interval count register
INOP	81	When written to, posts interrupt to other CPU
IPL	12	Interrupt priority level register
ISP	04	Interrupt stack pointer
KSP	00	Kernel stack pointer
MAPEN	38	Memory mapping enable register
MCSTS	26	Machine check status
NICR	19	Next interval count register
NMIEAR	84	NMI error address register
NMIFSR	82	NMI fault/status register
NICTRL	80	NMI interrupt control
NMISILO	83	NMI silo data
P0BR	08	P0 base register
P0LR	09	P0 length register
P1BR	0A	P1 base register
P1LR	0B	P1 length register
PCBB	10	Process control block base register
PME	3D	Performance monitor enable register
REVR1	86	Revision control
REVR2	87	Revision control
SBR	0C	System base register

Table B–3 (Cont.): Internal Processor Registers

Mnemonic	Numeric Address	Description
SCBB	11	System control block base register
SID	3E	System identification register
SIRR	14	Software interrupt request register
SISR	15	Software interrupt summary register
SLR	0D	System length register
SSP	02	Supervisor stack pointer
TBCHK	3F	Translation buffer check register
TBIA	39	Translation buffer invalidate all reg
TBIS	3A	Translation buffer invalidate single reg
USP	03	User stack pointer

* May be accessed using the numeric literal only. Symbolic names are not supported.

Table B–4: Special Addresses

Special Addresses	Definitions
+	Increments the EXAMINE/DEPOSIT address. (Last referenced address plus data length in bytes.)
-	Decrements the EXAMINE/DEPOSIT address. (Location preceding last reference.)
* or .	Leaves the address unchanged. (Last address referenced.)
@	Copies the previous EXAMINE/DEPOSIT data into the address or data argument.
CSM_INIT	Microcode address to initialize the micromachine to a known state.

How much the address increments or decrements depends on the size qualifier.

B.3 Console State Bit Definitions

Table B–5: Console State Bit Definitions

Mnemonic	Meaning	Modified by
AUTO_BOOT	Reboot following power failure and error halt if restart fails	ENABLE / DISABLE AUTO_BOOT
AUTO_POWERON	Turn on VAX following power failure	ENABLE / DISABLE AUTO_POWERON
AUTO_RESTART	Restart following power failure or error halt	ENABLE / DISABLE AUTO_RESTART
BBU_STATUS*	Battery Backup working	Console code
CLOCK_RUNNING	Clock running	SET CLOCK ON /OFF
CPU_AVAILABLE	CPU has not failed any tests	Enable: VERIFY MODULE PLACEMENT. Disable: LOAD parity errors among other things.
DIAG_CSM_LOADED	Diagnostic console support microcode loaded	Micromonitor
EMM_STATUS	EMM checks out ok	Console software
HALTED	CPU halted	HALT or START command
LEFT_CPU_SELECTED*	CPU specified by SET CPU	SET CPU LEFT /RIGHT
MICMON_MODE	Micromonitor mode enabled	Enable: TEST Disable: EXIT
POWER_STATUS	VAX power on	POWER ON/OFF
POWER_STANDBY	All power regulators are off except B and J	POWER ON/OFF

Table B–5 (Cont.): Console State Bit Definitions

Mnemonic	Meaning	Modified by
REMOTE_CONSOLE_ ENABLE	Remote console enabled	ENABLE / DISABLE REMOTE CONSOLE
REMOTE_MODEM	Remote modem enabled	ENABLE / DISABLE REMOTE MODEM
REMOTE_USER_ENABLE	Remote user enabled	ENABLE / DISABLE REMOTE USER
RIGHT_CPU_SELECTED*	CPU specified by SET CPU	SET CPU RIGHT /LEFT
SECONDARY_ENABLE*	Secondary CPU enabled for booting	ENABLE / DISABLE SECONDARY
$STATUS	Success or failure of last command	All commands, except IF
$VBUS	Last bit read from VBUS	PROBE

* VAX 8800 only.

B.4 General Purpose Registers (GPRs)

The VAX 8800, 8700, 8550, and 8500 General Purpose Registers are listed below.

Table B–6: General Purpose Registers (GPRs)

AP	12	Argument pointer
FP	13	Frame pointer
PC	15	Program counter
R0	00	Register 0
R1	01	Register 1
R2	02	Register 2
R3	03	Register 3
R4	04	Register 4

Table B–6 (Cont.): General Purpose Registers (GPRs)

R5	05	Register 5
R6	06	Register 6
R7	07	Register 7
R8	08	Register 8
R9	09	Register 9
R10	10	Register 10
R11	11	Register 11
SP	14	Stack pointer

B.5 Temporary Registers

Temporary registers (specific to microcode) are listed in Tables B–7 and B–8.

Table B–7: Readable and Writable Temporary Registers

10	WDR
11	STEMP12
12	STEMP13
13	STEMP14
14	CN.T0
15	CN.T1
16	CN.T2
17	CN.T3
18	CN.T4
19	MD0
1A	MD1
1B	MD2

Table B–7 (Cont.): Readable and Writable Temporary Registers

1C	MD3
1D	MD4
1E	MD5
2F	MD6
20	MM.MDR
21	CN.VA
22	BACK.UP.PC
23	shiftcntb
24	get x reg

B.5.1 Writable Only Temporary Registers

Table B–8: Temporary Registers - Writing Only

2E	Validate cache with 0s
2F	Initialize without clearing TB or cache.

Note that you cannot use symbolic names for reading/writing temporary registers.

Appendix C

Console Messages

This appendix lists and defines VAX console messages. Dual-CPU messages apply only to the VAX 8800.

C.1 Information Messages

AIR FLOW FAULT CONDITION CLEARED

EMM sensed that an air flow fault has gone away. Console reenables EMM's default mode monitoring and takes no further action.

Note: See other AIR FLOW FAULT entries.

AUTO POWERON DISABLED

The POWER OFF command saves the current setting of the AUTO_POWERON keyswitch and then disables it (the keyswitch). POWER ON restores the saved keyswitch setting.

BURST COMPLETE

Clock burst requested by MICROSTEP command completed.

CLOCK STOPPED

Clock status changed from running to stopped.

CSEQ PREVIOUSLY LOCKED HAS BEEN UNLOCKED

CSEQ MCA on the VAX Clock module detected a "console gone" condition (console-only power failure) and disabled communication with the VAX console. When power is restored to the VAX console, it attempts to reestablish the communication link through its RTI.

This message indicates that the VAX console was able to reestablish the communication link to VAX clock module. CSEQ LOCKED occurs because the console was unable to unlock it.

EMM RAM PARITY ERROR CHECKING DISABLED

Console found the EMM RAM parity checking circuits disabled after powering on the regulators and checking the EMM's status.

LOAD DONE, n BYTES LOADED STARTING AT ADDRESS xyz

Format of status string when microcode is loaded.

LOAD DONE, n WORDS LOADED, REV = m

Format of status string when load is completed.

LOAD TERMINATED BY CTRL/C

Operator interrupted the console LOAD command.

POWER ALREADY IN REQUESTED STATE

VAX power system already in the state requested by a POWER ON, OFF, or STANDBY command.

POWER IS NOW IN STANDBY MODE

As a result of the POWER STANDBY command, the VAX power has changed to STANDBY.

POWER IS NOW OFF

Power has unexpectedly changed, however, not because of a console command.

POWER IS NOW ON

As a result of the POWER ON command, the VAX power has changed to the ON state.

REMOTE CONSOLE MODE DISABLED

Response to the command DISABLE REMOTE CONSOLE. Remote user mode is still enabled.

REV HIGHER THAN TABLE ENTRY. HISTORY FILE OUT OF DATE.

The revision history file is out of date.

SHUTTING DOWN REMOTE PORT

Connection to the remote port, if established, is broken. The remote user and console modes are disabled.

STARTING UP REMOTE PORT IN REMOTE USER MODE

Remote console mode is disabled, but remote user mode is still enabled.

STOP ON MICROMATCH IN (LEFT/RIGHT) CPU

VAX micro-PC matched contents of Micromatch register.

n BLOCKS LOADED STARTING AT PHYSICAL ADDRESS

Format of status string when LOAD/MAINMEMORY completes.

C.2 Warning Messages

ACLO NOT DEASSERTED

EMM failed to deassert the ACLO signal after a POWER ON command was issued. The power system has a fault.

ACLO REMAINED DEASSERTED ON POWER OFF

Console found MPS regulator modules B and J powered off, DCLO asserted, but ACLO not asserted following a POWER OFF command.

AUTO POWERON REENABLED

The AUTO_POWERON command keyswitch setting saved by the POWER OFF command was restored by the POWER ON command.

BBU IS UNAVAILABLE (VAX 8800)

Console read the BBU available bit and found it deasserted, meaning that the Battery Backup Unit was unavailable.

CPU(S) BEING POWERED DOWN

Previous power-up sequence was interrupted; consequently, the console powered down the VAX to a known state.

CPU(S) ALREADY HALTED

A HALT command was issued, but the CPUs were already in a halted state.

CROWBAR FAULT LATCHED

EMM detected a crowbar fault.

CSEQ LOCKED; CONTROL OF CPU INHIBITED

CSEQ MCA on the VAX Clock module detected a "console gone " condition (console-only power failure) and disabled communication with the VAX console. When power was restored to the VAX console, it attempted to reestablish the communication link through its RTI.

The CPU ignored the console. Control functions attempted from the console were ignored by the VAX hardware.

DCLO NOT ASSERTED

DCLO signal was not asserted within 10 microseconds after a POWER OFF or POWER STANDBY command, signalling a power system failure.

DESIGNATED NEXT_PRIMARY NOT PRESENT; CHANGING PRIMARIES (VAX 8800)

The console changed the NEXT_PRIMARY to the working CPU. If the NEXT_PRIMARY CPU was not available (VERIFY MODULE PLACEMENT or LOAD parity error), a subsequent INIT and SET CPU CURRENT_PRIMARY directed the console to a missing or a broken CPU. To prevent this, the console changes NEXT_PRIMARY to the working CPU.

EMM FAILED MOD OK INTERRUPT TEST

EMM 5.5 interrupt test failed during the power-up verification, impacting the EMM's ability to detect and report power system problems.

EMM RAM PARITY ERROR CHECKING DISABLED

After powering on the regulators, the console checked EMM status for unusual conditions. EMM RAM parity checking was disabled during the check.

EMM REVISION NOT GUARANTEED TO BE COMPATIBLE

EMM revision did not match the level expected by the console application software.

EMM SELF TEST DONE, EMM RAM UNPREDICTABLE

Each time the EMM ran self-tests, its RAM contents were unpredictable and needed to be reinitialized by the console.

FORMAT ERROR IN MEMORY CONFIGURATION TABLE

MEMCONFIG.DAT did not have the correct format, possibly because it was edited incorrectly. Use INIT/CREATE to create a properly formatted file.

INVALID MARGINING LEVEL

Internal console error.

LEFT AND RIGHT CPU REVS MISMATCHED (VAX 8800)

VERIFY REVISION CPU computed different revision levels, based on different module revision levels.

