EK-11785-UG-PRE

VAX-11/785 Hardware User's Guide

digital™

EK-11785-UG-PRE

VAX-11/785 Hardware User's Guide

Prepared by Educational Services
of
Digital Equipment Corporation

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

Printed in U.S.A.

The following are trademarks of Digital Equipment Corporation:

digital™
DATATRIEVE
DEC
DECmate
DECnet
DECset
DECsystem-10
DECSYSTEM-20
DECtape
DECUS
DECwriter
DIBOL
MASSBUS
PDP
P/OS
Professional
Rainbow
RSTS
RSX
UNIBUS
VAX
VMS
VT
Work Processor

CHAPTER 1

SYSTEM OVERVIEW

1.1 SCOPE

The VAX-11/785 Hardware User's Guide contains basic hardware operating and customer care procedures. It provides an overview of the system and peripheral hardware and serves as a general reference document for the system operator. This manual contains the following information.

Chapter 1 - An introduction to the system organization and documentation.

Chapter 2 - System operation from the operator's console.

Chapter 3 - The console subsystem and console command language.

Chapter 4 - The diagnostic supervisor.

Chapter 5 - Operating information for the most commonly used peripheral devices.

Table 1-1 is a list of VAX-11/785 system hardware manuals and related reference handbooks. Table 1-2 lists manuals that are available for the standard peripheral subsystems and options. All VAX/VMS customers receive the software documentation listed in Table 1-3. Manuals for optional software are available under separate license from the address below.

Table 1-1 VAX-11/785 System Hardware Manuals

Title	Document Number
VAX-11/785 Central Processor Unit Technical Description	EK-KA785-TD
VAX-11/785 TB, Cache and SBI Control Technical Description	EK-MM785-TD
VAX-11/785 Console Subsystem Technical Description	EK-KC785-TD

VAX-11/780 Power System Technical Description	EK-PS780-TD
MS780E Memory System Technical Description	EK-MS78E-TD
MS780 Memory System Technical Description	EK-MS780-TD
FP785 Floating-Point Accelerator Technical Description	EK-FP785-TD
DW780 UNIBUS Adapter Technical Description	EK-DW780-TD
RH780 MASSBUS Adapter Technical Description	EK-RH780-TD
CI780 Computer Interconnect Technical Description	EK-CI780-TD
VAX-11/785 Installation Manual	EK-SI785-IN
VAX-11/785 Hardware User's Guide	EK-11785-UG
VAX Diagnostic User Guide	EK-VX11D-UG
DS780 Diagnostic System User Guide	EK-DS780-UG
DS780 Diagnostic System Technical Description	EK-DS780-TD
VAX Maintenance Handbook, VAX Systems	EK-VAXV1-HB
VAX Maintenance Handbook, VAX-11/780	EK-VAXV2-HB
VAX Hardware Handbook, 1982-83	EB-21710
VAX Architecture Handbook, 1981	EB-19580
Computer Programming and Architecture - The VAX-11	EY-AX008-DP
System Maintenance Print Set	MP01747-**
Console Maintenance Print Set	MP01748-**
CPU Maintenance Print Set	MP01749-**
FPA Maintenance Print Set	MP01754-**
LA120 Technical Manual	EK-LA120-TM

LA120 User Guide	EK-LA120-UG
LA120 Operator Reference Card	EK-LA120-RG
LSI-11, PDP-11/03 User's Manual	EK-LSI11-TM
PDP-11 Bus Handbook	EB-17525-20
Microcomputers and Memories Handbook	EB-20912-20

**Note: The latest revision is sent if no revision is specified.

Table 1-2 VAX-11/785 Peripheral Manuals

Title	Document Number
Peripherals Handbook	EB-20443-20
RA60 Disk Drive User Guide	EK-ORA60-UG
RA60 Disk Drive Service Manual	EK-ORA60-SV
RA80 Disk Drive User Guide	EK-ORA80-UG
RA80 Disk Drive Service Manual	EK-ORA80-SV
RA81 Disk Drive User Guide	EK-ORA81-UG
RA81 Disk Drive Service Manual	EK-ORA81-SV
RM05 Disk Subsystem User Guide	EK-ORM05-UG
RM05 Disk Drive Maintenance Manual	EK-ORM05-EC
RP07 Technical Description Manual	EK-ORP07-TD
RP07 User Guide	ER-ORP07-UG
TU77 Magtape Technical Manual V1	EK-1TU77-TM
TU77 Magtape Technical Manual V2	EK-2TU77-TM
TU77 Magtape Transport User Guide	EK-OTU77-UG
TU78 Magtape Technical Manual V1	EK-1TU78-TM
TU78 Magtape Technical Manual V2	EK-2TU78-TM
TU78 Magtape Transport User Guide	EK-OTU78-UG

TU80 Tape Subsystem Technical Manual	EK-OTU80-TM
TU80 Subsystem User Guide	EK-OTU80-UG
DZ11 Asynchronous Multiplexer Technical Manual	EK-DZ110-TM
DZ11 User Guide	EK-DZ110-UG
DZ32 Technical Manual	EK-ODZ32-TM
DZ32 Asynchronous Multiplexer User Guide	EK-ODZ32-UG
DMF32 Multi Function Communications Interface Technical Description	EK-DMF32-TD
DMF32 User Guide	EK-DMF32-UG

Table 1-3 VAX/VMS Software Documentation

Volume	Title	Document Number
1	GENERAL INFORMATION	
	VAX-11 Information Directory and Index	AA-D016E-TE
	VAX/VMS Summary Description and Glossary	AA-D022C-TE
	VAX/VMS Primer	AA-D030C-TE
2A	COMMAND LANGUAGE AND SYSTEM MESSAGES	
	VAX/VMS Command Language User's Guide	AA-D023C-TE
2B	COMMAND LANGUAGE AND SYSTEM MESSAGES	
	VAX/VMS Guide to Using Command Procedures	AA-H782B-TE
	VAX/VMS System Messages and Recovery Procedures Manual	AA-D017C-TE
3	TEXT EDITORS AND FORMATTERS	

	VAX-11 SOS Text Editing Reference Manual	AA-M538A-TE
	EDT Editor Manual	AA-J726A-TC
	VAX-11 DIGITAL Standard Runoff User's Guide	AA-J268B-TK
4A	PROGRAM DEVELOPMENT TOOLS	
	VAX-11 Utilities Reference Manual	AA-H781B-TE
	VAX/VMS Magnetic Tape User's Guide	AA-M539A-TE
	VAX-11 SORT/MERGE User's Guide	AA-D113C-TE
4B	PROGRAM DEVELOPMENT TOOLS	
	VAX-11 MACRO User's Guide	AA-D033D-TE
	VAX-11 MACRO Language Reference Manual	AA-D032D-TE
	VAX-11 Linker Reference Manual	AA-D019C-TE
	VAX-11 Symbolic Debugger Reference Manual	AA-D026D-TE
5A	SYSTEM SERVICES AND I/O	
	VAX/VMS System Services Reference Manual	AA-D018C-TE
5B	SYSTEM SERVICES AND I/O	
	VAX/VMS I/O User's Guide (Vol. 1)	AA-M540A-TE
	VAX/VMS I/O User's Guide (Vol. 2)	AA-M541A-TE
6A	RUN-TIME LIBRARY	
	VAX-11 Run-Time Library User's Guide	AA-L824A-TE
	VAX-11 Run-Time Library Reference Manual	AA-D036C-TE
6B	RUN-TIME LIBRARY	

	VAX-11 Guide to Creating Modular Library Procedures	AA-H500C-TE
	VAX-11 Run-Time Library Language Support Reference Manual	AA-J107B-TE
7	VAX-11 RECORD MANAGEMENT SERVICES	
	Introduction to VAX-11 Record Management Services	AA-D024D-TE
	VAX-11 Record Management Services Tuning Guide	AA-M542A-TE
	VAX-11 Record Management Services Utilities Reference Manual	AA-M554A-TE
	VAX-11 Record Management Services Reference Manual	AA-D031D-TE
8	COMPATIBILITY MODE (RSX-11M)	
	VAX-11/RSX-11M User's Guide	AA-D037C-TE
	VAX-11/RSX-11M Programmer's Reference Manual	AA-D020C-TE
9	SYSTEM PROGRAMMING	
	VAX/VMS Real-Time User's Guide	AA-H784B-TE
	VAX/VMS Guide to Writing a Device Driver	AA-H499C-TE
	VAX/VMS System Dump Analyzer Reference Manual	AA-J526B-TE
	VAX-11 PATCH Utility Reference Manual	AA-H785B-TE
10	SYSTEM MANAGEMENT AND OPERATION	
	VAX-11/780 Software Installation Guide	AA-M545A-TE
	VAX/VMS System Management and Operations Guide	AA-M547A-TE

	VAX/VMS UETP User's Guide	AA-D643C-TE
11	NETWORKING	
	DECnet-VAX System Manager's Guide	AA-H803B-TE
	DECnet-VAX User's Guide	AA-H802B-TE
	DECnet-VAX Cross-System Notes	AA-M544A-TE

Hard copy documents may be ordered through the nearest DIGITAL sales office or the Accessories and Supplies Group catalog, or from one of the following sources.

Hardware documents:

Digital Equipment Corporation
Publishing and Circulation Services, NRO3-1/W3
10 Forbes Road
Northboro, MA 01532

Handbooks and software manuals:

Digital Equipment Corporation
Educational Services Distribution Center
12A Esquire Road
Brookside Industrial Park
North Billerica, MA 01862

Technical and service documents that support the VAX-11/785 (including maintenance print sets and diagnostic listings) are also available on microfiche. For information on microfiche libraries contact:

Digital Equipment Corporation
Micropublishing Systems, FPO/B5
30 North Avenue
Burlington, MA 01803

1.2 HARDWARE SYSTEM

The VAX-11/785 system contains a central processor unit (CPU) with an optional floating-point accelerator, an I/O subsystem, a main memory subsystem, a console subsystem, and peripheral devices. Figure 1-1 is a block diagram of a VAX-11/785 system.