LEFT CPU MARKED UNAVAILABLE (VAX 8800)

A problem in the left CPU made it unavailable. Operation continued with the right CPU, if available. Use the command VERIFY MODULE PLACEMENT to mark the CPU available again.

(LEFT) CPU NOT HALTED

The console command requires the CPU to be halted.

(LEFT) CPU STALLED

An error condition caused the left CPU to stall. The stall interval exceeded 5 seconds.

MICROCODE IN (LEFT/RIGHT) CPU EXECUTING AT UNKNOWN UPC

After loading microcode, the console tried to load a NOP into the microcode pipe. However, the NOP instruction had not been loaded.

MICROCODE INIT FUNCTIONS NOT PERFORMED

Microcode-assisted initialization functions were not performed by the console's INITIALIZE command.

MICROCODE REV DOES NOT MATCH PREVIOUS REV

The console compared the microcode revision stored in the file specified by the LOAD command to microcode revisions previously loaded. The revision of the microcode files changed since last loaded.

POWER MUST BE OFF TO TEST MOD NOT OK INTERRUPT

The POWER ON command routine that tests the EMM's MOD NOT OK interrupt found an MPS regulator on. All MPS regulators must be powered off. Use SHOW POWER to determine which regulators are on.

POWER NOT ON

VAX power is not on. Diagnostics cannot run.

PREVIOUS INITIALIZATION FAILED

During the power-up sequence, the console application found the SYS INIT IN PROGRESS bit set, which indicated that the previous sequence was interrupted prior to completion. System initialization is defined as the execution of SYSINIT.COM through the execution of a START command.

PRINTER OFFLINE

XOFF was received from printer, and XON was not received within 10 seconds.

PRINTER ONLINE

XON was received from the printer. (Printer is ready.)

REMOTE CONNECTION BROKEN

The modem detected a hang up.

REMOTE CONNECTION ESTABLISHED

The modem answered an incoming call.

RETRYING FIRST TIME INITIALIZATION

The previous system initialization failed and was attempted again.

REVISION HISTORY FILE CONTAINS BAD RECORDS.

The file REVHIST.DAT, containing revision history information, was out of date or corrupted.

REVISION OF EMM IS INCOMPATIBLE

Console application software determined during power up that the firmware revision of EMM did not match that expected.

RIGHT CPU MARKED UNAVAILABLE (VAX 8800)

A problem in the right CPU made it unavailable. Operation continued with the left CPU, if available. Use the command VERIFY MODULE PLACEMENT to mark the CPU available again.

RIGHT CPU NOT HALTED (VAX 8800)

The right CPU was not halted. The console command requires it to be halted.

RIGHT CPU STALLED (VAX 8800)

An error condition caused the right CPU to stall. The stall interval exceeded 5 seconds.

SYSTEM BEING POWERED DOWN

The flag indicating that system initialization was in progress was set and the VAX power was on. System power was turned off so that system initialization could proceed from a known state. To reexecute the system initialization procedure, enter

POWER ON
@SYSINIT

TOP REVISION IS UNAVAILABLE. CAN'T CHECK COMPATIBILITY.

A problem reading revisions of system components made verification at top-level revision uncertain.

UNEXPECTED 6.5 INTERRUPT ON EMM

The console's once-a-minute EMM polling routine detected an EMM problem, causing the console to reload the EMM parameters.

UNEXPECTED TRANSFER TO PC 0 ON EMM

The console's once-a-minute EMM polling routine detected an EMM problem, causing the console to reload the EMM parameters.

UNEXPECTED TRAP INTERRUPT ON EMM

The console's once-a-minute EMM polling routine detected an EMM problem, causing the console to reload the EMM parameters.

C.3 Error Messages

64KBYTES OF GOOD MEMORY NOT FOUND

The Console Support Microcode (CSM) failed to locate a 64KB block of contiguous error-free memory necessary to cold start the system. The operation was requested by the FIND/MEMORY command.

ACLO NOT ASSERTED

EMM failed to assert ACLO signal 5 MS after a POWER OFF or POWER STANDBY command. The console powered down the VAX due to a problem with the power system.

ADDRESS NOT LONGWORD ALIGNED

The address specifier in a DEPOSIT to Cache or TB was not longword aligned.

ADDRESS TOO LARGE

An illegal address was specified in a DEPOSIT or EXAMINE command.

AIR FLOW FAULT CONDITION PERSISTS, FORCING TOTAL OFF

The EMM still reported an air flow fault after the console's 5-minute timer elapsed. (See next message). Console initiated a total power off.

Note: Operator intervention is required to restore power. See description of the POWER ON command in Chapter 5.

AIR FLOW FAULT CONDITION PERSISTS, POWERING OFF REGULATORS

The EMM still reported an air flow fault after the console's 30-second timer elapsed. The console issued a POWER STANDBY command and set a 5-minute timer. (See previous message.)

AIR FLOW FAULT LATCHED

The EMM reported to the console that an air flow fault condition exists after the POWER ON command.

Problem may be due to:

- A restriction in the air cooling path
- A defective air mover

- Missing or misplaced dummy modules
- A malfunctioning air flow sensor
- A malfunctioning EMM

ALGORITHM ERROR

An internal console application software error occurred while trying to initialize and create the memory configuration file MEMCONFIG.DAT with the INIT/CREATE command.

(ALL AVAILABLE) CPU(S) NOT HALTED

CPU(s) were not halted. The UNJAM command requires all CPU(s) to be halted, because bus traffic may be corrupted.

ARRAY SPECIFIED NOT BETWEEN 0 AND 7 (VAX 8800)

An illegal memory array value was specified in the Micromonitor SELECT command.

ARRAY SPECIFIED NOT BETWEEN 0 AND 4 (VAX 8700/8550/8500)

An illegal memory array value was specified in the Micromonitor SELECT command.

ASSIGNMENT OF LUN TO RTI FAILED

The console application software failed to establish a logical connection with the RTI driver by assigning a Logical Unit Number to the device during console application initialization.

Error is fatal. Console application exited to P/OS.

AT LEAST ONE MODULE OK SIGNAL DEASSERTED

After the POWER ON command, the console checked the EMM MPS regulator status register and found that one or more regulators did not come up to power. The console powered off the VAX.

AUTOMATIC SHUTDOWN TIMEOUT IN PROGRESS

After the POWER ON command, the console checked the EMM status and found that the EMM initiated an automatic shutdown. A severe environmental error is indicated.

The ASD bit was set for one of the following conditions:

- "Red" zone temperature violation

- Air flow fault

If the problem persists for 2 minutes or more, EMM automatically forces a total off.

CAN'T OPEN REVISION HISTORY FILE

Console application software could not open the REVHIST.DAT file, which stores revision compatibility information.

CAN'T READ P/OS VERSION

During the SENSE command, QIO failed to determine the revision of P/OS.

CAN'T READ REVISIONS ON VBUS

SENSE command failed. VBUS not read successfully.

CAN'T SENSE REVISION

The console could not read the revision of the requested component.

CAN'T SHOW REVISION HISTORY

The console could not read the revision history file REVHIST.DAT.

CAN'T VERIFY REVISIONS

The console could not match the sensed revision to the revision history file REVHIST.DAT.

CANNOT RESUME FROM TRAP, DID NOT GET TO CSM BY TOMM

The console directed the microcode to release the trap, but a trap had not occurred.

CLOCK CRITICAL COMMAND ON OTHER CONSOLE—TRY AGAIN LATER

The remote port was active (see the REMOTE CONNECTION ESTABLISHED informational message) and in console mode. Commands that require clocks to be on or off or require the CPU to be halted cannot be accepted from the local port.

CLOCK MUST BE STOPPED

The command required the clock to be stopped.

CLOCK NOT STARTED

The SET CLOCK ON command failed to start the clock.

CLOCK NOT STOPPED

The SET CLOCK OFF command failed to stop the clock.

CLOCK ON ERROR

The console detected that the clocks were on when previously they had been off, and no intervening operation had started them.

CLOCKS COULD NOT BE SET CORRECTLY

The clock state could not be set as required to execute the specified command.

CLOCKS NOT STOPPED AT END OF BURST

The MICROSTEP command stepped through microinstructions by bursting the clock. The clocks did not stop at the end of a burst.

CLOCKS STOPPED FOR UNKNOWN REASON

The MICROSTEP command stepped through microinstructions by bursting the clock. At the end of the burst, the clock status was checked. The clock was stopped, but not as a result of burst completion or Stop On Micromatch.

COMMAND FILE LINK.COM NOT FOUND

The LINK.COM file could not be opened for read access; most likely because the file does not exist.

COMMAND FILE NOT FOUND OR IN USE

Command procedure could not be opened for read access. Possible reasons:

- File name specified incorrectly in an "@" command
- File locked or nonexistent
- File already opened by console or logical block server

COMMAND NOT EXECUTED

A console command was rejected due to a mismatch between the current clock state and that required by the command.

If the command is not being executed from a command procedure and the clock is not in the required state, the console asks if the clock state should be changed. If the answer is no, this message is displayed to confirm that the command was aborted before any system state was changed.

COMMAND NOT VALID WHILE REMOTE IS ACTIVE

Command issued from the local console was rejected to prevent the state from being manipulated by the remote console.

Example: The local console user deposits to registers that the remote user is examining. The remote port must exit from console mode to allow the local console to perform certain commands.

COMMAND ONLY VALID FROM LOCAL CONSOLE

A command was issued from the remote port that could only be executed from the local console.

COMMAND ONLY VALID FROM LOCAL OR REMOTE

A command was issued from a command procedure that could only be executed from the local or remote terminal.

COMMAND ONLY VALID IN MICMON MODE

A Micromonitor command was attempted in console mode.

To enter Micromonitor mode, issue the TEST/COMMAND from the console prompt >>> . Micromonitor mode is indicated by the *MIC>* prompt.

CONSOLE INTERNAL ERROR

An error has been encountered in the Console Operating System or one of its applications. Contact your DIGITAL Field Service Representative for assistance.

COULD NOT FIND VALID RPB

The Console Support Microcode (CSM) could not find a Restart Parameter Block in memory while servicing a FIND/RPB command.

The error usually causes a cold restart to be performed instead of a warm restart.

COULDN'T CLEAR COLD BIT

INITIALIZE/MEMORY command failed to clear the cold start bit in Memory Controller (MCL) register CSR3.

The cold start bit is set when the memory power is lost and indicates that memory should be initialized to generate good parity and ECC. Most likely cause of the error is an MCL problem.

COULDN'T CREATE CONFIGURATION TABLE

The memory configuration file MEMCONFIG.DAT could not be created by the INITIALIZE/CREATE command.

COULDN'T INITIALIZE MEMORY

The Console Support Microcode (CSM) returned an error status to the console request to initialize memory by writing good ECC. The initialization request was made as a result of an INITIALIZE/MEMORY command.

CPU HALTED

The CPU was halted, which prevented a console command from executing.