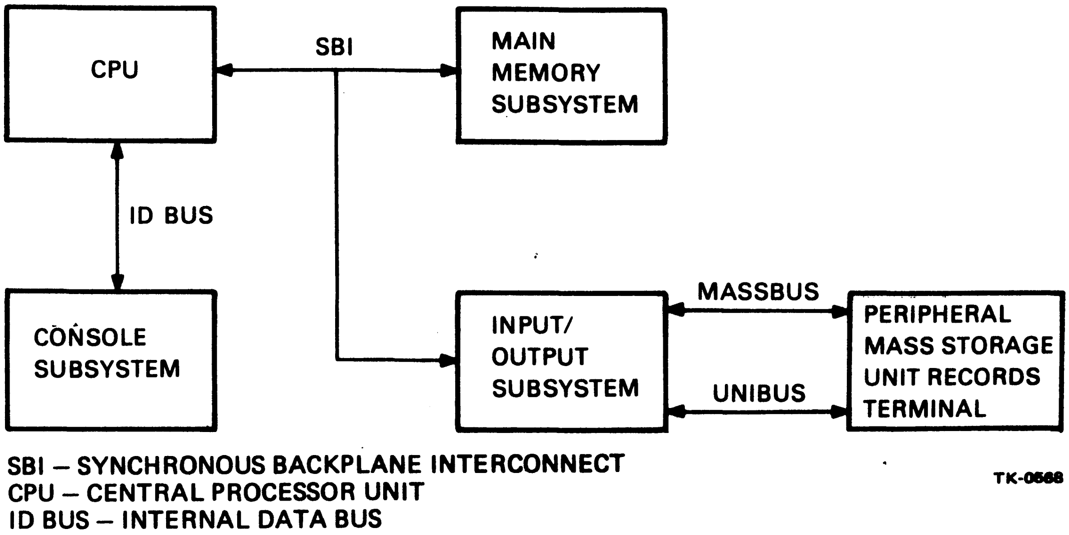


Figure 1-1 VAX-11/785 System Block Diagram

1.2.1 Central Processor Unit (CPU)

The VAX-11/785 CPU is a high-speed microprogrammed 32-bit computer which runs on a 133 nanosecond clock cycle. Figure 1-2 is a functional block diagram of the CPU.

In normal operation, data enters and leaves the CPU over the synchronous backplane interconnect (SBI). Since data is fetched from main memory eight bytes at a time, some may be stored temporarily in the data cache before going to a buffer to be decoded. The required data from a physical address is retrieved from memory if it is not already in the data cache.

The instruction buffer decodes instructions, and the translation buffer stores addresses that have been translated from virtual to physical. The translation buffer is a two-way buffer that stores virtual to physical address translations for the process currently executing in the system. The CPU microcode performs the virtual to physical translation. For a description of virtual memory concepts, refer to Appendix A.

The decoded instruction controls the joint control store (JCS) function. The JCS contains the two microprogrammed instruction sets (VAX-11/785 native mode and PDP-11 compatibility mode). The selected instruction set controls the operations in the data paths function.

The data paths function performs the following: packing and unpacking of floating-point instructions, address translation, and arithmetic and logical functions. These sections are independent and can operate simultaneously.

When an instruction is completed, the data that has been operated on is transferred back to memory.

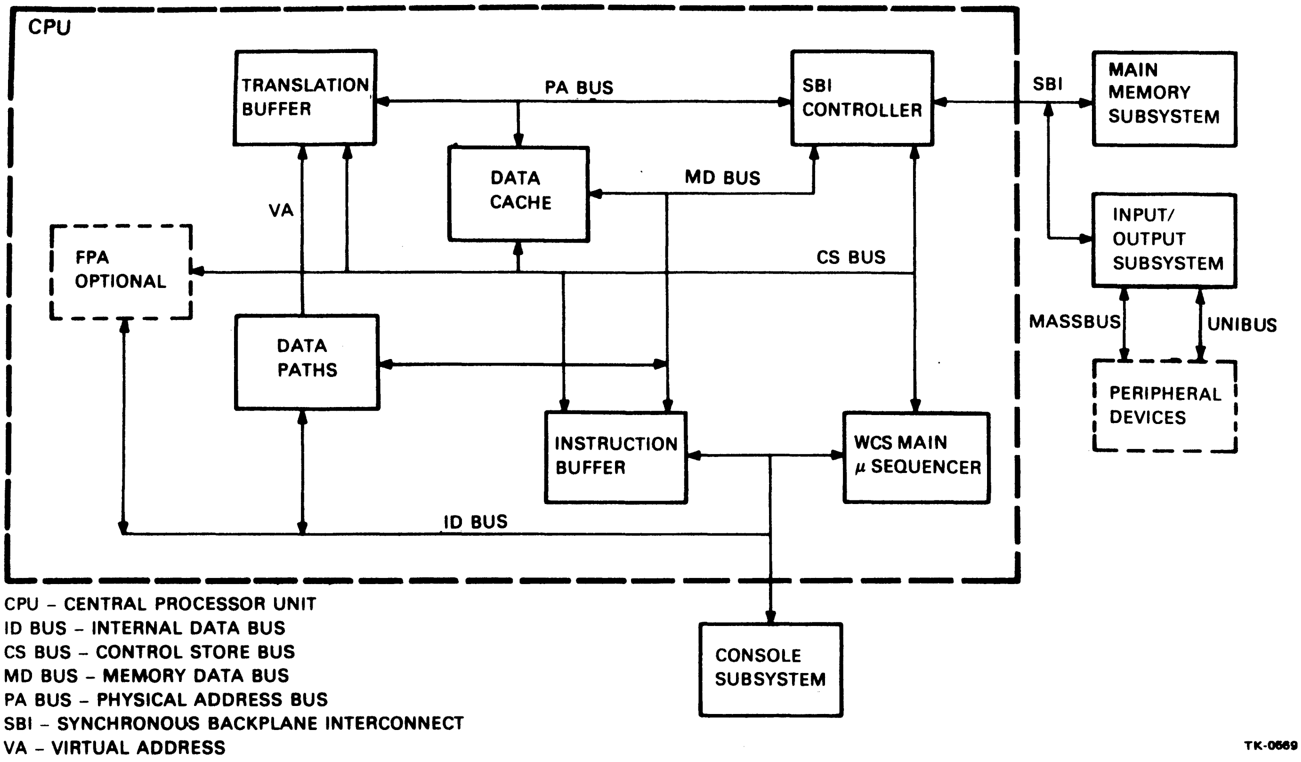


Figure 1-2 CPU Block Diagram

1.2.2 Floating-Point Accelerator Option

The floating-point accelerator (FP785) is an optional high-speed processor extension that uses the 133 nanosecond CPU clock. When included in the processor, the floating-point accelerator executes the addition, subtraction, multiplication, and division instructions that operate on single- and double-precision floating-point operands. It also executes the special EMOD and POLY instructions in both single- and double-precision formats. In addition, the floating-point accelerator enhances the performance of 32-bit integer multiply instructions.

The VAX-11/785 need not include the floating-point accelerator to execute floating-point instructions. Its purpose is to speed execution of these instructions. The floating point accelerator can be added or removed without changing any existing software. The only changes required to the hardware are switch settings on the CPU clock board.

1.2.3 I/O Subsystem

The I/O subsystem contains the SBI, the MASSBUS, and the UNIBUS. Figure 1-3 is a functional block diagram of the I/O subsystem.

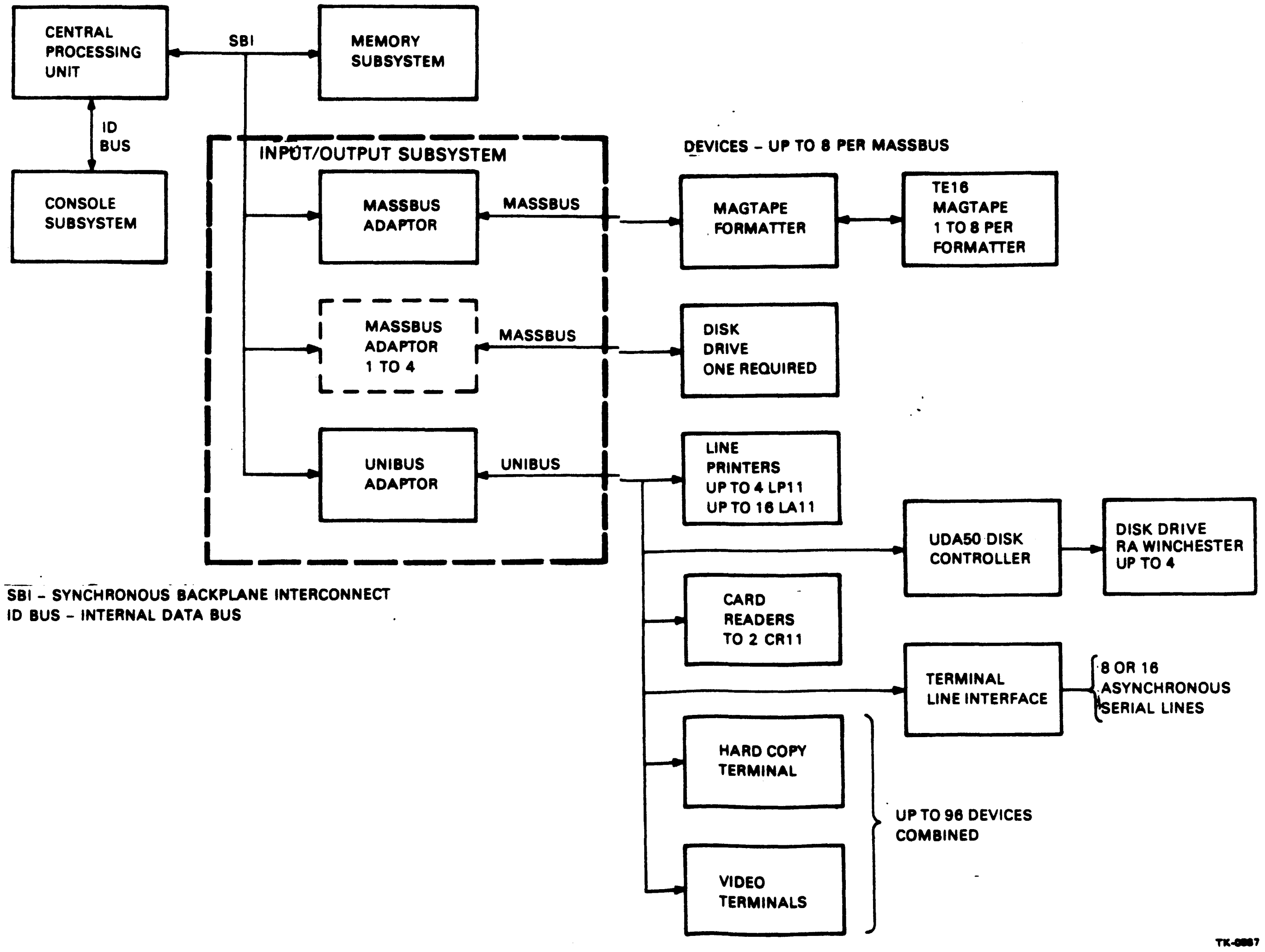


Figure 1-3 I/O Subsystem Block Diagram

1.2.3.1 SBI -

The SBI is the system's internal backplane and bus that conveys addresses, data, and control information between the CPU, main memory, and peripheral device controllers. The SBI has a cycle time of 200 ns and can transfer 32 bits of data, along with numerous control, arbitration, and data integrity signals, each cycle.

1.2.3.2 MASSBUS -

The MASSBUS adapter (MBA) is the hardware interface between the SBI and the high speed MASSBUS storage devices. The MBA performs control, arbitration, and buffering functions. Up to four MBAs can be placed on the SBI, with up to eight high speed disks or 64 tapes on each MBA.

1.2.3.3 Unibus -

The Unibus is connected to the SBI by the Unibus adapter (UBA). All devices other than the high-speed disk drives and magnetic tape transports connect to the Unibus, an asynchronous bidirectional bus. The UBA does priority arbitration among devices on the Unibus. See Figure 1-3 for possible configurations.

1.2.4 Main Memory Subsystem

The main memory subsystem (MS780E) contains an SBI interface and two memory controllers. Between one and eight MOS RAM arrays are attached to each controller. Each array can store up to one megabyte of information. Figure 1-4 is a functional block diagram of the main memory subsystem.

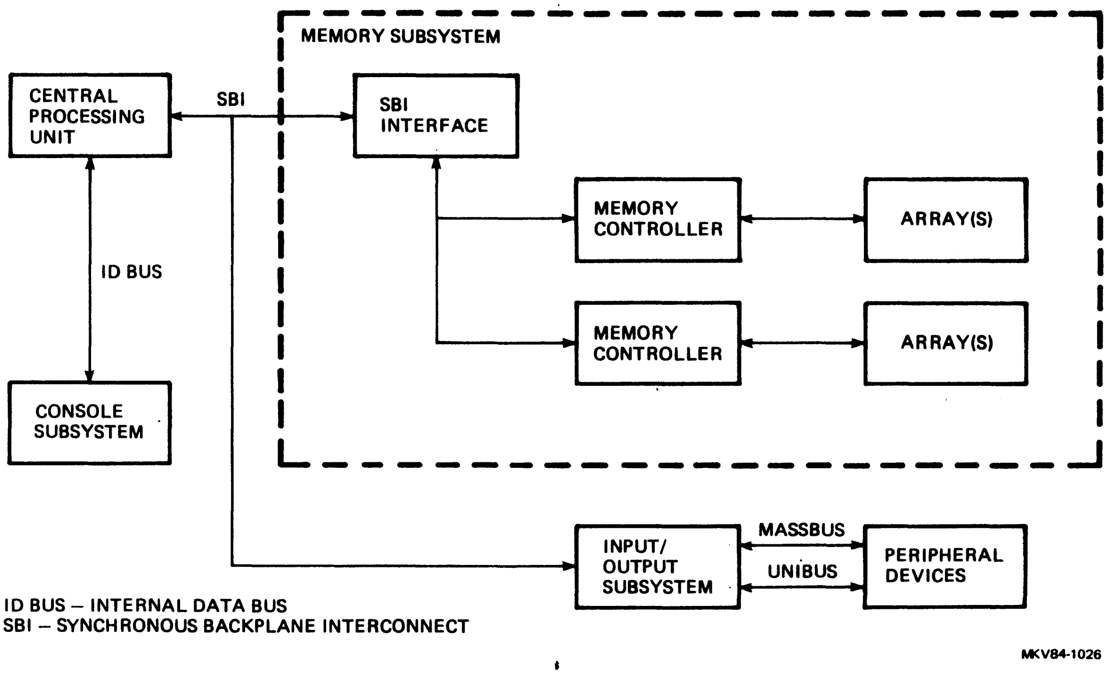


Figure 1-4 Main Memory Subsystem Block Diagram

1.2.5 Console Subsystem

The console subsystem consists of a console interface, a control panel on the VAX-11/785, an LSI-11 microprocessor, an RX01 diskette drive and controller, and a serial line interface connected to an LA120 console terminal. A remote terminal and an interconnecting line unit are optional. Figure 1-5 is a functional block diagram of the console subsystem.

Using the console terminal, the operator can examine and deposit data in memory locations or the processor registers, halt the processor, step through instruction streams, and boot the operating system.

The console is the means by which the system can be diagnosed. The diskette serves as the basic load device for microdiagnostics and microcode.

An EIA serial line interface and modem can be added to the console to provide remote diagnosis and automated testing.

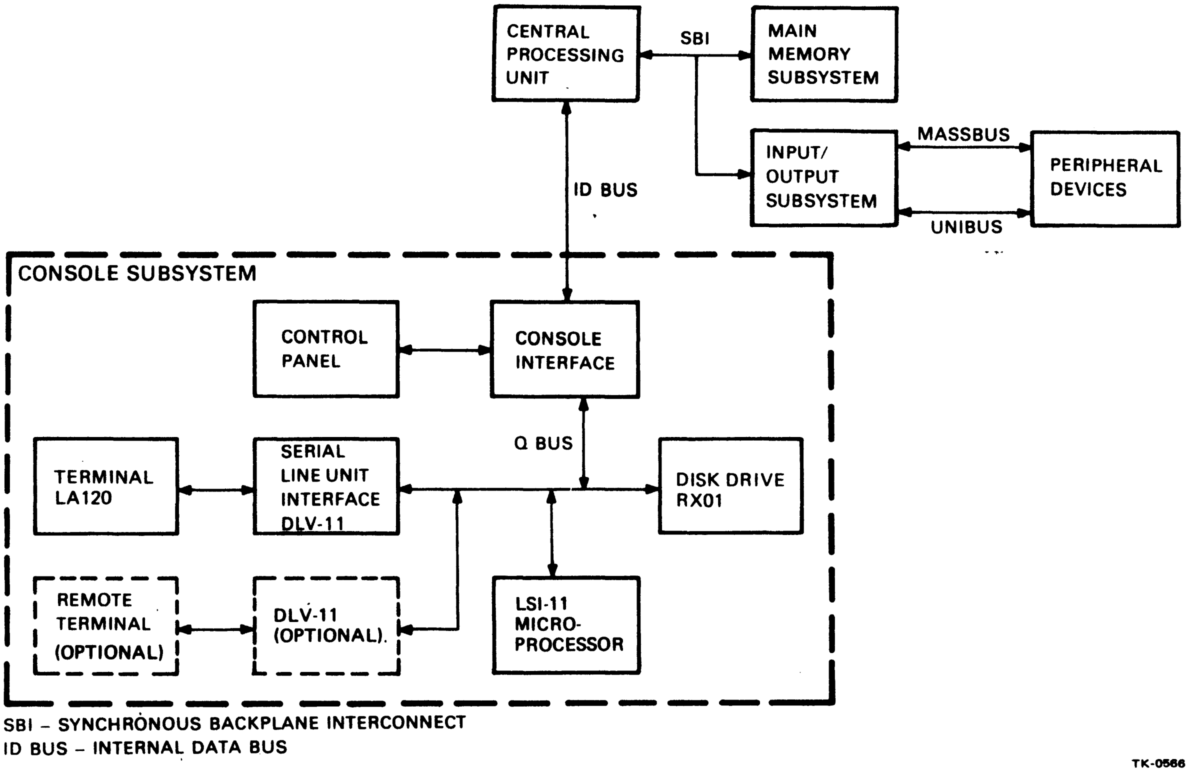


Figure 1-5 Console Subsystem Block Diagram

1.3 SOFTWARE SYSTEM

The VAX-11/785 software system includes the VAX/VMS operating system, on-line diagnostics, and the supported languages.

1.3.1 VAX Virtual Memory Operating System (VAX/VMS)

1.3.1.1 Virtual Memory -

NOTE

> A description of the virtual memory concept is contained in Appendix B.

Virtual memory extends the system memory onto disk space. From the programmer's viewpoint, the secondary storage locations are treated as memory. The size of virtual memory depends on the amount of physical memory available plus the amount of disk storage used for nonresident data and code. This design provides some obvious advantages.

With virtual memory the user can write and execute large programs without worrying about addressing limitations. VAX/VMS provides more virtual memory addresses than physical addresses. The system hardware and software cooperate to translate the virtual memory requirements of the user's program into whatever physical memory is available on the system. The VAX-11/785 physical memory configuration can change without requiring program redesign or relinking. Programs need not be structured, although they may be to achieve maximum efficiency and performance.

Virtual memory allows large programs (in excess of physical memory) to be executed. The program's virtual addresses are translated into the available physical memory.

Methods by which the VAX/VMS operating system benefits from a virtual memory system are developed in the following paragraphs.

1.3.1.2 User Environment -

System Management -- VAX/VMS provides tools for a system manager and a system operator to control the configuration and operation of the system to realize its most effective use. The system manager authorizes system users, determines quotas to control virtual memory usage, and defines privileges and priority to each user. At run time, the system operator can initiate and terminate sessions, monitor system operations, load and unload volumes, and perform backup and other on-line maintenance activities.

User Access -- To use the system, each user logs in by typing an identifying user name, account number (optional), and a password. If the user is authorized, access is granted and the resources of the system are available for use.

Users communicate with the system with DCL, Digital Command Language. It enables users to log onto the system, manipulate files, develop and test programs, and obtain system information. It also enables users to extend or redefine their set of commands and to combine commands to create command procedures.

VAX/VMS supports two instruction sets: native mode and compatibility mode. Native mode contains over 200 opcodes which constitute the full range of VAX-11 instructions. Compatibility mode enables the execution of most PDP-11 instructions. The exceptions are memory management instructions and floating-point instructions. Most user mode programs written for PDP-11, except stand-alone software, can be run under VMS compatibility mode.

Processes -- When a user logs into VAX/VMS, a process (job) is created. In VAX/VMS a process has two parts: the image that executes, and the process context. The process context is the virtual address space and control information required for image execution.

For example, system and user programs exist under VAX/VMS as image files. An image file is the result of linking one or more object modules together. The image file is loaded by the image activator (a system service) and executes in the context of the process.

The process executes all of the images needed to perform the user's requests. If the user chooses to edit, assemble, and link a source program, for example, the process executes three images: the editor, the assembler, and the linker.

A VAX/VMS process can execute any image. Furthermore, a VAX/VMS process remains to execute subsequent images when the current or initial image is exited.

1.3.1.3 Virtual Address Space -

Virtual address space is defined as all possible virtual addresses that an image executing in the context of a process can use. The available virtual address space of this system is an array of bytes labeled 0 through 2(32)-1 or 4.3 gigabytes.

The virtual address space is divided into four regions, each approximately 1 gigabyte long. The top 1 gigabyte region is reserved for future system expansion. Addresses in the system region are used by the VAX/VMS software.