CPU NOT SCOPED

The LOAD command could not perform the requested operation. The VAX CPU(s) (one or two) are marked unavailable due to a failure in:

- Power-up diagnostic
- VAX module revision check
- VAX module keying check
- Previous parity error during LOAD

CPU NOT HALTED

The CPU was not halted, which prevented a console command from executing.

CSM COULD NOT COMPLETE COMMAND

The Console Support Microcode (CSM) failed to complete a console command.

CSM RETURNED BAD STATUS

The Console Support Microcode (CSM) returned a bad status at the completion of a console command.

DATA FIELD OVERFLOW

The LOAD command failed. The data specified in a command procedure or command argument was too large.

DATA NOT STORED IN LOGFILE

One or more buffered LOGFILE.DAT blocks could not be written while executing the EXIT command to return to P/OS.

DATA SIZE MUST BE LONGWORD

The data size specifier for DEPOSIT or EXAMINE of a GPR or IPR was not a longword.

DATA SIZE NOT LONG, QUAD

The data size specifier for DEPOSIT to Cache or TB was not in multiples of 4 bytes.

DATA TOO LARGE

The data loaded with the /DATA and /START switches was too large. Works the same as DEPOSIT.

DATA VALUE TOO LARGE

The data specified was too big for the specified location. Maximum data sizes (in bits) are:

DRAM 24
CCS 32
SDF 32
Memory 32

DCLO NOT DEASSERTED

The DCLO signal was not deasserted after a POWER ON command. The console powered off the VAX due to a problem with the power system.

DEPOSIT, EXAMINE ERROR REGISTER NOT ACCESSIBLE FROM CONSOLE

The console uses some registers to communicate to the VAX, RXDB, and TXDB registers. Reading these registers is invalid.

DMMREG.DAT NOT FOUND

Console application software could not locate the file containing the Default Mode monitoring parameters to be loaded into the EMM's DMMREG register. There are three files, DMMREGON.DAT, DMMREGSB.DAT, and DMMREGOF.DAT. Selection depends on the current power state.

EMM BROKEN OR NOT POWERED ON

The console could not read the EMM revision level during a power-up sequence to verify communication with the EMM. The console assumed it could not communicate with the EMM, exited the power-up sequence, and entered VAX console I/O mode (> > > prompt).

The possible cause is a bad or disconnected cable between console and EMM.

EMM COMMUNICATION ERROR, CHECK EMM

The RTI driver failed to complete the command sequence to the EMM (for reasons other than a time-out).

EMM DEFAULT PARAMETERS NOT LOADED

The console failed to load a complete set of EMM default mode parameters while attempting to initialize or reinitialize the EMM.

EMM DIAGNOSTICS FAILED

The VERIFY EMM command issued the EMM self-tests, and they failed.

EMM TIMED OUT

The EMM waited the time-out period and powered off. The possible causes are a bad connection or EMM power is off.

EMM IS UNAVAILABLE

The console failed to communicate with the EMM during the power-up sequence. The console marked the EMM unavailable and entered VAX console I/O mode.

EMM IS UNVERIFIED

The VERIFY EMM command was not executed prior to the POWER ON command, marking the EMM unavailable.

EMM REGISTER OPERATION NOT ALLOWED

An attempt was made to WRITE, BIS, or BIC an EMM read-only register; or attempted to READ a write-only register.

ENVIRONMENTAL MONITORING MESSAGE RECEIVED

An environmental problem or a power system exception was detected by the EMM.

Detailed information about an exception is reported in the following format:

```
DMCODE:nn <type> EXCEPTION REPORTED, dd-mmm-yy hh:mm:ss

<exception specific message>

AUTOMATIC SHUTDOWN IN PROGRESS (VAX 8800. For 8700/8550/8500, omit C and E
regs.)

REG: A B C D E F H J K L KAC LAC KEY
MOK: 0 0 0 0 1 1 1 1 1 1  1   0   1

SIG: AF1 BBF CBF AF2 ACL DCL PEN PER  (VAX 8800. For 8700/8550/8500, omit BBF
sig.) VAL:  0   0   1   0   1   0   1   1
```

NOTE

This report can also be requested with the SHOW POWER command.

Legend:

DMCODE—default mode message code

REG—MPS regulator

MOK —status of MODOK for regulator

SIG—status of selected EMM monitored signals, such as ACLO and DCLO

VAL —0, signal deasserted; 1, asserted

Additional information is provided for voltage and temperature exceptions and air flow faults.

- Voltage exceptions

  Transition from/to voltages
  Margin status

- Temperature exceptions

  Measured temperature
  Zone transition
  Compared limits

- Air Flow Faults

  No additional information provided
  Console sets timers to initiate staged air flow status checks and selectively power down as necessary

- Battery Back-up Faults (VAX 8800)

  Since console can take no corrective action, the BBU is disabled.

ERROR INITIALIZING MEMORY

The Console Support Microcode (CSM) returned bad status as a result of a console request to initialize memory (write good parity).

ERROR LOADING PHYSICAL MEMORY

The LOAD command failed to write to physical memory.

ERROR SENDING TO FILE SERVER PROGRAM

The console was unable to communicate with the logical block server (LBS). The LBS is a separate task image, which may have aborted.

ERRORS DETECTED

The test executed by the DIAGNOSE command detected errors.

FAILED PPI REGISTER TEST

The power-up test of Programmable Peripheral Interface (PPI) registers on the Console's Real Time Interface (RTI) module failed.

FAILED RTI LOOPBACK A TEST

FAILED RTI LOOPBACK B TEST

The console's Real Time Interface module failed the power-up tests.

Loop-back A tested the address, data, and control paths of the Programmable Peripheral Interface (PPI), of the RTI, and of the VAX Clock module (CLK). Loop-back A faults indicate problems with the RTI module.

Loop-back B tested the data paths on the VAX CLK module that allow the console to communicate with the CPU(s) through the receive and transmit data buffers (RXDB and TXDB). Loop-back B faults indicate problems with the VAX CLK and, possibly, the Decoder (DEC) module.

FAILURE READING REVISION RECORD FROM FILE

The console could not read REVHIST.DAT.

FEWER THAN 256 RECORDS IN CONFIGURATION TABLE

File MEMCONFIG.DAT, which contains memory configuration parameters, had fewer than 256 records. The error may indicate a file editing error or an illegal file format. Create a valid file with the INIT /CREATE command.

FILE NOT FOUND

The console application software could not open the specified file. The file was already opened by the console or LBS or the file does not exist.

FUNCTION ALREADY DISABLED

A specific function, switch, or mode was already disabled by default or by a previous DISABLE command.

FUNCTION ALREADY ENABLED

A specific function, switch, or mode was already enabled by default or by a previous ENABLE command.

HELP COMMAND NOT VALID INSIDE COMMAND FILE

The HELP command cannot be included in a command procedure.

HELP FOR THIS TOPIC IS NOT AVAILABLE YET.

The Help file for the specified topic could not be opened. Possible causes:

- The Help file does not exist.
- The other console is viewing the Help file.
- An illegal parameter was used with the HELP command.

ILLEGAL IPR# OR INVALID DATA

One of the following states occured:

- Nonexistent IPR
- Reading a write-only IPR
- Writing a read-only IPR
- Providing an unacceptable value (that is, mapen can only be 0 or 1)

ILLEGAL REGISTER FOR READ/WRITE IPR

The Console Support Microcode (CSM) rejected a console read or write request of an Internal Processor Register because it was reserved or improperly specified.

ILLEGAL TEST NUMBER(S)

Illegal test numbers were specified in a DIAGNOSE command.

INPUT FILE EMPTY

The file specified by the LOAD command was empty.

INPUT FILE INCOMPATIBLE WITH DESTINATION

The file header of the file being loaded did not match the command specifier indicating where the file was to be loaded.

Example: Attempting to load a Decoder RAM data file into the WCS RAMs.

INVALID COMMAND

An illegal command was issued in a logfile. Legal SHOW LOGFILE commands are: ↑, ↓, Prev Screen, Next Screen, 8, CTRL/C, CTRL/P, CTRL/W.

INVALID COMMAND SYNTAX

The input command string was not recognized by the command interpreter.

INVALID CSM FUNCTION

The Console Support Microcode (CSM) cannot execute indicated function. Indicates a console internal error.

INVALID DEVICE SPECIFICATION

The boot device specified was not understood by the BOOT command. The device specification must be in the form dddnnn. (See the description of the BOOT command in Chapter 5.)

INVALID EMM FUNCTION CODE IN COMMAND MESSAGE

The console EMM command contained an invalid EMM function code. The EMM returned a bad status to console. Indicates a console internal error.

INVALID FILENAME SPECIFICATION

The file name contains illegal characters.

INVALID PARAMETER IN QIO CALL

This message indicates an internal console application coding error.

INVALID TXDB ID RECEIVED FROM CSM

The console received TXDB data with an invalid ID field from the Console Support Microcode (CSM).

INVALID VALUE; DATABASE CORRUPTED

This message indicates an internal console error.

EMM TIMED OUT, CHECK EMM CONNECTION

The RTI driver timed out waiting for the EMM to respond to a command message. Enter message again, then check EMM.

IPR NOT ACCESSIBLE FROM CONSOLE

The EXAMINE command specified an invalid IPR.

LBS NOT INSTALLED

The console application software could not spawn the Logical Block Server program.

(LEFT) CPU IS NOT WORKING (VAX 8800)

This message indicates that the SET CPU LEFT command sensed the left CPU is unavailable due to a diagnostic, revision, a module keying problem or a LOAD command encountering a fatal error while loading a Control Store RAM.

LINK COMMAND NOT VALID INSIDE COMMAND FILE

A LINK command was issued from a command procedure.

LOGFILE OPENED BUT NOT READABLE

An error was encountered while reading the console log file LOGFILE.DAT.

MACHINE CHECK WHILE IN CONSOLE MODE

Operator tried to access nonexisting memory or I/O space while in console mode.

MACROCODE NOT STARTED

Macrocode was not running during attempt to transfer from VAX console I/O mode to program mode.

MEMORY ACCESS VIOLATION

An EXAMINE or DEPOSIT to virtual memory failed due to an access control violation.

MEMORY CONFIGURATION TABLE NOT FOUND

MEMCONFIG.DAT could not be opened by the INIT/MEMORY or UNJAM command. File either did not exist or was opened by another console port.

MICMON PROTOCOL ERROR

The Micromonitor could not communicate with a WCS-based test.

MICROCODE ERROR - EMPTY LOCATION EXECUTED

The Console Support Microcode (CSM) detected an attempt to execute a microword that does not contain a valid code.

MICROSTEP COUNT NOT IN RANGE 1 - 255

An invalid microstep count was specified in a command.

MODULE A NOT OK ON POWERUP

The console detected Module A did not assert its MODOK signal during its EMM polling routine (once a minute). Module A should assert its MODOK signal when the main power breaker is on.

NMI RESET RECEIVED—REBOOTING SYSTEM

NMI reset was received by the console, which then:

1. Waited for NMI RESET to be negated.
2. Halted the CPU(s) by asserting HALT PENDING.
3. Set the SYSINIT IN PROGRESS flag.
4. Executed the command procedure NMIRESET.COM to reboot the system.