The addresses available for user code are in the process virtual address space. This comprises the lower 2 gigabytes, the control region and the program region. This address space contains context information and user stacks, one per access mode per process, maintained by the system on behalf of the process. It also contains the image currently executing.

Figure 1-6 illustrates the VAX-11/785's 4 gigabytes of virtual address space and how it is allocated between the VAX/VMS and each process.

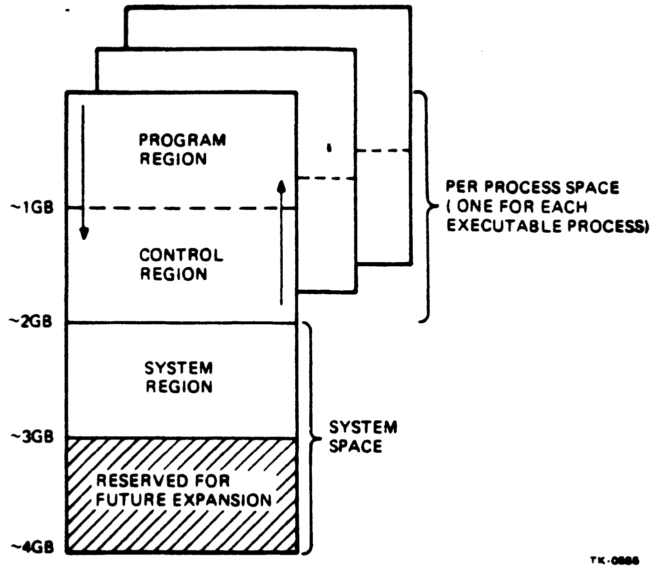

PROGRAM REGION
~1GB
CONTROL REGION
~2GB
SYSTEM REGION
~3GB
RESERVED FOR FUTURE EXPANSION
~4GB
PER PROCESS SPACE (ONE FOR EACH EXECUTABLE PROCESS)
SYSTEM SPACE
TK-0886

Figure 1-6 Virtual Address Space

1.3.1.4 Memory Management -

Memory management involves both the VAX-11/785 hardware and the VAX/VMS software with control of the pages of a process (through image activation and paging) and the protection of a process.

Pages -- In the VAX-11/785 system, a process's virtual memory is subdivided into pages of 512 contiguous bytes. This number makes efficient use of memory and it is the size of a physical block on a disk.

Image Activator -- The image activator is responsible for making an image capable of running in the context of the requesting process. The image activator locates the image and sets up the proper control information.

Paging -- Paging is the transparent moving of pages to and from secondary storage as they are needed. As a process executes, only those pages required for execution, the process's working set, need reside in physical memory. Remaining pages of the process can reside on disk and be brought into physical memory when needed.

A maximum working set size for each process is set by the system manager so that individual processes will not use enough physical memory to degrade the execution of other processes. If a process is time-critical, the system manager may assign a large maximum working set size. Parts or all of a working set of a given process can be locked into memory to gain maximum performance.

The paging facility maintains page tables on the status and locations of all virtual pages of processes in the system. The system page table (always in physical memory) maps individual process page tables (which may or may not be in physical memory).

If a page is not in physical memory, the process page table entry indicates where the page is on disk. The translation buffer speeds page table lookup procedures and CPU performance by storing the most frequently referenced page table entries in a high-speed memory.

The VMS operating system does process-oriented paging. It discards the oldest pages from a process when it is necessary to make room for new pages required by that process. The process pages only against itself. Individual processes doing a great deal of paging do not significantly degrade the performance of other active processes in the system.

Software caches in physical memory, called the free page list (for unmodified pages) and the modified page list (for modified pages), are reserved for the temporary storage of pages removed from processes' working sets. As unmodified newly removed pages are added to the bottom of the free page list, older pages work their way to the top and the space is eventually reused. As modified newly removed pages are added to the bottom of the modified page list, older pages work their way to the top and are eventually written to a paging file on disk. If a process needs a page that happens to be on one of these lists, the page is simply reallocated to the process, rather than reading the page back into memory from secondary storage.

The system handles all of the virtual to physical translations for the user. The user can, if desired, control the paging structure for the process. A page cluster size can be set at link-time, thereby causing multiple pages to be mapped from disk to memory to minimize disk accesses. Pages can be locked in the user's working set. Privileged users can lock pages or the entire working set in memory to gain maximum program performance.

Protection -- Four hierarchical access modes provide protection for the system (Figure 1-7). These modes, from most to least privileged, are: kernel, executive, supervisor, and user.

Each page table entry contains a code specifying, for each of the four modes, the type of access (read only, read write, and no access) allowed to the associated memory page. If a less privileged mode can access a page, then a more privileged mode can also access the page.

In VAX/VMS, the system services (record management services, I/O drivers, etc.) are part of every process's virtual address space. However, individual pages are protected in different access modes. For example, the I/O drivers, scheduler, and record management service execute in kernel mode; the record manager executes in executive and kernel modes; the command language interpreters execute in supervisor mode; user software and utility programs usually execute in user mode (although the system manager can assign privileges to users allowing them to protect pages at higher levels).

The access modes only determine access to pages (they do not affect the virtual space in which programs are mapped). By assigning appropriate access modes to pages of a process, the operating system is protected from users who might inadvertently try to destroy operating system information.

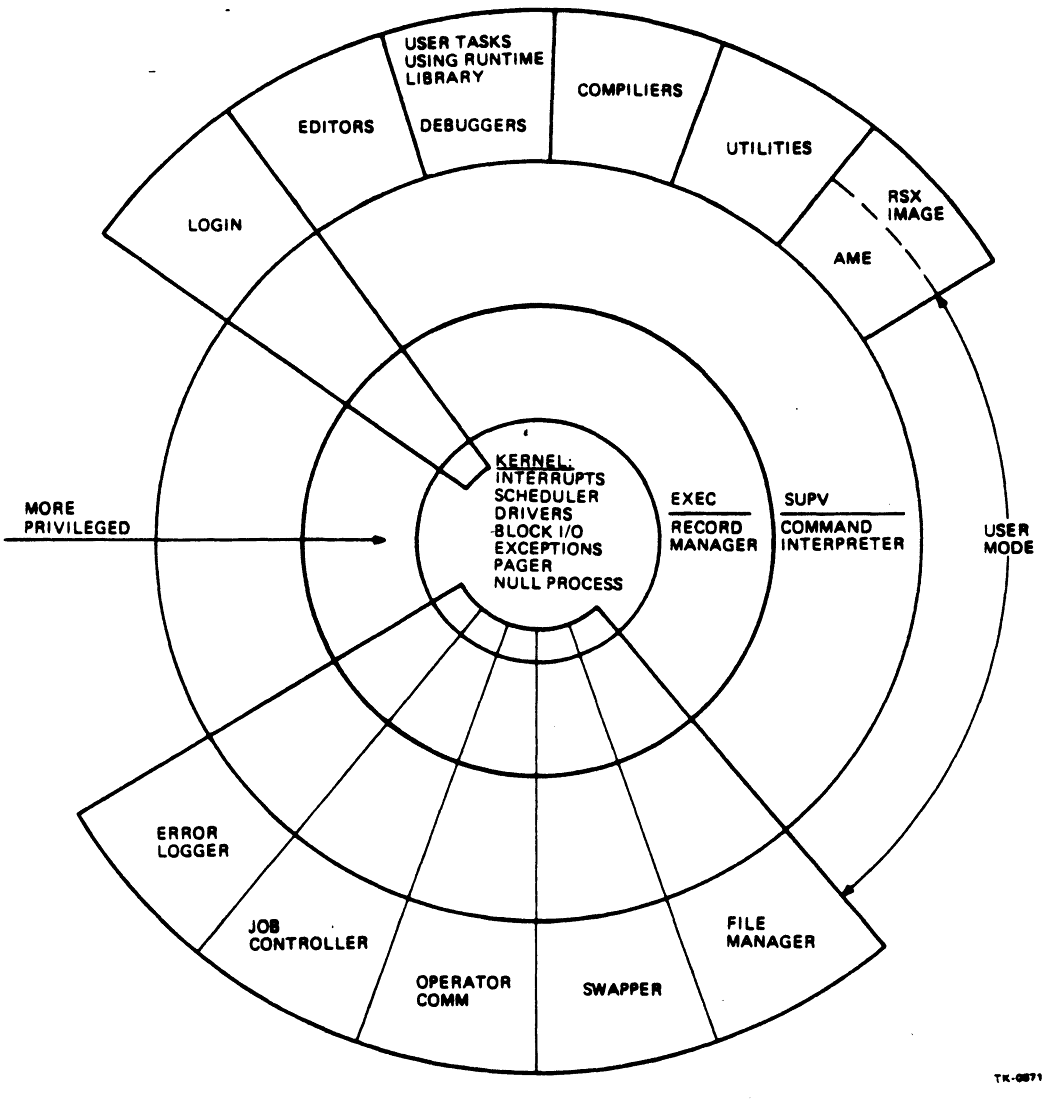


Figure 1-7 VMS Hierarchical Access Modes

1.3.1.5 Interprocess Communication -

VAX/VMS provides a number of methods by which processes can communicate with one another.

- Implicit communication using shared files or shared memory (the run-time library is an example of system code that is shared).

NOTE

> Memory data can be shared between native and compatibility mode programs, but code cannot because the instruction sets are different.

- Cooperating processes can synchronize using common event flags.
- Message communication is allowed through mailboxes. A mailbox is a buffer in virtual memory that is treated as if it were a record-oriented I/O device. Mailboxes are accessed through the operating system's I/O system, and processes can communicate by reading and writing messages through a mailbox. This method provides more protection, but requires system overhead.

1.3.1.6 Scheduling And Swapping -

The VAX/VMS scheduler and swapper are responsible for ensuring that processes receive processor time that is appropriate for their priority and their ability to execute.

The scheduler determines the order in which processes are executed. It uses fixed priorities for time-critical applications and recomputed priorities for interactive and batch processes.

VAX/VMS scheduling is priority-ordered, preemptive, round-robin/time-slicing, and event-driven. The VAX operating system defines 32 levels of priority. Any number of processes can execute at each level.

Of the 32 software levels, levels 16 through 31 are reserved for time-critical processes. Once a priority is assigned to a process, either at login time, by the system manager, or by privilege system service, the system does not modify it.

Each process assigned to levels 0 through 16 is assigned a base priority; the process never drops below its base, and may even receive a temporary boost. The actual priority at which it is scheduled is recomputed by the system (based on the type of process and its execution history) to achieve the best possible balance among all processes. For example, the scheduler tries to maintain a balance between interactive and computer-bound processes; it favors interactive processes and processes doing I/O for faster response.