NON-CSM RXDB DATA RECEIVED

The Console Support Microcode (CSM) could not understand the data passed to it through RXDB by the console.

NON-EXISTENT PHYSICAL MEMORY

The console requested the Console Support Microcode (CSM) to read /write physical memory, but memory did not respond.

NON-EXISTENT OR UNIMPLEMENTED FAST TEMP

The console requested the Console Support Microcode (CSM) to read /write an illegal Fast Data File register.

NO ACCESS TO PAGE IN VIRTUAL MEMORY, INSUFFICIENT PRIVILEGE

A DEPOSIT or EXAMINE VIRTUAL command was rejected by the Console Support Microcode due to a privilege violation.

NO ACCESS TO PAGE IN VIRTUAL MEMORY, LENGTH ERROR

A DEPOSIT or EXAMINE VIRTUAL command was rejected by the Console Support Microcode due to a page-length violation.

NO CPU(S) AFFECTED BY THIS COMMAND

The CPU was not selected and/or available. Therefore, the command could not manipulate the CPU.

A CPU is selected with the SET command. It is marked available with the VERIFY MODULE PLACEMENT command. It can be disabled by failure of either power-up tests or revision-compatibility checks.

NO CPU(S) AVAILABLE

The CPU was not available. Therefore, the command could not manipulate the CPU.

A CPU is selected with the SET command. It is marked available with the VERIFY MODULE PLACEMENT command. It can be disabled by failure of either power-up tests or revision-compatibility checks.

NO MEMORY ARRAYS IN SYSTEM

The Memory Controller (MCL) could not detect any memory array cards during the INITIALIZE/MEMORY command.

NO SECTION CURRENTLY LOADED

A section was not specified or previously loaded, preventing DIAGNOSE from executing.

NO USER WCS

A test was requested to exercise the User-Writeable Control Store, but it was not loaded.

NUMERIC VALUE TOO LARGE OR SMALL

An illegal numeric literal value was specified in a command.

OTHER REMOTE PORT FUNCTION IN PROGRESS

A remote port function was running when another function was issued. Wait until the current function is complete before entering another.

OVERRUN ERROR

The Console Support Microcode (CSM) rejected a console request due to a Console Support Microcode protocol violation.

PARITY ERROR DETECTED DURING LOAD

The Console application detected bad parity from parity checking logic in VAX CPU(s) while loading firmware. Parity error bits are checked after every microword is written.

PC NOT AVAILABLE

The console application software was unable to examine the VAX PC after the VAX CPU halted.

PL1REG.DAT NOT FOUND
PL2REG.DAT NOT FOUND

The console application software could not locate the file containing Environmental Monitoring Module parameters to be loaded into the EMM's PL1REG or PL2REG register.

POWER IN REQUESTED STATE

The power is already in the state requested by POWER ON, OFF, or STANDBY command.

POWER MUST BE ON

A command failed because power was off. Change the power state before issuing the command.

POWER NOT SET TO DESIRED STATE

After completing a POWER sequence, the console could not receive verification from the power system that the power is in the requested state. To verify power state, use the SHOW POWER command.

POWER RETURNED TO ORIGINAL STATE

An error occurred in the POWER ON command. The original power state is still in effect.

POWER STANDBY ONLY VALID WHEN POWER IS ON

A POWER STANDBY command was issued while power was off. Power can go to STANDBY from an ON state only.

R5 DEPOSIT FAILED

The BOOT command failed to deposit VAX General Purpose Register 5 with a parameter to be passed to VMB. The error was reported by the Console Support Microcode (CSM).

REGISTER NOT ACCESSIBLE FROM CONSOLE

An EXAMINE or DEPOSIT command to the VAX register failed, possibly because the Console Support Microcode (CSM) was not loaded.

REGULATOR MODOK SIGNAL NOT ASSERTED

The MODOK signal was not set in 100ms after the POWER ON command issued to the EMM to turn on regulator power. The console shut off the power and reported the offending regulator.

RESTART 1 INSTRUCTION EXECUTED ON EMM

The console's once-a-minute EMM polling routine detected an EMM problem. The console reloaded EMM parameters.

RESTART FLAG WAS ALREADY SET

The COLD_RESTART or WARM_RESTART flag was already set by the FIND/64K or FIND/RPB command respectively.

The COLD_RESTART and WARM_RESTART flags were cleared when the operating system booted. If either flag was set, the previous BOOT or RESTART would not complete. The console aborted the FIND command to prevent repetitive BOOT or RESTART attempts.

RIGHT CPU IS NOT AVAILABLE (VAX 8800)

The SET CPU RIGHT command sensed that a CPU is unavailable because of a diagnostic, revision, or module keying problems or because a LOAD command encountered a fatal error while loading a Control Store RAM.

RPB NOT FOUND (VAX 8800)

The Console Support Microcode (CSM) could not find a valid Restart Parameter Block during a FIND/RPB command.

RTI NOT INSTALLED

The console application software could not spawn the Real Time Interface task RTI.TSK into the operating system data base. An SCB read error is indicated.

SECONDARY NOT AVAILABLE (VAX 8800)

The secondary CPU was marked unavailable due to the failure of a previous command, such as VERIFY MODULE PLACEMENT or LOAD.

SECTION NAME IS MORE THAN 9 CHARACTERS

The test section name parameter in the DIAGNOSE command exceeded the allowable character length.

SHOW LOGFILES COMMAND NOT VALID INSIDE COMMAND FILE

A SHOW LOGFILE command was issued from a command procedure. SHOW LOGFILE must be issued separately.

SHUTTING DOWN REMOTE PORT

The EXIT command terminated a console application, disabling the remote port.

SPECIFIED ADDRESS IS NOT LOADED

The mnemonic address in the START command was not defined by loading a WCS file with the LOAD/WCS command.

START ADDRESS MUST BE LONGWORD ALIGNED

The LOAD/START command attempted to load memory into a location that is not longword aligned.

When loading memory with the LOAD/START command, the console application software only supports loading files to locations starting at longword boundaries.

SYNTAX ERROR

The format of the command is incorrect. See the manual for the correct command syntax.

TIMED OUT WAITING FOR CSM STATUS

The Console application Real Time Interface driver timed out waiting for the Console Support Microcode (CSM) to respond through the TXDB to a request made by the console.

UNABLE TO INIT CPU, START ABORTED

The initialization failed, preventing the START command from completing. The flag for system initialization remained set.

UNABLE TO LOAD CONSOLE-BASED SECTION

The Micromonitor DIAGNOSE command was unable to do one or more of the following:

- Spawn a P/OS-based task to execute a console-based diagnostic section.
- Communicate with the task through a P/OS-based intertask communication.

UNABLE TO LOAD DIAGNOSTIC CSM

The Micromonitor DIAGNOSE command could not load WCS with diagnostic Console Support Microcode from the file DIAGCSM.BIN.

UNABLE TO LOAD WCS-BASED SECTION

The Micromonitor DIAGNOSE command was unable to do one or more of the following:

- Load diagnostics into the VAX WCS.
- Spawn the console-based section.
- Communicate with the console-based section through SEND /RECEIVE intertask communication messages.

UNABLE TO OPEN LOGFILE

The console could not open the LOGFILE.DAT file during the power-up sequence, so it created a new LOGFILE.DAT file.

UNKNOWN CSM ERROR

The Console Support Microcode (CSM) rejected a console request, but the console had no translation for the reason code.

UNKNOWN EMM ERROR CODE

The console detected an unknown error code during its once-a-minute EMM polling routine. The problem may have been from a communication failure between the console and the EMM. The console reloaded the EMM parameters.

UNKNOWN MACHINE HALT EXECUTED

A macrocode halt occurred, but the halt code was not known to the console.

UNKNOWN SUBCOMMAND VALUE

The internal console application subroutine call passed an invalid parameter, which may indicate corruption of console code.

UNSUPPORTED CSM COMMAND

The Console Support Microcode (CSM) rejected a console request because the CSM does not support or recognize the request.

VALID PTE CANNOT BE FOUND

The Console Support Microcode (CSM) could not find a valid Page Table Entry to complete the console request.

VALID RPB NOT FOUND

The Console Support Microcode (CSM) could not find a valid restart parameter block while executing the FIND/RPB command.

VALUE TOO LARGE

The data specified in BOOT/R5=data command exceeded 4 bytes.

The data value exceeded the maximum size for specified data type on a DEPOSIT command.

WCS MUST BE LOADED BEFORE DECODER RAM

WCS microcode was not loaded, preventing the execution of the LOAD /DRAM command and an NOP microinstruction.

WRITE-ONLY IPR

An attempt was made to EXAMINE a write-only IPR. This is not allowable.

WRITING NON-EXISTENT I/O SPACE

The console requested the Console Support Microcode (CSM) to read /write an I/O page, but the specified address did not respond.

Appendix D

Bootstrap Control Flags

The bootstrap control flags provides VMB.EXE, the primary bootstrap program, with additional information about the type of bootstrap being performed. Table D–1 is a list of the flags that you can pass to VMB.EXE. These flags are passed by setting the appropriate bits in R5 of your boot command procedure.

Table D–1: Bootstrap Control Flags

Bit Position	Symbolic Name	Meaning
0	RPB$V_CONV	Conversational boot. At various points in the system boot procedure, the bootstrap code solicits parameters and other input from the console terminal
1	RPB$V_DEBUG	Debug. If this flag is set, VMS maps the code for the XDELTA debugger into the system page tables of the running system.
2	RPB$V_INIBPT	Initial breakpoint. If RPB$V_DEBUG is set, VMS executes a BPT instruction in module INIT immediately after enabling mapping.

Table D–1 (Cont.): Bootstrap Control Flags

Bit Position	Symbolic Name	Meaning
3	RPB$V_BBLOCK	Secondary boot from boot block. Secondary bootstrap is a single 512-byte block, whose LBN is specified in R4.
4	RPB$V_DIAG	Diagnostic boot. Secondary bootstrap is image called [SYSMAINT]
5	RPB$V_BOOBPT	Bootstrap breakpoint. Stops the primary and secondary bootstraps with breakpoint instructions before testing memory.
6	RPB$V_HEADER	Image header. Takes the transfer address of the secondary bootstrap image from that file's image header. If RPB$V_HEADER is not set, transfers control to the first byte of the secondary boot file.
7	RPB$V_NOTEST	Memory test inhibit. Sets a bit in the PFN bitmap for each page of memory present. Does not test the memory.
8	RPB$V_SOLICT	File name. VMB prompts for the name of a secondary bootstrap file.
9	RPB$V_HALT	Halt before transfer. Executes a HALT instruction before transferring control to the secondary bootstrap.
10	RPB$V_NOPFND	No PFN deletion (not currently used). Intended to tell VMB not to read a file from the boot device that identifies bad or reserved memory pages, so that VMB does not mark these pages valid in the PFN bitmap.