The scheduler always selects the highest priority, executable, resident process, and schedules by round-robin within the same priority level. The scheduler guarantees execution for a minimum amount of time of any process that it brings into memory from disk.

Events are occurrences that cause the status of a process to change. When the status changes, the process is moved to an appropriate priority queue to be scheduled for execution.

Swapping involves the moving of a process's entire working set in and out of memory in a multiuser environment. Maintenance of the balance set (processes in physical memory) is handled by the swapper, whose function is to keep the system scheduler supplied with executable processes by removing and adding processes to the balance set.

1.3.1.7 File And Record Management Facilities -

Files-11 -- Files-11, the VAX/VMS file manager, provides a facility for the access, creation, extension, and deletion of files. Designed into the file system is a scheme for volume and file protection that allows the owner of a file to deny all access or certain kinds of access to all users, others in the group, the system, and the owner.

Additional file management features include:

Alphabetically ordered directories for faster search

Multi-volume files

Duplication of critical information (index file header and home block) for improved reliability. Loss of this information is fatal to the entire volume.

Improved allocation and deallocation of storage space

Dynamic bad block handling

File protection by user identification codes, passwords, and read/write protection.

RMS/VAX -- Record management services (RMS) is a set of general-purpose file handling capabilities that allow user written application programs to create, access, and maintain data files and records within the files.

1.3.1.8 Network Communication (DECnet) -

VAX/VMS supports DECnet Phase 2.

Access to network facilities is device-independent. A user can exert control over operations to obtain error reports or notification of broken connections. Full login protection applies to all network access.

1.3.1.9 Batch Facilities -

VAX/VMS supports an extensive batch facility that utilizes the same command languages as the interactive portion of the system.

Batch facilities under VAX/VMS include the following.

Multistream Batch -- Jobs may be submitted to batch streams from the interactive environment using a terminal command, from another batch job, or by any program using a system call. Batch jobs are queued, and a time may be specified after which a batch job will execute. The operator controls the number of batch jobs that can run concurrently. Whenever that limit is exceeded, the remaining jobs are held in a queue until a batch stream becomes available.

Job Control -- The method chosen to submit a batch job does not affect the structure of the application program. The user can control the flow of the batch job by specifying what error conditions cause the job to stop, and by suspending execution until an operator loads a device.

Conditional Branching -- The user can specify in a batch stream where program control should transfer on command errors or errors returned by the programs that are executed in the stream.

Spooled Printing -- Printer output is generally spooled to a file-structured volume first, unless the (privileged) program allocates a printer. Print jobs consisting of one or more files to be printed together can be submitted from a terminal, from another batch job, from any program using a system call, and automatically at the end of the batch job. Multiple print queues are also available. The operator defines the number of queues and the user indicates in the print command which one is to be used.

Spooled Input -- Input spooling is available from the card reader.

Operator Control -- The operator can control batch jobs and print jobs, and can stop a job or change its priority.

Accounting -- Batch accounting includes user and account identification, connect time, CPU usage, I/O usage, and page listing count.

In addition to the batch capabilities described, command procedures are supported by the command language. A command procedure is a file that contains a sequence of system operating commands. It is implemented as an indirect command file and can be initiated from an active terminal or from within a batch stream. Command procedures are most often used as an easy method of invoking a series of commands required frequently.

Batch command files and command procedures are completely compatible. Command procedures are executed immediately, within the context of the initiating process and from an active terminal. Batch jobs are queued and executed as detached processes under control of the system.

1.3.2 On-line Diagnostics

An operator can run diagnostics to check the operation of both hardware and software. An operator can run system exercisers and device verification diagnostics while normal operations proceed. System exercisers test general purpose software and compare the results with known answers, reporting any discrepancies to the error logger.

Operators can run device verification diagnostics either stand-alone or concurrently with other processes. Diagnostics check the peripheral functions. In addition, fault isolation diagnostics, which isolate problems to replaceable units, are available for stand-alone use.

1.3.3 Languages

Supported native mode languages include VAX-11 MACRO and, optionally, VAX-11 FORTRAN, VAX-11 COBOL, VAX-11 C, VAX-11 BASIC, VAX-11 PL/I, VAX-11 PASCAL, VAX-11 BLISS-32, and VAX-11 CORAL 66. Also supported in compatibility mode are PDP-11 FORTRAN IV/VAX-to-RSX, and MACRO-11. Development tools for both native and compatibility mode programs include editors, linkers, libraries, and debuggers.

CHAPTER 2

OPERATION

2.1 GENERAL

This chapter contains a description of the VAX-11/785 CPU specifications, cabinet controls and indicators, and the bootstrap operations and procedures.

2.2 VAX-11/785 SPECIFICATIONS

Table 2-1 gives the primary operating characteristics of the VAX-11/785 CPU.

Table 2-1 VAX-11/785 Operating Specifications

CHARACTERISTIC	SPECIFICATION
Operating Environment	
Temperature	15 to 32 degrees C (59 to 90 degrees F)
Relative humidity	20% to 80%
Power Requirements	
ac Power	120/208 volts, 60 Hz
Phases	3 phase 3W+N+G
Input power	3300 watts

2.3 CPU CABINET COMPONENTS

2.3.1 Controls And Indicators

Figure 2-1 shows the CPU cabinet controls and indicators. With the cabinet doors closed, only the console control panel controls and indicators are visible. Access to the other CPU cabinet controls and indicators is necessary only for specific test and maintenance operations.

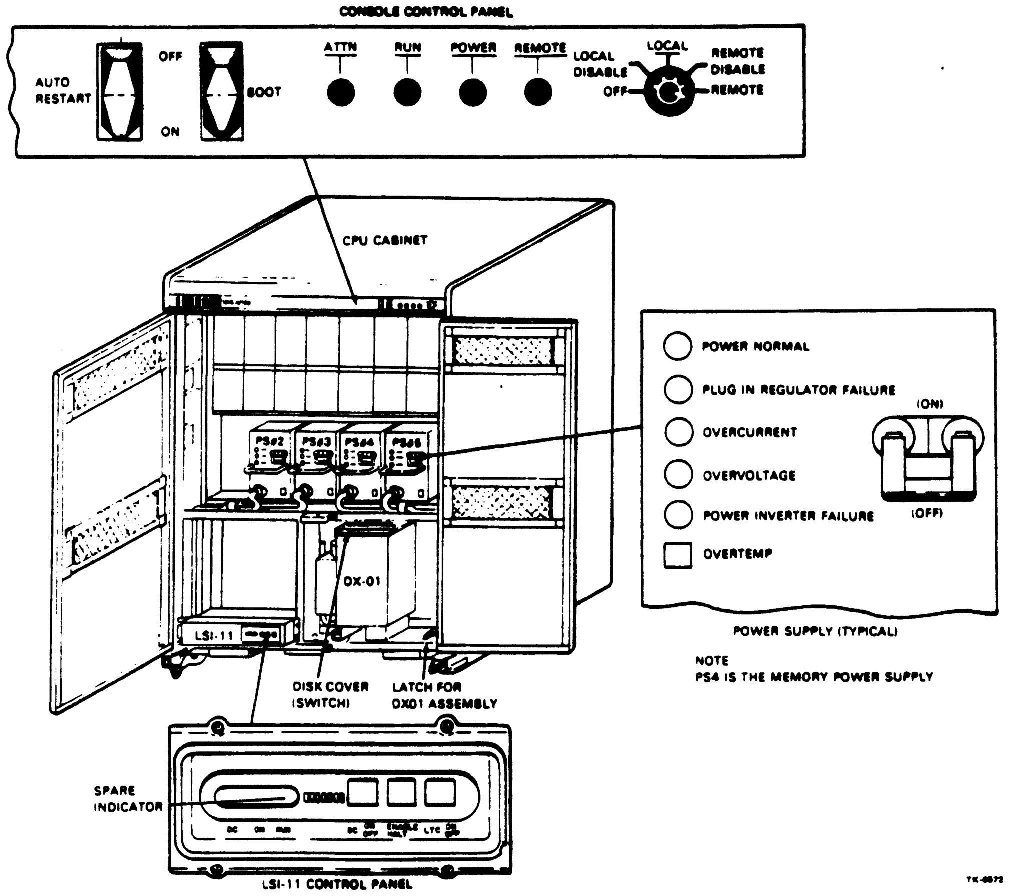


Figure 2-1 CPU Cabinet Controls and Indicators

2.3.1.1 Console Control Panel -

The console control panel has a limited number of controls and indicators. Operations normally associated with a console control panel (such as examine and halt) are performed through the console software package using the console command language. The console control panel is connected to the console interface board. The console control panel controls and indicators are described in Table 2-2.

Table 2-2 Console Control Panel Controls and Indicators

Control/Indicator	Function
AUTO RESTART rocker toggle switch	When in the ON (down) position, the VAX-11/780 CPU is restarted automatically following a power failure or error halt.
BOOT rocker pushbutton switch	When pressed, the operating system is bootstrapped. When the bootstrap operation is completed, the console is set to the program I/O mode of operation.
ATTN indicator (red)	When lit, indicates that the CPU is halted.
RUN indicator (green)	When lit, indicates that the CPU is strobing interrupts.
POWER indicator (green)	When lit, indicates that the +5 V power supply is on.
REMOTE indicator (red)	When lit, indicates that remote access is enabled. When extinguished, indicates that remote access is disabled.
Five position key rotary selection switch	This is an off/on select switch.
OFF position	In this position, the VAX-11/780 is turned off.
LOCAL DISABLE position	In this position, remote access is disabled and console I/O mode of operation is inhibited.
LOCAL position	In this position, remote access is disabled, but console I/O mode of operation is not inhibited.
REMOTE DISABLE position	In this position, remote access is enabled and console I/O mode of operation is not inhibited.
REMOTE position	In this position, remote access is enabled and the console I/O is not inhibited.

2.3.1.2 Power Supplies -

Contained in the CPU cabinet are four outwardly identical power supplies (five if the optional floating-point adaptor is installed). The power supply marked PS4 (Figure 2-1), which is internally different from the others, is the memory power supply. The other power supplies provide power to the remaining logic cards. All power supplies are normally left ON (circuit breaker up). The power supply controls and indicators are described in Table 2-3.

Table 2-3 Power Supply Controls and Indicators

Control/Indicator	Function
POWER NORMAL (green)	When lit, indicates that the power supply is operating normally.
PLUG IN REGULATOR FAILURE (red)	When lit, indicates that there is a malfunction in the plug-in regulator in that power supply. The POWER NORMAL indicator should extinguish.
OVERCURRENT (red)	When lit, indicates that an overcurrent condition exists. The POWER NORMAL indicator should be extinguished.
OVERVOLTAGE (red)	When lit, indicates that an over voltage condition exists. The POWER NORMAL indicator should be extinguished.
POWER INVERTER FAILURE (red)	When lit, indicates that a failure has occurred in the inverter. The POWER NORMAL indicator should be extinguished.
OVERTEMP circuit breaker	When tripped (extended), indicates that an overtemperature condition exists. The POWER NORMAL indicator should be extinguished.
on/off circuit breaker	In the on position the power supply is operational if the POWER NORMAL indicator is lit.

2.3.2 LSI-11 Microcomputer

The PDP-11/03 is a packaged version of the LSI-11 microcomputer. It includes a rack mountable enclosure containing the LSI-11 processor, memory, an LSI-11 bus-structured backplane, a power supply, and a control panel containing three indicators and three switches.

NOTE

The LSI-11 is set to RUN and is not touched unless maintenance operations are required.

2.3.2.1 LSI-11 Controls And Indicators -

The LSI-11 controls and indicators are described in Table 2-4.

Table 2-4 LSI-11 Controls and Indicators

Control/Indicator	**Function**
DC ON LED indicator	Illumintes when the DC ON/OFF toggle switch is set to ON and proper dc output voltages are being produced by the LSI-11. If either the +5 V or +12 V output from the LSI-11 is faulty, the DC ON indicator does not illuminate.
RUN LED indicator	Illuminates when the LSI-11 processor is in the run state (refer to ENABLE/HALT).
SPARE LED indicator	Not used.
DC ON/OFF Two-position toggle switch	When set to ON, enables the dc outputs. The DC ON indicator illuminates if the dc output voltages are of proper values. When set to OFF, the dc outputs are disabled and the DC ON indicator is extinguished.
ENABLE/HALT Two-position toggle switch	When set to ENABLE, the B HALT L line is not asserted and the processor is in the run mode (RUN indicator illuminated). When set to HALT, the B HALT L line is asserted allowing the processor to execute console ODT microcode (RUN indicator extinguished).
LTC ON/OFF Two-position toggle switch	When set to ON, enables the generation of the Line Time Clock (LTC) BEVNT L signal. When set to OFF, disables the line time clock.
AC ON/OFF (rear panel) Two-position toggle switch	When set to ON, applies ac power to the LSI-11. When set to OFF, removes ac power from the LSI-11.
FUSE (rear panel) 5 A fast-slow	Protects LSI-11 from excessive current.

2.3.2.2 Powering-up The LSI-11 -

1. Place the DC ON/OFF switch in the down position (DC OFF).
2. Place the AC ON/OFF switch on the rear of the H780 power supply in the AC ON position.
3. Place the HALT/ENABLE switch in the desired power-up position.

NOTE

> DC power can be applied with the HALT/ENABLE switch in either position. However, processor power-up response is affected by this switch.

4. Place the LTC ON/OFF switch in the OFF position.

5. Place the DC ON/OFF switch in the up (DC ON) position. The console terminal should respond with a printout.

2.3.3 RX01 Diskette Drive

2.3.3.1 Diskette Insertion And Removal -

1. Unlock and open the front doors of the CPU cabinet.

2. Grasp the handle on the side of the diskette drive assembly and swing the drive out perpendicular to the CPU cabinet.

3. Grasp the diskette by the label area and remove it from the protective cover.

4. Squeeze the drive assembly handle with the thumb and forefinger; the handle will spring aside.

CAUTION

> The drive should not be opened while it is being accessed. Data may be recorded incorrectly, resulting in a CRC error when the sector is read.

5. Hold the diskette by the label area and insert it into the drive assembly.

6. Close the drive assembly slot cover.

7. Follow the same procedure to remove the diskette from the drive.

2.3.3.2 Diskette Handling Practices And Precautions -

To prolong the diskette life and prevent errors when recording or reading, reasonable care should be taken when handling the diskette. The following handling recommendations should be followed to prevent unnecessary loss of data or interruption of system operation.

1. Do not write on the envelope containing the diskette. Write any information on a label prior to affixing it to the diskette.

2. Paper clips should not be used on the diskette.

3. Do not use writing instruments that leave flakes, such as lead or grease pencils, on the diskette jacket.

4. Do not touch the disk surface exposed in the diskette slot or index hole.

5. Do not clean the disk in any manner.

6. Keep the diskette away from magnets or tools that may have become magnetized. Any disk exposed to a magnetic field may lose information.

7. Do not expose the diskette to a heat source or sunlight.

8. Always return the diskette to the envelope supplied with it to protect the disk from dust and dirt. Diskettes not being used should be stored in the file box if possible.

9. When the diskette is in use, protect the empty envelope from liquids, dust, and metallic materials.

10. Do not place heavy items on the diskette.

11. Do not store diskettes on top of computer cabinets or in places where dirt can be blown by fans into the diskette interior.

12. If a diskette has been exposed to temperatures outside of the operating range, allow 5 minutes for thermal stabilization before use. The diskette should be removed from its packaging during this time.

2.3.3.3 Diskette Storage -

Short Term (Available for Immediate Use)

1. Store diskettes in their envelopes.

2. Store diskettes horizontally, in piles of ten or fewer. If vertical storage is necessary, the diskettes should be supported so that they do not lean or sag, but they should not be subjected to compressive forces. Permanent deformation may result from improper storage.

3. Store diskettes in an environment similar to that of the operating system; as a minimum, store within the operating environment range.

Long Term

When diskettes do not need to be available for immediate use, they should be stored in their original shipping containers.

The RX01 contains one switch, which is connected to the disk access door. If the door is not closed the disk cannot be run. To insert and remove disks from the RX01, the RX01 must be swung out on its hinges. The RX01 is held in place by a pressure latch on the mounting plate.

2.4 BOOTSTRAPPING

The bootstrapping of the VAX-11/785 is an automatic operation performed when one of the listed conditions exists within the VAX-11/785.

1. The console/CPU power is applied with the control panel AUTO/RESTART switch set to ON (down).
2. The VAX-11/785 has been on and the control panel BOOT switch is pressed.
3. The VAX-11/785 is operational and the CPU enters either a time out/odd address or an unused vector trap.
4. The VAX-11/785 is operational and either the CPU or the clock stops.

Some bootstrap operations require little operator intervention while others require none. Conditions 1 and 2 require operator action.

The description of the series of events during each of these sequences is contained in Appendix B.

2.4.1 Bootstrap Terms

A number of terms for data files and programs are used in describing the bootstrap sequence. These terms are explained here.

1. Console Boot -- The console boot is a program resident in the LSI-11 ROM. It is initiated every time the VAX-11/785 is turned on and every time the console program crashes. This program loads the console program from the RX01 console diskette (disk RX1A AS-T793*-ME) to the LSI-11 RAM.

2. Console Program -- The console program is loaded into the LSI-11 RAM by the console boot. This program contains the console command language interpreter and the VAX-11/785 bootstrap functions.

3. Primary Boot Command File -- The primary boot command file is contained on the RX01 console diskette. This file contains the console commands for loading and starting the primary loader program. These console commands are executed by the indirect command file processor in the console program. There can be more than one primary boot indirect command file, but the only one that will boot automatically is DEFBOO.CMD.

4. Primary Loader -- The primary loader is a VAX/VMS operating system level program that loads a secondary loader program from the system load device. The parameters for the system load device are contained in the primary boot indirect command file. The primary loader file is contained on the RX01 diskette and is loaded into main memory and started by the primary boot command file.

5. Restart Referee -- The restart referee is a ROM operating system level program that is initiated by the VAX-11/785 console program during an automatic restart sequence. The restart referee determines whether a restart will be cold or warm.

6. Memory Sizer -- The memory sizer is a ROM operating system level program that is initiated by the primary boot file during a cold restart or a boot sequence. The memory sizer locates a contiguous block of 64K in memory and sends the starting address to the VAX-11/785 primary boot. The primary boot then loads the primary loader in this location.

7. Secondary Loader -- The secondary loader is an operating system level program that loads the VAX/VMS or diagnostic supervisor from the system load device. The secondary loader is a file stored on the system disk (Files 11 boot block) or tape (ANSI boot record). The secondary loader is loaded and started by the primary loader.

2.4.2 Console (LSI-11) Bootstrap

The LSI-11 bootstrap consists of the following.

1. The console ROM bootstrap program is started by the operator as part of the power-on sequence.
2. A series of LSI-11 tests contained in the console ROM is executed.
3. The console program is loaded from the RX01 diskette to the LSI-11 memory.
4. The console program is initiated.
5. If the AUTO RESTART switch is ON, the CPU bootstrap is initiated.
6. If the AUTO RESTART switch is OFF, the console is held in the console I/O mode of operation (awaiting operator input).

2.4.3 CPU/Memory Bootstrap

The CPU/memory bootstrap consists of the following.

1. The CPU initialization microcode is executed as part of the power-on sequence.
2. The CPU waits for the start of a console boot sequence. The console boot can be initiated by one of the following.

    a. The console BOOT command is entered by the operator.

    b. The console BOOT switch is pressed by the operator.

    c. An auto-restart sequence is initiated. (The AUTO RESTART switch is ON and a warm restart attempt fails).

3. When any one of the preceding conditions occurs, the console loads a bootstrap into the CPU memory from the RX01.
4. The console starts the CPU in the bootstrap just loaded.
5. The console enters the program I/O mode of operation.

2.4.4 Performing A Bootstrap

To perform a VAX-11/785 bootstrap operation from a cold start, dc the following.

1. Insert the console diskette RX1A (AS-T793*-ME) into the RX01 disk drive.

2. Place the AUTO RESTART toggle switch in the ON (down) position.

3. Place the five-position rotary selector switch on the control panel to the LOCAL position.

4. The following is printed out at the console terminal.

```
CPU HALTED,SOMM CLEAR,STEP=NONE,CLOCK=NORM
RAD=HEX,ADD=PHYS,DAT=LONG,FILL=00,REL=00000000
INIT SEQ DONE
HALTED AT 00000000

(RELOADING WCS)
LOAD DONE, 00004800 BYTES LOADED
VER: PCS=01-01 WCS=00-M CON=V06-00-C
```

5. The console enters the program I/O mode of operation and the bootstrap operation is complete.

CHAPTER 3

CONSOLE OPERATOR/PROGRAM COMMUNICATION

3.1 GENERAL

This chapter contains an introduction to the console subsystem and a detailed description of the console command language. The console subsystem is the operator interface for initializing, troubleshooting, and establishing system controls for the VAX-11/785 System.

The console subsystem includes the following hardware:

1. An LA120 terminal and LSI-11 interface
2. An RX01 diskette drive and interface
3. An LSI-11 microcomputer with 48K bytes of memory
4. A console and VAX-11/785 CPU interface
5. A VAX-11/785 control panel
6. An optional remote terminal interface

The primary interface between the operator and the VAX-11/785 is the LA120 terminal.