Table D–1 (Cont.): Bootstrap Control Flags

Bit Position	Symbolic Name	Meaning
11	RPB$V_MPM	Specifies that multiport memory is to be used for the total executive memory requirement; no local memory is to be used. This bit applies to the VAX–11 /782 only. If the bit RPB$V_ DIAG is set, the diagnostic supervisor enters AUTOTEST mode.
12	RPB$V_USEMPM	Specifies that multiport memory can be used in addition to local memory (as though both were one single pool of pages).
13	RPB$V_MEMTEST	Specifies that a more extensive algorithm is to be used when testing main memory for hardware uncorrectable (RDS) errors.
14	RPB$V_FINDMEM	Requests the use of MA780 memory if MS780 memory is insufficient for a bootstrap. This flag is used when performing software installations on a VAX–11/782.
15	RPB$V_AUTOTEST	Used by diagnostic supervisor.
16	RPB$V_CRDTEST	Specifies that memory pages with correctable (CRD) errors are NOT discarded at bootstrap time. By default, pages with CRD errors are removed from use during the bootstrap memory test.
<31:28>	RPB$V_TOPSYS	Specifies the top-level directory number for system disks with multiple systems.

Appendix E

Installing the Console Software

Your console is actually a computer with an operating system (P/OS) and application software. This software is separate from the VAX operating system and application software running on the VAX system. The console's operating system and required application (PRODCL, PRO /Communications) have been installed on your console's hard disk at the factory. Under most circumstances, you do not need to reinstall software on the console. However, if your console experiences software or disk problems, you must reinstall the operating system and/or applications. Detailed instructions for installing the P/OS operating system, PRO/Communications, and Console/PRO DCL application are provided in this appendix.

Before installing console software, make certain that your hardware components are correctly installed (see the *System Hardware User's Guide*) and check with your DIGITAL Field Service representative.

Read the following before installing console software:

- To choose a menu item, use the arrow keys to place the pointer and press DO.
- To insert diskettes, push the drive doors to open them. Insert the diskettes, lining up the arrows on the drive with those on the diskettes. Close the drive doors.
- Device names of diskette drives 1 (top) and 2 (bottom) are DZ1: and DZ2: respectively. The fixed disk device name is DW1:

E.1 Installing The Console Operating System (P/OS)

The P/OS diskette volumes listed in Table E–1 are in the release package accompanying this product. Store your P/OS diskettes in this box or in some other container made for storing diskettes. Use these diskettes as backups if the copy on the disk becomes unusable or if you need to reinstall P/OS.

Table E–1: P/OS Volumes

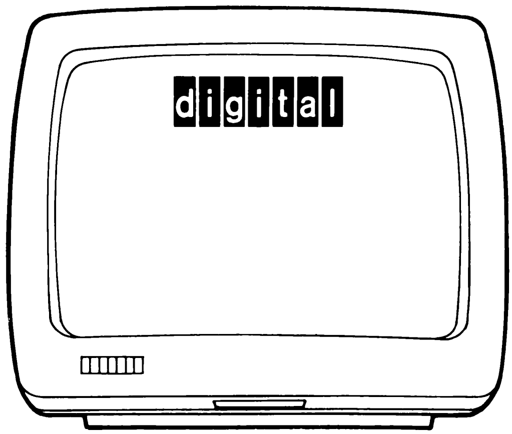
Order Number	Title	Label
BL-FH31A-ME	P/OS HARD DISK SYSTEM CONSOLE	PROSYSTEMV2
BL-FH32A-ME	P/OS HARD DISK SETUP CONSOLE	PROSETUPV2
BL-FH33A-ME	P/OS LIBRARIES CONSOLE	PROLIBRARYV2
BL-FH34A-ME	P/OS UTILITIES CONSOLE	PROUTILV2
BL-FH35A-ME	P/OS DISPATCHER CONSOLE	PRODISPATV2

When you install P/OS, the information on the diskettes is copied onto the disk in the console system unit. The console disk is erased (initialized) during this procedure. Stay near the console during the installation, since you may need to write down some information. The installation takes from 30 minutes to one hour.

To install the Console Operating System, follow this procedure:

1. Press the power switch on the system unit to 0 (off).
2. Remove the write-protect silver tape from the upper right side of the PROSYSTEMV2 diskette. The installation fails if this diskette is write protected.
3. Open the top (or left) diskette drive door and line up the orange arrow on the diskette with the arrow on the diskette drive. Insert the diskette VOL NAME: PROSYSTEMV2 into the drive and close the door. **Leave this diskette in the drive until the installation procedure is completed.**
4. Open the bottom (or right) diskette drive door and line up the arrows on the diskette drive. Insert the diskette VOL NAME: PROSETUPV2 into the drive and close the door. This diskette contains information about the keyboard and the monitor.
5. Press the power switch to "1" (on). The system runs a brief self-test, and the DIGITAL logo appears as in Figure E–1.

Figure E–1: DIGITAL Logo



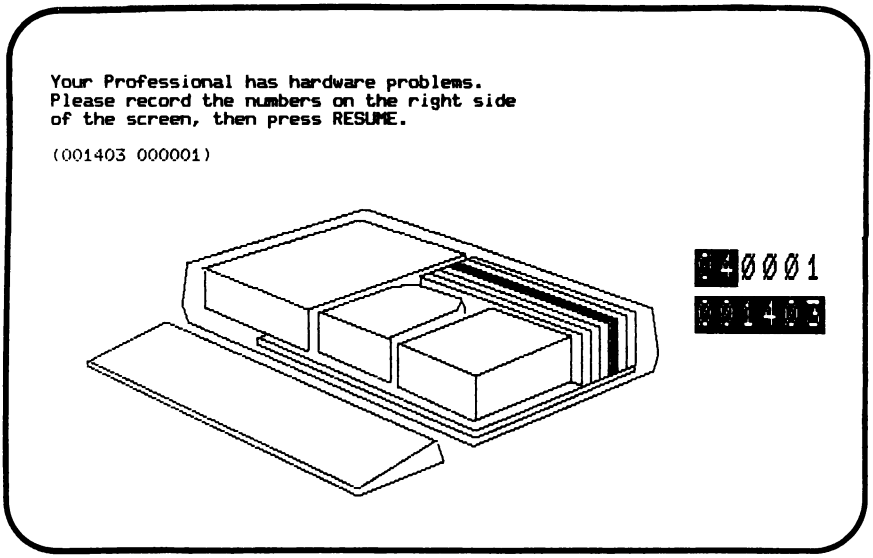
If the self-test detects problems, a diagram of the console components appears as in Figure E–2.

The console component with the problem is highlighted on the screen. Write down the numbers you see on the screen and call your DIGITAL Field Service representative for help. If the message indicates a problem that prevents the operating system installation, remove the diskettes and call your DIGITAL Field Service representative. If the message indicates that an error is interfering with the operating system after installation, either press the DO key at the top of the keyboard to continue installing P/OS (and fix the problem later) or remove your diskettes and contact your DIGITAL Field Service representative.

If this display appears after you have installed your operating system software, write down the numbers that appear on your screen. These numbers will help your DIGITAL Field Service representative determine the source of the problem.

If nothing happens when you turn on the power, call your DIGITAL Field Service representative.

Figure E–2: Console Screen Showing Hardware Problem



6. After the DIGITAL logo, the Installation Display appears on your screen as in Figure E–3.
7. Press the DO key at the top of the keyboard to continue the installation procedure. Press the EXIT key instead of DO to discontinue the installation.
8. If the operating system has been installed on your disk, the following message appears:

```
                 WARNING

Information has been found on your disk.

If you are sure you want to erase the information, press DO.

If you do not want to begin the installation, press EXIT and take out the
installation diskettes.
```

CAUTION

If you are reinstalling the console operating system, you must reinstall your applications and personal files, using

Figure E–3: Installation Display

your backup diskettes, because the installation procedure erases any information previously on the console disk.

At this point, press DO to continue the installation procedure or EXIT to terminate the installation.

9. Wait for the date and time form to appear:

   The blinking rectangle on the screen is the cursor, which marks the "active position" on the screen.

10. Fill in the form as instructed by typing the appropriate numbers from the auxiliary keypad (far right on your keyboard). Press RETURN to move from one entry to another. Be sure to enter 24-hour time. For example, if the time is 3:30 p.m., enter 15:30.

    If you make a mistake and have not yet press RETURN, press <X] to erase the mistake and then retype. If you want to change an entry after you pressed RETURN, press CANCEL and start again. If you type an impossible number (such as 13 for the month), the keyboard beeps, and you can enter a correct number.

11. When you have entered the date and time, press DO to continue (or EXIT to discontinue the installation).

12. Wait for P/OS to format and initialize the console disk. This procedure, which takes about 10 minutes, formats the console disk so the operating system can write information on it.

    Throughout the rest of the installation procedure, messages on your screen let you know what is happening and what to do next.

13. If a problem develops during the rest of the installation procedure, a series of numbers appears on the screen. Write down these numbers. They indicate a problem, and the operating system cannot be installed until the problem is corrected. After you write down the numbers, remove all diskettes from the drives and repeat the installation procedure, beginning at step 1. If you still get an error message, call your DIGITAL Field Service representative.

14. When the formatting and initialization completes, a message asks you to press DO to change your keyboard setting or RESUME to continue without changing your keyboard setting. Press RESUME.

15. Wait 5 to 10 minutes while P/OS copies the information from the PROSETUPV2 diskette to the disk.

16. When the diskette is copied, a message tells you to remove the PROSETUPV2 diskette from slot 2 and to insert the next P/OS diskette. (Make sure you leave the PROSYSTEMV2 diskette in slot 1.) When you have inserted the new diskette, close the diskette slot door and press RESUME to continue.

    If you do not insert the correct diskette, the same message tells you to insert the diskette, allowing you to try again.

17. Wait a few minutes while the information is copied from the next P/OS diskette to the disk.

18. When the diskette is copied, a message tells you to remove that diskette and insert the next diskette. When you have inserted the next diskette, press RESUME.

19. Screen messages prompt you to insert the following diskettes in turn:

    PRODISPATV2
    PROLIBRARV2
    PROUTILV2

    You are informed when the installation is complete.

20. Remove the diskettes from both drives and store them. **These are your backup diskettes. If anything should happen to the operating system on the disk, you will need to use these diskettes again.**

21. Turn off the power switch and then turn it on. The console runs the self-test. Then the console returns to the P/OS Main Menu.

22. The next step is to install PRO/Communications. See the next section.

E.2 Installing PRO/Communications

NOTE

You must be certain the PRO/Communications application is installed before installing the CONSOLE/PRODCL application (described in the next section).

The following diskettes are required to intall PRO/Communications.