Integral to the console subsystem is a software interface package. This software interface package provides the means of communication between the operator (at the operator terminal) and the console subsystem, and between the console subsystem and the VAX-11/785 CPU.

3.2 OPERATOR/PROGRAM COMMUNICATION

Communication between the operator and the console subsystem is through the console command language, the diagnostic supervisor, the microdiagnostic monitor, or ODT (online debugging tool). The console command language is described in this chapter. The diagnostic supervisor is described in Chapter 4. Both are English-language based. The microdiagnostic monitor is discussed in VAX Diagnostic User Guide (EK-VX11D-UG). See Microcomputers and Memories Handbook (EB-20912-20) for information on ODT.

3.3 CONSOLE SUBSYSTEM OPERATIONAL MODES

The console subsystem operates in either of two modes, the console I/O mode or the program I/O mode.

In the console I/O mode, operator inputs in the form of console commands are interpreted by the LSI-11 based software interface package. The software interface is loaded into the LSI-11 memory from the floppy disk or main disk during the boot operation.

In the program I/O mode, the inputs of the operator to the terminal are transferred to the CPU. The data is handled by the VAX-11/785 operating system.

3.3.1 Console Panel Functions

The functions performed by the console panel include operating system program control, the entry and display of memory data, and CPU clock control.

3.3.1.1 Program Control -

The console is used for several program control functions: it initializes the CPU using the CCL (console command language) INITIALIZE command; it can initiate and terminate operating system instruction execution; and it is used for bootstrapping the VAX-11/785.

3.3.1.2 Memory Data Entry And Display -

The console can display and modify data contained in the LSI-11 memory, the VAX-11/785 main memory, I/O registers, general registers, and internal registers. The address locations can be read and written in the following quantities:

1. Main memory: byte, word, longword, and quadword.

2. General registers (R0 through R13, SP, and PC) and processor registers: byte, word, longword.

3. I/O registers: byte, word, longword (dependent upon register data length).

4. LSI-11 memory: byte, word.

3.3.1.3 Clock Control -

The CPU clock rate can be controlled via the console. Console clock rate control is used for hardware and software troubleshooting. There are three console clock rate control modes:

1. Single instruction step

2. Single SBI bus cycle step

3. Single CPU time state step.

In the single instruction step mode, the operating system executes one instruction at a time. The CPU enters the halt state after the instruction execution.

In the SBI bus single cycle step mode, the CPU clock is stopped at CPT1 or CPT3 when SBI time state 0 (SBI T0) is set. T0 remains set until another cycle is initiated from the console.

In the single CPU time state step mode, the CPU holds in one of the CPU time states (CPT0, CPT1, CPT2, or CPT3) until a control signal is received from the console. The CPU then advances to and holds in the next time state.

3.3.2 Console Control Functions

Console control functions are provided for the following:

1. Controlling numeric radices of console input
2. Controlling console addressing modes
3. Controlling console data length
4. Displaying console and CPU status
5. Repeating commands
6. Linking commands

3.3.2.1 Default Settings -

The console specifies defaults for the addressing modes, defines the radix of numeric inputs and outputs, and defines the data length of addressable memory. The default options are set by the SET DEFAULT command. Default settings can be overridden within the context of a command by the use of a qualifier or changed with the SET DEFAULT command.

1. The available addressing modes are virtual, physical, console, and register addressing (general and internal).
2. The radices for console numeric inputs and outputs are octal and hexadecimal.
3. The addressable memory data lengths are byte, word, longword, and quadword.
4. The defaults for power up are:

   Radix = hexadecimal

   Data length = longword

   Addressing = physical

3.3.2.2 Status -

The status of the console and CPU subsystems can be displayed on the terminal. The CPU status includes the stop/run state of the ISP (instruction set processor) level software and current clock step mode. The console status includes the current default settings, the console status, and the number of terminal fill characters. The status is displayed for each boot cycle, for each return to the console program, and for each console command language SHOW command.

3.3.2.3 Command Linking And Repeating -

A group of console commands can be linked into a single executable list. The linked commands are entered into a queue. The commands in the queue can be executed once or repeatedly, under the control of the operator. Single commands or linked commands can be repeated. Once initiated, the repeating continues until terminated by the operator. These features are aids to troubleshooting.

3.4 CONSOLE SOFTWARE

The console software package consists of a series of modules. It is contained in the 4K word ROM (microroutine) on the CIB board and in the CONSOL.SYS file on the console diskette. The console software provides for console control, operator interface, the console support function, and the initiation of microdiagnostic testing.

The console is in the null or wait loop the majority of the time. This null loop consists of a series of tests and conditional branches. When any branch condition is met, the flag for the branch is set, and the program branches to the signified routine.

3.5 CONSOLE COMMAND LANGUAGE SYMBOLS AND DEFINITIONS

A number of terms and symbols are used to describe the console command language. A list of these terms and symbols and their definitions is contained in Table 3-1.

Table 3-1 Console Command Language Symbols and Definitions

Term/Symbol	Definition
/	Delimits a command from its qualifiers.
+	Represents the default address when used as an address argument in an examine or deposit command. (The default address is the sum of the last address used plus the current data length in bytes.)
,	Used as a separator within a list.
*	Can be used as an address argument in an examine or deposit command to represent the last address referenced.
:	Used as a separator within a command line.
!	Indicates the exclusive OR operation (i.e., selection of parameters within a command line). For example, <A> ! <B> means either <A> or <B> but not both is to be selected.
>>>	Console command language level input prompt character, requesting operator response.
DS>	Diagnostic supervision command language level input prompt, requesting operator response.
MIC >	Microdebugger input prompt, requesting operator response.
<<<	Link mode input prompt, requesting operator response.
$	VAX/VMS input prompt, requesting user response.
()	Indicates that one of the syntactic units of the expression are to be selected.
[ ]	Indicates the part of an expression that is optional e.g., NEXT [<blank> <count>] indicates that the count argument for the NEXT command is optional.
< >	Denotes a category name or label e.g., <address> indicates that a valid address is to be inserted as part of the command.
<blank>	Represents one or more blank spaces or tabs.
<count>	Represents a numeric count argument.
<XYZ - list>	Indicates one or more occurrences from the category indicated by XYZ.
<address>	Represents an address argument.
<data>	Represents a numeric argument.
<qualifier>	Represents a command modifier or default.
<input prompt>	Represents the console's input prompt string.
<CR>	Represents a console terminal carriage return. Used to terminate each command line.
<LF>	Represents a console terminal line feed.

3.5.1 Notation Examples

The following is an example of notation within a command line.

EXAMINE [</qualifier - list>] [<blank> <address>]

For each EXAMINE command there are two options.

1. A list of one or more qualifiers.
2. An address argument. If an address is specified, it must be preceded by one or more spaces or tabs.

The following are valid EXAMINE commands:

EXAMINE

No qualifiers are entered. The examine qualifiers are then entered under the SET DEFAULT command.

EXAMINE/BYTE/VIRTUAL

Word and address qualifiers entered but the address is controlled by default entered under the SET DEFAULT command.

EXAMINE <space> 1234 or
EXAMINE <tab> 1234

No qualifiers selected. The qualifiers are controlled by th default entry in the SET DEFAUI command.

EXAMINE/WORD <space> 1234 or
EXAMINE/WORD <space> <tab> 1234

Word qualifier entered (address by default) a memory argument i entered.

3.5.2 Command Abbreviations

Console command words may be abbreviated. The minimum abbreviation for each command word is specified in parentheses in each console command description.

```
Example
>>>EXAMINE/VIRTUAL/BYTE 1234 may be abbreviated to:
>>>E/V/B 1234
```

If an incorrect command entry is given, the console software responds by repeating the command and stating that it is incorrect.

```
Example
>>>HLP
(>>> ? 'HLP' is incorrect.)
```

3.6 CONSOLE COMMANDS

The console commands are English-language words or their accepted abbreviations entered by the operator at the console terminal. These commands can be accepted only when the console is in the console I/O mode. When a valid command is entered, the indicated sequence of events is initiated. When that set of events is completed, another console command can be entered.

The console commands are described in the following paragraphs. These descriptions include, as applicable, the following items.

1. Syntax -- the structure of the command.

2. Command description -- a brief description of the command operations, general restrictions, and available options.

3. Response -- a description of the console program response to the command.

The terms and symbols used in the descriptions are defined in Table 3-1.

NOTE

> Every console command line must be terminated with a carriage return (<CR>).

3.6.1 BOOT Command (B)

Syntax -- BOOT [<device name>] <CR>

The optional <device name> has the format DDn. DD is the two-letter device mnemonic and n is a one-digit unit number.

Description -- The BOOT command initiates a VAX-11/785 system bootstrap sequence. The command can support bootstrap operations from or for a set of VAX-11/785 system devices.

When an optional <device name> is not given with the BOOT command, the console performs the boot sequence for the device preset under the default listing. The boot program DEFBOO.CMD, written on the console diskette RX1A, is executed.

When an optional <device name> is given with the command, the console executes boot program DDBOO.CMD where DDN is the device name given in the command.

Bootstraps from other than the system default device are performed by indirect command files containing the console commands necessary to boot from the device named.

Response -- The status of the CPU and CPU controls is printed on the console terminal at the completion of a boot operation.

```
>>>BOOT

        CPU HALTED
        INIT SEQ DONE
        HALT INST EXECUTED
        HALTED AT 00000000

        G  0000000E  00000200
        LOAD DONE, 00004400 BYTES LOADED
```

With this command the indirect command file DRBOO.CMD is booted. The console enters the program I/O mode after executing the command file.

Table 3-2 is a list of device names for VAX-11/785 load devices.

Table 3-2 Load Device Names

LOAD DEVICE	DEVICE NAME
RA60	DUan
RA80	DUan
RA81	DUan
RM05	DRan
RP07	DBan
TU77	MTan
TU78	MFan
TU80	MSan

3.6.2 CONTINUE Command (C)

Syntax -- CONTINUE <CR>

Description -- The CONTINUE command causes the CPU to begin operating system execution at the address currently contained in the CPU program counter (PC). The CPU is not initialized. The console subsystem enters the program I/O mode after the CONTINUE command is executed.

```
    Response
    >>>CONTINUE
```

3.6.3 DEPOSIT Command (D)

Syntax -- DEPOSIT [<qualifier - list>] <blank> <address> <blank> <data> <CR>

The qualifiers are /BYTE, /WORD, /LONG, /QUAD, /NEXT, /VIRTUAL, /PHYSICAL, /IDBUS, /VBUS, /INTERNAL, /GENERAL, and /CONSOLE. Refer to Table 3-4 for a description of these qualifiers.