Order number	Title	Volume Label
BL-FH36A-ME	PRO/COMM 1/3 CONSOLE	COMV20BL6HD1
BL-FH37A-ME	PRO/COMM 2/3 CONSOLE	COMV20BL6HD2
BL-FH38A-ME	PRO/COMM 3/3 CONSOLE	COMV20BL6HD3

Before installing PRO/Communications, you must remove the existing version, if any, from the disk using these steps:

1. At the Main Menu, select "Disk/diskette services."
2. At the Disk/Diskette Services Menu, select "Remove application".
3. At the Application Group Menu, select "Main Menu Applications".
4. At the Applications Within a Group Menu, select "PRO/Communications

5. After the application is removed, press EXIT to return to the Main Menu.

To install the PRO/Communications application, use the following procedure:

1. From the MAIN screen, select "Disk/diskette services".
2. From the Disk/diskette services screen, select "Install Application".
3. Insert diskette VOL NAME: COMV20BL6HD1 into the top (or left) diskette drive and press RESUME.
4. From the Application Install menu, select "PRO/Communications".
5. From the Application/Group menu, select "Main menu application".
6. From the Application Group Name Change menu, press DO.
7. When prompted to do so, remove the first diskette, insert Diskette VOL NAME: COMV20BL6HD2 in its place and press RESUME.
8. When prompted to do so, remove the second diskette, insert the diskette VOL NAME: COMV20BL6HD3 in its place and press RESUME.
9. When the application is installed, remove the final diskette and press EXIT. Press MAIN SCREEN to display the Main Menu. PRO /Communications appears on the Main Menu.
10. Select "PRO/Communications" from the Main Menu and press RESUME.
11. Cycle power to the console (turn it off and then on). PRO /Communications is now operational.

E.3 Installing the CONSOLE/PRODCL Application

NOTE

Before installing the CONSOLE/PRODCL application, the PRO /Communications application must be installed.

If you have installed the Console Operating System, you also need to install PRO/Communications and the Console/PRODCL application. If you have not installed the Console Operating System and need to install the Console/PRODCL application only, you must first remove the existing Console/PRODCL application, using steps 1 and 2.

This installation procedure assumes that the console already has POS Rev 2.0A and PRO COMMUNICATIONS installed.

The Console/PRODCL application consists of seven diskettes, labeled as follows:

Name	Volume name
COMMAND LANGUAGE 1/2 CONSOLE	CONSOLEDCL1
COMMAND LANGUAGE 2/2 CONSOLE	CONSOLEDCL2
RX99 REVISION HISTORY	NCON3
RX100 CONSOLE TASKS	NCON1
RX101 ISP MICROCODE	NCON2
RX102 BOOT COMMAND FILES	NCON4
RX107 MICRODIAG #1	NDIAG4

Follow these steps to install the Console/PRODCL application:

1. Power up the console. The first time the console system is powered up it displays the Main Menu. If you are reinstalling the Console /PRODCL application, powering up the console will run the Console application and enter console mode, (the > > > prompt). If this occurs, perform the following steps:
   a. Exit console application to the $ prompt by typing the console command, EXIT.
   b. Type EXIT again at the $ prompt to display the Main Menu.
2. If Console/PRODCL does not appear on the Main Menu, skip to step 3. If the Console/PRODCL application appears on the Main Menu, remove it with these steps:
   a. At the Main Menu, select "Disk/Diskette services".
   b. At the Disk/Diskette Services Menu, select "Remove application".
   c. At the Application Group Menu, select "Main Menu Applications".
   d. At the Applications Within a Group Menu, select "CONSOLE /PRODCL".
   e. When the application is removed, press EXIT to return to the Main Menu. Now you may install the application. The Console /PRODCL application diskettes can be inserted into either diskette drive.

3. From the Main Menu, select "Disk/diskette services".
4. From the Disk/diskette services menu, select "Install application".
5. You are now instructed to insert the application diskette CONSOLEDCL1. Insert the diskette in the top (or left) drive and press RESUME.
6. From the Application Installation Menu, select "Console/PRODCL".
7. From the Application Group Menu, select "Main Menu Application".
8. You are then asked if you want to change the application name. Press DO. (Do **not** change the application name).
9. You are then asked to insert CONSOLEDCL2. Remove CONSOLEDCL1, insert diskette labeled CONSOLEDCL2, and press RESUME.
10. You are then asked to insert NCON1. Remove CONSOLEDCL2, insert the diskette labeled NCON1, and press RESUME.
11. You are then asked to insert NCON2. Remove the diskette labeled NCON1, insert the diskette labeled NCON2, and press RESUME.
12. You are then asked to insert NCON3. Remove the diskette labeled NCON2, insert the diskette labeled NCON3, and press RESUME.
13. You are then asked to insert NCON4. Remove the diskette labeled NCON3, insert the diskette labeled NCON4, and press RESUME.
14. You are then asked to insert NDIAG4. Remove the diskette labeled NCON4, insert the diskette labeled NDIAG4, and press RESUME.
15. The installation is now complete. Remove the last diskette from the drive and save all the diskettes as backups.
16. After the Console/DCL application is installed, exit to the Main Menu.
17. At the Main Menu, select "Console/PRODCL". The >>> prompt appears.
18. Exit from console mode to the $ prompt by typing the EXIT console command.
19. Use the following steps to set up a pointer in [ZZSYS] to make the P/OS boot procedure automatically run the console when the console system unit is powered on:
    a. Enter the following:

       $ *SHOW LOGICAL APPL$DIR*

       P/OS will display:

       $ APPL$DIR = SYSDISK:[ZZAP000nn]

b. To make [ZZSYS]FIRSTAPPL.PTR point to the APPL$DIR, enter:

$ RUN EDT
EDT> [ZZSYS]FIRSTAPPL.PTR

P/OS will display:

$ ZZAP000nn

c. Edit this line to matche the value returned by P/OS in response to the SHOW LOGICAL APPL$DIR command. For instructions on using the editor see Chapter 4.

d. Exit EDT by typing CTRL/Z and then EXIT.

20. Before powering up, be sure ALL floppies have been removed from the drives.

21. Switch the console system unit to off (0) and then back on (1)

Appendix F

SYSINIT and BOOT Files

This appendix lists the command file sequences that are invoked during the VAX 8800 system initialization, restart, and boot. The sequences are very similar for the VAX 8700, 8550 and 8500, except for commands that involve dual CPUs. The SET VERIFY command can be used to display the commands in these files as they are invoked.

F.1 SYSINIT.COM - System Initialization Procedure

Following is a listing of the SYSINIT.COM file:

```
SET VERIFY
SET DEF HEX                          ! Set default to hexidecimal
!+
!  THIS IS SYSINIT.COM FOR A VAX 8800 SYSTEM
!-
SET TERMINAL OPAO                    ! Set up logging
SHOW TIME                            ! Put time stamp in logfile
VERIFY EMM                           ! Make sure we can talk to the EMM
IF NOT $STATUS THEN @EXIT            ! If no EMM, abort system init.
IF NOT POWER_STATUS THEN             ! If power is not on
IF NOT AUTO_POWERON THEN @EXIT       ! and AUTO_POWERON keyswitch not enabled
                                     ! exit system initialization procedure.
IF NOT POWER_STATUS THEN POWER ON    ! If power is not on and AUTO_POWERON
                                     ! keyswitch is enabled, turn on power.
IF NOT $STATUS THEN @EXIT            ! Abort if POWER ON failed
!
SET CLOCK OFF                        !
SET SYNC O                           ! Clear SOMM and TOMM
SET CLOCK NORMAL                     ! Set clock period
!
VERIFY MODULE PLACEMENT              ! See if any CPUs are present
IF NOT $STATUS THEN @EXIT            ! If no CPUs, quit.
```

```
SET CPU BOTH                            ! For whatever CPUs are there, ...
SENSE REVISION CPU                      ! Read CPU board revisions
SENSE REVISION CONSOLE                  ! Read console, EMM, drivers, etc.
SET CLOCK ON
!+
! Perform power up diagnostics
! After this test only the RIGHT CPU is selected, reselect both CPUs
!-
TEST/COMMAND                            ! Enter micromonitor mode
SET FLAG TRACE
                                        ! Display section and test names
DIAGNOSE/SECTION:EZKPA                  ! Execute console-based powerup tests
IF NOT $STATUS THEN @MICEXIT            ! If diagnostic fails, exit
DIAGNOSE/SECTION:EZKPB                  ! Execute more powerup tests
IF NOT $STATUS THEN @MICEXIT            ! If diagnostic fails, exit
EXIT                                    ! Exit micromonitor mode
SET CLOCK ON                            ! Make sure clocks are on
!+
! Power up diagnostics done
!-
SET CPU BOTH                            ! Reset CPU selection
LOAD/WCS UCODE.BIN                      ! Load ISP microcode,
IF NOT $STATUS THEN @EXIT               ! Exit if any error is encountered
LOAD/CACHECONTROL CCODE.BIN             ! Cache microcode,
IF NOT $STATUS THEN @EXIT               ! Exit if any error is encountered
LOAD/IBDECODER DRAM.BIN                 ! IBOX decoder tables.
IF NOT $STATUS THEN @EXIT               ! Exit if any error is encountered
SET CLOCK OFF                           ! Stop clocks to start microcode
START/CONTROLSTORE CSM_INIT             ! Start the CPUs!
IF NOT $STATUS THEN @EXIT               ! Exit if something went wrong
WAIT                                    ! Wait for CPUs to HALT
LOAD/SDF SDFDEF.BIN                     ! Load microcode constants
IF NOT $STATUS THEN @EXIT               !
!+
!  Remove the comment flags on the following lines if you are loading user WCS
!-
! LOAD/UWCS USERWCS.BIN                 ! Must come after SDF is loaded
! SET CLOCK OFF                         ! Clocks off to start microcode
! START/CONTROLSTORE CSM_INIT           ! Restart microcode with user WCS
! IF NOT $STATUS THEN @EXIT             !
! WAIT                                  ! Wait for CPUs to halt again
INITIALIZE                              ! Initialize hardware states & VAX regs
IF NOT $STATUS THEN @EXIT               !
UNJAM                                   ! Reset I/O devices
SENSE REVISION MCL                      ! Read memory controller revision
SENSE REVISION NBI                      ! Read NBI adapter revision
```

```
!+
!  All of the component revision levels have been sensed, now verify them
!-
VERIFY REVISION CPU                     ! Check CPU revision compatibility
VERIFY REVISION CONSOLE                 ! Check console revision compatibility
VERIFY REVISION MICROCODE               ! Check microcode revision compatibility
VERIFY REVISION MCL                     ! Check revision compatibility
VERIFY REVISION NBI                     ! Check revision compatibility
VERIFY REVISION KERNEL                  ! Check compatibility of all components
INITIALIZE                              ! Load SID and rev IPRs
IF NOT $STATUS THEN @EXIT
!+
!  Before memory can be initialized, a memory configuration file must have been
!  created. This command will create the file memconfig.dat if it doesn't exist.
!-
INITIALIZE/MEMORY                       ! Write 0's to all of memory if memory
                                        ! were not backed up

!+
!  Initialize halt code on each CPU
!-
SET CPU LEFT
DEPOSIT/SDF OCA 3                       ! Powerfailure halt code
SET CPU RIGHT
DEPOSIT/SDF OCA 3                       ! Powerfailure halt code
!+
!  Attempt to restart or boot the primary
!-
SET CPU CURRENT_PRIMARY                 ! Single-CPU operations from here on
IF AUTO_RESTART THEN @RESTAR.COM        ! Attempt to restart if AUTO_RESTART
                                        ! enabled
IF AUTO_BOOT    THEN @DEFBOO.COM        ! If AUTO_RESTART disabled, and
                                        ! AUTO_BOOT enabled attempt to boot
                                        ! Otherwise, EXIT.
CLEAR RESTART_FLAGS                     ! Clears WARM and COLD restart flags
```

F.2 RESTAR.COM - Restart Sequence

RESTAR.COM is executed following a power failure or an error halt when the AUTO_RESTART keyswitch is enabled.

```
SET VERIFY
!
! RESTAR.COM
!
! This command file is invoked after a HALT, if a reboot has not been
! requested by the VAX, and after power failure recovery if AUTO_RESTART
! is enabled.
!
! The following tests are required:
!
IF NOT AUTO_RESTART THEN  ! Execute defboo if
IF AUTO_BOOT THEN @DEFBOO ! auto_restart is not enabled but auto_boot is.
IF NOT AUTO_RESTART THEN @EXIT ! if neither auto_restart nor auto_boot are
                                ! enabled then remain halted
                          !
SET TERMINAL OPAO         ! Set up logging
SET DEFAULT HEXADECIMAL,PHYSICAL,LONGWORD
SET CPU NEXT_PRIMARY      ! Boot the processor specified by last SET NEXT_PRIMARY
IF NOT HALTED THEN HALT   ! Halt processor.
WAIT                      ! for processor to halt
EXAMINE/SDF OCA           ! Retrieve HALT code
DEPOSIT AP @              ! Put halt code into the AP
EXAMINE PC                ! Retrieve old PC
DEPOSIT R10 @             ! Put old PC into R10
EXAMINE PSL               ! Retrieve old PSL
DEPOSIT R11 @             ! Put old PSL into R11
INITIALIZE                ! Initialize processor.
EXAMINE AP
DEPOSIT/SDF  OCA @
DEPOSIT R0 0              ! Clear unused register.
DEPOSIT R1 0              ! Clear unused register.
DEPOSIT R2 0              ! Clear unused register.
DEPOSIT R3 0              ! Clear unused register.
DEPOSIT R4 0              ! Clear unused register.
DEPOSIT R5 0              ! Clear unused register.
FIND/RPB                  ! Locate restart parameter block and put addr + 200
                          !  in SP and contents of RPB+4 into PC
IF NOT $STATUS THEN       ! If an RPB was not found and
IF AUTO_BOOT THEN @DEFBOO ! auto_boot is enabled go try to reboot
IF NOT $STATUS THEN @EXIT ! If an RPB was not found and auto_boot is
                          ! Disabled then remain halted
EXAMINE PC                ! Get start address
START @                   ! go
```

F.3 DEFBOO.COM - Default Boot Command Procedure

The following command procedure is executed when the BOOT command is entered. It also executes following a power failure or error halt when the AUTO_BOOT keyswitch is enabled and a restart attempt fails. DEFBOO.COM consists of a single BOOT command that allows you to maintain a single copy of your boot command file.

```
!DEFBOO.COM
!
!  This command procedure will be executed automatically during system
!  initialization or following a power failure or an error halt if you have
!  enabled the AUTO_BOOT keyswitch.  It will also be executed when the command
!  BOOT is entered.
!
!  Replace DDDnnn with the name of the boot command procedure that you
!  wish to execute.
!       DDD = device name:  BCI, BDA, UDA
!       nnn = boot device unit number
!
SET VERIFY
BOOT DDDnnn             !  Boot from device type DDD, unit nnn
```

F.4 SECBOO.COM, Secondary Processor Boot Command Sequence - VAX 8800

On the VAX 8800, the main difference between the Primary and Secondary CPUs is that the Secondary never actually "boots." A "boot" in the Secondary CPU is implemented as a restart. The Secondary never needs to initialize its state from scratch. The Secondary uses a restart parameter block (RPB) that the Primary built and put into memory for the Secondary to find. The console reboots the Secondary when requested by the Primary (provided the command DISABLE SECONDARY has not been issued), using the SECBOO.COM command file. The following commands are in SECBOO.COM:

```
!  SECBOO.COM
!
!  Command procedure to boot VMS on the secondary CPU.  The
!  primary CPU must be booted first.  This procedure is invoked
!  by the primary as a result of a VMS DCL START/CPU command.
!
IF NOT SECONDARY_ENABLE THEN @EXIT      !Boot only if secondary enabled
SET TERMINAL OPA0                       !Set up logging
SET DEFAULT HEXADECIMAL,PHYSICAL,LONGWORD
SET CPU CURRENT_SECONDARY               !Select CPU
iF NOT HALTED THEN HALT                 !Halt CPU
WAIT                                    !Wait for halt
DEPOSIT MAPEN 0                         !Disable memory management
DEPOSIT PSL 41F0000                     !Set IPL=31, interrupt stack
FIND/RPB                                !Locate the restart parameter block
IF NOT $STATUS THEN @EXIT               !Boot if we find one
EXAMINE/SDF OCB                         !Show the restart routine address
DEPOSIT PC @                            !Set the PC with the start address
CONTINUE                                !VMS entry at RPB+100
```

F.5 EXIT Sequence

When a command exits with bad $STATUS, a status check chains into EXIT.COM which contains an <EOF> character to signal the console program that the command file is finished. The EXIT.COM command file exits from the command file to console mode.

```
!+
!  EXIT.COM
!-
```

F.6 BCIBOO.COM - Boot From CIBCI750

The BCIBOO.COM boot command procedure boots from unit 53, which is dual ported between HSC 1 and HSC 2. The path to the CI is BI 1, which contains the BCI750 at BI node 4.

```
! BCIBOO.COM
!
! Command procedure to boot VMS from an HSC over the BCI.
!
! NEXT_PRIMARY is expected to point to the CPU that is to be used
! as the primary CPU.
!
! The following register deposits must be done before executing this
! command procedure or must be edited to correspond to the hardware
! configuration:
!
! R1 - bus address information
! R2 - CI port # of HSC(s) to which drive is ported
! R3 - device unit number
!
SET VERIFY
SET TERMINAL OPAO               !Set up logging
SET DEFAULT HEXADECIMAL,PHYSICAL,LONGWORD
SET CPU BOTH                    !Select both CPUs
IF NOT HALTED THEN HALT         !Halt both CPUs
WAIT                            !Wait for both CPUs to halt
INITIALIZE                      !Init both CPUs, copy next_primary to current
SET CPU CURRENT_PRIMARY         !Select current_primary
UNJAM                           !Clear BI errors, NMI and BI node state info
EXAMINE/L/P 3E000010            !Read mem CSR4 (clears interrupts)
EXAMINE/L/P 3E000014            !Read mem CSR5 (clears NMI fault)
DEPOSIT/L/P 3E000000 020F00F0   !Clear interlock timeout in mem CSR0
DEPOSIT/L/P 3E000008 F0000000   !Clear RDS error bits in mem CSR0
INITIALIZE                      !Init primary
DEPOSIT R0 20                   !CI boot device type code
DEPOSIT R1 14                   !Boot device bus address:
                                !<3:0>=BI node #, <5:4>=BI #
DEPOSIT R2 21                   !CI port # of HSC:
                                !<7:0>=CI port # of HSC,
                                !<15:8>=CI port # of 2nd HSC if dual ported
DEPOSIT R3 %D53                 !Unit # of drive, decimal radix
SET DEFAULT HEXADECIMAL         !Reset radix
DEPOSIT R4 0                    !Not applicable
DEPOSIT R5 20000000             !Software boot control flags and
                                !root directory # in <31:28>.  Defaults
                                !to SYS0 or unrooted directory
FIND/MEM                        !Find 64kb of working memory; set cold start bit
IF NOT $STATUS THEN @EXIT       !Boot if find was successful
EXAMINE SP                      !Show address of working memory + %X200
LOAD/MAINMEMORY/START=@ VMB.EXE !Load VMB into good mem + %X200
START @                         !Start executing VMB
```

F.7 BDABOO.COM - Boot From KDB50

The BDABOO.COM boot procedure boots from unit 53, which is connected to a KDB50 at BI node 4 on BI 1.

```
!  BDABOO.COM
!
!  Command procedure to boot VMS from a BDA disk.
!
!  NEXT_PRIMARY is expected to point to the CPU that is to be used
!  as the primary CPU.
!
!  The following register deposits must be done before executing this
!  command procedure or must be edited to correspond to the hardware
!  configuration:
!
!  R1 - Bus address information
!  R2 - <31:24> = optional controller letter specifier, remaining bits
!               depend on device type in RO
!  R3 - device unit number
!
SET VERIFY
SET TERMINAL OPAO                !Set up logging
SET DEFAULT HEXADECIMAL,PHYSICAL,LONGWORD
SET CPU BOTH                     !Select both CPUs
IF NOT HALTED THEN HALT          !Halt both CPUs
WAIT                             !Wait for both CPUs to halt
INITIALIZE                       !Init CPUs, copy next_primary to current
SET CPU CURRENT_PRIMARY          !Select current_primary
UNJAM                            !Clear BI errors and NMI and BI node state info
EXAMINE/L/P 3E000010             !Read mem CSR4 (clears interrupts)
EXAMINE/L/P 3E000014             !Read mem CSR5 (clears NMI fault)
DEPOSIT/L/P 3E000000 020F00F0    !Clear interlock timeout in mem CSR0
DEPOSIT/L/P 3E000008 F0000000    !Clear RDS error bits in mem CSR0
INITIALIZE                       !Init primary
DEPOSIT R0 21                    !BDA boot device type code
DEPOSIT R1 14                    !Boot device bus address:
                                 !<3:0>=BI node #, <5:4>=BI #
DEPOSIT R2 0                     !<31:24>=optional controller letter specifier
DEPOSIT R3 %D53                  !Unit # of drive, decimal radix
SET DEFAULT HEXADECIMAL          !Reset radix
DEPOSIT R4 0                     !Not applicable
DEPOSIT R5 0                     !Software boot control flags and
                                 !Root directory # in <31:28>.  Defaults
                                 !to sys0 or unrooted directory
FIND/MEM                         !Find 64KB of working memory; set cold start bit
IF NOT $STATUS THEN @EXIT        !Boot if find was successful
EXAMINE SP                       !Show address of working memory + %X200
LOAD/MAINMEMORY/START=@ VMB.EXE  !Load VMB into good mem + %X200
START @                          !Start executing VMB
```

F.8 CSBOO.COM - Stand-alone Backup

CSBOO.COM is used to boot stand-alone backup from the console diskette DZ1. For information on booting stand-alone backup from an alternate root on your system disk, see Section 4.10.1.

```
!  CSBOO.COM
!
!       Standalone backup conversational boot command file.
!       Boot from CS1:
!
!
SET VERIFY
SET TERMINAL OPAO                !Set up logging
SET CPU BOTH                     !Select both CPUs
IF NOT HALTED THEN HALT          !Halt processor.
WAIT                             !Wait for both CPUs to halt
INIT                             !Init both CPUs, copy next_primary to current
SET CPU CURRENT_PRIMARY          !Select the primary to be booted next
UNJAM                            !Clear BI errors, NMI and BI node state info
SET DEFAULT HEXADECIMAL          !Set radix
EXAMINE/L/P 3E000010             !Read mem CSR4 (clears interrupts)
EXAMINE/L/P 3E000014             !Read mem CSR5 (clears NMI fault)
DEPOSIT/L/P 3E000000 020F00F0    !Clear interlock timeout in mem CSR0
DEPOSIT/L/P 3E000008 F0000000    !Clear RDS error bits in mem CSR0
DEPOSIT R0 40                    !Device type (console storage device).
DEPOSIT R1 0                     !Clear unused register.
DEPOSIT R2 0                     !Clear unused register.
DEPOSIT R3 1                     !Unit number.
DEPOSIT R4 0                     !Clear unused register.
DEPOSIT R5 4000                  !Sftwr boot flags.
FIND/MEM                         !Find 64kb of working memory, set cold start bit
IF NOT $STATUS THEN @EXIT        !Boot if successful
EXAMINE SP                       !Show address of working memory + 512
LOAD/MAINMEMORY/START=@ VMB.EXE  !Load primary bootstrap...
START @                          !  and start it.
```

Glossary

argument

An optional qualifier that can be entered with a command.

available

A state in which the CPU is present (it passed VERIFY MODULE PLACEMENT), enabled (it is the primary or ENABLE SECONDARY was entered), and working (it has not failed any diagnostics).

back-up process

The process of making copies of the data stored on your disk so that you can recover that data after an accidental loss. You make backup copies from the console's fixed disk to the diskettes.

back-up copy

A copy of data stored on your disk. The duplicate copy is stored on diskette.

baud rate

The speed at which signals are serially transmitted along a communications line. One baud equals one bit/second.

BI

Backplane Interconnect. It is the successor to the SBI, CBI, and UNIBUS for VAX–11 systems.

binary

A number system that uses two digits: 0 and 1. They are represented in VAX circuitry by two voltage levels. VAX programs are executed in binary form.

Glossary

bit

A binary digit; the smallest unit of information in a binary system of notation, designated as a 0 or a 1.

board

A printed circuit board. The board contains chips, electrical components, and electrically conductive pathways between components. A board stores data and memory or controls the functions of a device.

boot

See Bootstrap.

bootable medium

A disk containing software (such as an operating system) that a bootstrap program can load into the VAX system memory and begin program execution.

bootstrap

1. A program that you start when you turn on the VAX. The bootstrap loads software contained on a fixed disk into memory. After the software is loaded, the VAX stops executing the bootstrap program and starts executing the software in memory. The software usually loads an operating system or other software into memory so that the VAX can start processing. The bootstrap program (VMB) is loaded into memory by executing a boot command procedure. The command procedure is executed from SYSINIT (which runs when the console is powered up or after a power failure or error halt) if AUTO_BOOT is enabled. The boot command procedure can also be executed through the BOOT command.
2. To use a bootstrap program.

byte

A group of eight binary digits (bits). A byte is one-quarter of the size of a VAX word.

central processing unit (CPU)

The part of a VAX system that controls the interpretation and execution of instructions. The VAX 8800 system has two CPUs, a primary and secondary. The VAX 8700, 8550, and 8500 systems have one CPU only.

clock

The VAX CPU master clock, which synchronizes the CPU's activities. The VAX 8800 two clocks: Clock A (Left CPU) and Clock B (Right CPU).

clock burst

When the clock steps a specified number of cycles and stops.

command

An order you can give to the VAX, often through a terminal keyboard.

communications line

A cable along which electrical signals are transmitted. Devices or VAX systems that are connected by communications lines can share information and resources.

computer system

A combination of VAX hardware and software and external devices that perform computing operations.

console mode

The state in which the console terminal can be used to control the VAX CPU(s).

CPU

An abbreviation for central processing unit. See central processing unit.

cold start

To start the system from a powered-off state.

debug

To detect, locate, and correct errors (bugs) in system hardware or software.

diagnostic program (diagnostics)

A program that detects and identifies abnormal VAX hardware operation. Diagnostic programs are run in Micromonitor mode on the VAX system.

EMM

Environmental Monitoring Module. By means of its own microcomputer, the EMM enables the console to control and monitor the VAX power system.

error message

A message displayed by the VAX to indicate a mistake or malfunction.

file

A collection of related information treated by the VAX as a single item.

formatted data

A pattern of data that conforms to a structure dictated by the system software.

halted

A CPU state in which power to the CPU(s) is on and the clocks are running, but all CPU activity is on hold.

hard-copy terminal

A terminal that displays information on paper. See video terminal.

hardware

The physical components, mechanical and electrical, that make up a VAX system. See software.

IBDECODER

The Instruction Buffer Decoder holds VAX instructions for the CPU to execute, parses the instructions, and directs the CPU to execute the instruction.

interface

A device or software that allows the components of a VAX to communicate.

I/O

Abbreviation for input/output.

IPR

Internal Processor Register. A hardware register that is specific to one processor.

K

The symbol for 2 to the 10th power (or 1024 in decimal notation).

keyswitches

Software switches that can be enabled to start/restart the system and/or operating system. They are: AUTO_POWERON, AUTO_BOOT, AUTO_RESTART.

kilobyte

1024 bytes of information.

LED

Light-emitting diode. LEDs are indicators on the control panel. A segmented LED display on the CPU distribution panel insert on the back of the VAX cabinet displays the characters F–A and 8–0 during the power-on sequence to indicate CPU status and normal/abnormal operation.

load

1. To move software (usually from a peripheral device) to memory.
2. To physically place a disk on a disk drive or a tape on a tape drive.

M

The symbol for 1024 squared (1,048,576 in decimal notation).

MCL

Memory controller.

megabyte

1,048,576 bytes.

memory

The temporary storage area of the VAX. Information held here is erased when the system power is turned off.

memory module

A logic circuit board that contains additional memory for the system. Three memory modules with 1, 2, or 4 megabytes of memory can be added to the VAX.

menu

A displayed list of P/OS options that you can select to run.

Micromonitor mode

The mode in which the diagnostics software is operable for troubleshooting the VAX system.

micro-PC

A microcode program counter; the address of a microcode instruction.

NBI

An NMI to BI adapter interfaces the NMI bus to two Backplane Interconnect (BI) buses. It is the principal I/O path to BI-based disk storage, terminals, and other BI peripheral devices.

NBIA

An extended hex module mounted in the CPU backplane and contains the NMI port and NBI transaction ports.

NBIB

A BI Eurocard containing the BI user ports. It is located in the BI backplanes and connected to the NBIA through an interface cable.

NMI

Nautilus Memory Interconnect. The primary interconnect medium in the VAX system. The NMI can support up to two processors, a memory subsystem, and two I/O adaptors. It is a limited-length synchronous bus with centralized arbitration and control.

off line

The state at which equipment and devices are unable to communicate with the VAX or console.

on line

The state at which equipment and devices are able to communicate with the VAX system.

operating system

A collection of VAX programs that control VAX operation and perform tasks, such as assigning memory to programs and data, processing requests, scheduling jobs, and controlling the operation of input and output devices.

parameter

A term used with a command to specify the object of the command. Examples are a file, an address, or data.

P/OS

VAX console operating system.

power-up sequence (power up)

A series of ordered events that occur when you supply power by turning on the console system.

PPI

Programmable Peripheral Interface. It offers three 8-bit programmable ports used in the VAX system as data, address, and control ports linking the Console Interface on the VAX clock module to the rest of the console subsystem.

printer

An optional hard-copy device. With the VAX 8800, 8700, 8550, and 8500, it is used to provide paper copies of information stored in the VAX logfile.

program

The sequence of instructions the VAX needs to perform a task. See software.

program mode

A state in which the console communicates with the VAX as a terminal.

prompt

Words or characters the VAX displays to indicate that it is waiting for you to type a command.

PRO/DCL

The command language application used to control the console operating system (P/OS).

qualifier

A term that makes the action of a command more specific. Qualifiers are usually optional and are preceded by a slash (/).

RAM

Random-access memory. See Random-access memory.

random-access memory (RAM)

Memory that can be both read and written to during normal operations. The type of memory the VAX uses to store the instructions of programs currently being run.

read-only memory (ROM)

A memory whose contents cannot be modified. The VAX can use the data contained in a ROM but cannot change it.

reboot

To restart a VAX system. This causes a cold start in which the operating system must be reloaded.

restart

To start the VAX system via the RESTAR.COM command procedure and Restart Parameter Block (RPB) without reloading the operating system. See Warm start.

ROM

Read-only memory. See Read-only memory.

RTI

Real Time Interface. A console peripheral option containing an IEEE-488 bus, two serial line units, and a 24-bit wide programmable interface (PPI).

run

A single continuous execution of a program.

scope

A VAX 8800 CPU that is both selected and available. Console commands affect one or both CPUs. Commands that can affect both CPUs do so only if the scope is set for both CPUs. The scope of a command affected by the select and available criteria is controlled by the SET CPU command.

SDF

Slow Data File. A RAM used by the microcode to store constants and certain IPRs. See IPR.

selected

The CPU specified in a SET CPU command.

serial line unit

The connection between the console processor and the Environmental Monitoring Module (EMM) in the VAX power subsystem.

software

A program or programs executed by a VAX system to perform a chosen or required function. See hardware.

storage medium

A device that records information, for example a diskette.

store

To enter data into a storage device, such as a disk, or into memory.

sync pulse

A signal which is generated when the micro-PC matches the Micromatch register. Used by testing equipment.

system

A combination of VAX hardware, software, and peripheral devices that perform specific processing operations.

system image

The executable image of system software.

terminal

An input/output device that allows you to communicate with the VAX system. Terminals are divided into two categories: video and hard-copy.

trap silo

The stack of error traps used by microcode.

video terminal

A terminal that displays information on the screen of a cathode ray tube (CRT). See hard-copy terminal.

warm start

The system power is on, SYSINIT.COM is reexecuted following a power failure or error halt. RESTAR.COM will be executed to restart the operating system, provided the AUTO_RESTART soft keyswitch has been enabled. See Restart.

WCS

Writeable Control Store. It consists of three RAMs into which the system's microcode is written.

winchester disk

A hard disk permanently sealed in a drive unit to prevent contaminants from affecting the read/write head.

word

The number of bits (32) that the VAX handles in an operation in which the /WORD qualifier is used.

write protect

To protect a disk, diskette, or other storage medium from the addition, revision, or deletion of information.

write-protect notch

The small notch on the side of an RX50 diskette that you can cover with an adhesive-backed foil label or tab to prevent loss of data by accidental overwriting.

write-protect tab

A rectangular tab that is used to cover the write-protect notch of an RX50 diskette to prevent loss of data by accidental overwriting. See write-protect notch.

INDEX

A

B

C

D

E

F

G

H

I

T

U

V

W

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

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 ______

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

DIGITAL EQUIPMENT CORPORATION