The DEPOSIT command writes <data> into the specified <address>. The address space used depends on the qualifiers specified with the command. If no qualifiers are used, the data and address defaults (entered via the SET DEFAULT command) are used to determine the address space.

```
    Response
    >>>DEPOSIT 10AD  310014DA
    >>>
```

The <address> argument may be either a number identifying a location in one of the address options or one of the symbolic addresses. The symbolic address symbols and a description of them are contained in Table 3-3.

Table 3-3 Symbolic Address Symbols

Symbol	Definition
PSL	Deposits to or examines the processor status longword.
PC	Deposits to or examines the program counter (general register F).
SP	Deposits to or examines the stack pointer (general register E).
+	Deposits to or examines the location immediately following the last location referenced. For physical and virtual references the location referenced is the last address plus n where n = (1 for byte, 2 for word, 4 for longword, 8 for quadword). For all other address spaces, n is equal to 1.
-	Deposits to or examines the location immediately preceding the last location referenced.
*	Deposits to or examines the location last referenced.
@	Deposits to or examines the address represented by the last data examined or deposited.

3.6.4 DIRECTORY Command (DIR)

Syntax -- DIRECTORY <CR>

Description -- The DIRECTORY command provides the user with a simple directory of an RT-11 format diskette. The diskette must be in the RX01 drive, and the console must be in console mode. File size and creation date are not provided.

3.6.5 EXAMINE Command (E)

Syntax -- EXAMINE [<qualifier - list>] [<blank> <address>] <CR>

The qualifiers are /BYTE, /WORD, /LONG, /QUAD, /NEXT, /VIRTUAL, /PHYSICAL, /IDBUS, /VBUS, /INTERNAL, /GENERAL, and /CONSOLE.

Description -- The EXAMINE command reads and displays the contents of the specified <address>. If no <address> is specified, the contents of the address defined by the address default (entered via the SET DEFAULT command) are displayed.

The <address> argument may be one of the symbolic addresses (Table 3-3).

Response -- Sample command entries and console responses

```
>>>EXAMINE/PHYSICAL 1234

        P 00001234  EF89
>>>EXAMINE/VIRTUAL  1234

        P 0005634  4567
(The translated physical address is displayed for virtual
examines.)

>>>EXAMINE/G  0

        G 00000000  5432
>>>EXAMINE/BYTE 1234

        P 00001234  78
>>>EXAMINE/WORD 1234

        P 00001234  5678
>>>EXAMINE/LONG

        P 00001236  FFFF1234
>>>
```

3.6.6 INITIALIZE Command (I)

Syntax -- INITIALIZE <CR>

Description -- This command sets the CPU to a specified starting condition.

```
    Response
    >>>INITIALIZE

             INIT SEQ DONE

    >>>
```

3.6.7 HALT Command (H)

Syntax -- HALT <CR>

Description -- The HALT command stops the CPU after it completes execution of the instruction that is in process when the HALT command is entered.

```
    Response
    >>>HALT

             HALTED AT   00002436

    >>>
```

The number given is the location at which the process is halted.

3.6.8 HELP Command

Syntax -- HELP <CR>

Description -- The console opens and prints the contents of the console help file, an indirect command file. The help file contains a description of all console commands.

```
    Response
    >>>HELP

    ! CONSOL.HLP - VAX-11/785 Console Help File
    ! Rev: 0.2, 03-Nov-1983
    ! Copyright (c) 1983, Digital Equipment Corporation
        .
        .
        .
    <@EOF>
    <@EXIT>

    >>>
```

Refer to Appendix D for a sample printout of the console help file.

3.6.9 Error Help Command

Syntax -- @ERROR.HLP <CR>

Description -- The console opens and prints the contents of the error help file, an indirect command file. The error help file contains a list of all error messages for the console command language.

Response

```
>>>@ERROR.HLP

! ERROR.HLP - VAX-11/785 Error Message Help File
! Rev: 0.1, 10-Oct-1983
! Copyright (c) 1983, Digital Equipment Corporation
    .
    .
    .
<@EOF>
<@EXIT>

>>>
```

Refer to Appendix E for a sample printout of the console error help file.

3.6.10 LOAD Command

Syntax -- LOAD [<qualifier list>] <blank> <file specification> <CR>

The <qualifier list> is the storage location for the file being loaded; and the <file specification> is the identity of the file to be loaded.

Description -- The LOAD command is used to read or transfer file data from the console floppy disk to main memory, or to Joint Control Store (JCS). If no qualifier is entered, physical main memory, location 0, is the default location (entered via the SET DEFAULT command) to begin loading. The load defaults are described in Table 3-4.

```
    Response
    >>>LOAD ESSAA.EXE/START:FE00
             LOAD DONE. 000(000BYTE)LOADED

    >>>START 10000

    DS>
```

The diagnostic supervisor file is loaded into main memory beginning at location FE00, and it is started at 10000.

Table 3-4 Load Command Qualifiers

Qualifier	Definition
/START:<address>	This qualifier specifies a starting address for the file loaded. The console starts loading at address 0 if a qualifier is not given.
/WCS	The WCS is loaded with the specified file.
/PHYSICAL	The physical main memory is loaded with the specified file.

3.6.11 NEXT Command

Syntax -- NEXT [<blank> <count>] <CR>

Description -- With the NEXT command the CPU clock is stepped the number of times indicated by <count>. The type of step performed by the clock is determined by the previous SET STEP command. If a NEXT command is issued while the CPU is in normal clock mode, it defaults to single instruction step mode for the duration <count> of the command. <count> is in hexadecimal or octal, depending on the current default setting.

The console enters program I/O mode immediately prior to performing the STEP command. The console I/O mode is re-entered as soon as the STEP command is complete.

Response -- Step-dependent responses are displayed on the console terminal after the completion of each of the count steps as specified in the following.

a. Single Instruction Mode (SET STEP instruction).

```
>>>SET STEP INSTRUCTION

>>>ST 100

    HALTED AT 00000102

>>>NEXT

    HALTED AT 00000104
    HALTED AT 00000106
```

b. Single Bus Cycle Mode (SET STEP BUS)

```
>>>SET STEP BUS

>>>
     CPU CLK STOP
     CPT0 UPC=00FF

>>>NEXT

     CPT0 UPC-00FF
     CPT0 UPC-00FF
     CPT0 UPC-00FF
     CPT0 UPC-00FF
     CPT0 UPC-00FF

>>>NEXT 2

     CPT0 UPC-00FF
     CPT0 UPC-00FF
```

c. Single Time State Mode (SET STEP STATE)

```
>>>SET STEP STATE

>>>NEXT 6

     CPT1
     CPT2
     CPT3 APC=0000
     CPT0 UPC=00FF
     CPT1
     CPT2

>>>SET STEP NORMAL

>>>
```

If a <count> is not specified in the command, one step is performed, and then the console enters the space bar step mode. For each depression of the space bar there is one execution of the STEP option (BUS, TIME, INSTRUCTION) currently enabled.

If the NEXT command has an argument the space bar step mode is not enabled. For example, for a NEXT 2 input, two steps are executed and the console then prompts for another command.

To exit from the space bar step mode, enter any character except a space.

3.6.12 QUAD CLEAR Command (Q Clear)

Syntax -- QCLEAR <blank> <physical address> <CR>

Description -- The QUAD CLEAR command clears the quadword at the specified <address>. The command is used to clear an uncorrectable ECC error.

```
    Response
    >>>QCLEAR 4444

    >>>
```

The <address> is always interpreted as a physical main memory location. The <address> given is forced to a quadword boundary by unconditionally clearing the three low-order address bits.

3.6.13 REPEAT Command (R)

Syntax -- REPEAT <console command> <CR>

Description -- With a REPEAT command the specified console command is repeatedly executed until terminated by a CTRL C (^||C). Any console command, except the REPEAT command, may be specified.

```
    Response
    >>>REPEAT EXAMINE 100

         P   00000100   00000000
         P   00000100   00000000
         P   00000100   00000000
         P   00000100   00000000
         P   00000100   00000000
         P   00000100   00000000
         P   00000100   00000000
         P   00000100   00000000
       ^C

    >>>
```

The repeat response is dependent on the command specified to be repeated.

3.6.14 SET CLOCK Command (SE)

Syntax -- SET <blank> CLOCK [<blank> (SLOW FAST NORMAL)] <CR>

Description -- The SET CLOCK command sets the CPU clock to the speed specified by the command argument (FAST, SLOW, NORMAL). The clock is stopped (even if it is running at the speed specified in the command), the speed is changed, and the clock is restarted. The speeds are as follows:

Fast = 130 ns

Slow = 148 ns

Normal (or no argument) = 133 ns

The SET CLOCK command must be used when changing clock speed.

Response

```
>>>SET CLOCK SLOW

>>>
```

3.6.15 SET DEFAULT Command

Syntax -- SET <blank> DEFAULT [<blank> <default option>] <CR>

Description -- The SET DEFAULT command is used to set the defaults for address, data length, and radix. The <default options> and their definitions are contained in Table 3-5. The console applies the defaults from those set under this command when a console command does not contain a qualifier.

Response

```
>>>SET DEFAULT PHYSICAL, BYTE, HEX

>>>
```

Table 3-5 Set Default Command Options

Option Group	Item	Specification
Address defaults	Virtual	Sets the default to the specified address.
	Physical General Internal ID bus V bus CONSOLE	
Data defaults	Byte	Sets the default to the specified data length.
	Word Long Quad	
Radix defaults	Hex	Sets the default for the termi-nal I/O to the specified radix.
	Octal	

3.6.16 SET STEP Command

Syntax -- SET <blank> STEP [<blank> <step option>] <CR>

Description -- The SET STEP command sets the CPU clock mode. The CPU clock modes (<step option>) are defined and listed in Table 3-6.

```
    Response
    >>>SET STEP BUS

    >>>
```

Table 3-6 Set Step Command Options

Option Group	Item	Specification
Step	Instruction	Sets the CPU clock to the single instruction step mode.
	Bus	Sets CPU clock to the single Synchronous Backplane Interconn ect (SBI) cycle step mode.
	State	Sets CPU clock to the single SBI time state step mode

3.6.17 SET TERMINAL FILL Command

Syntax -- SET <blank> TERMINAL FILL <count> <CR>

Description -- The SET TERMINAL FILL command sets the number (<count>) of blanks that are transmitted to the console terminal after a carriage return (<CR>) or a line feed (<LF>).

```
    Response
    >>>SET TERMINAL FILL:4

    >>>
```

3.6.18 SET TERMINAL PROGRAM Command

Syntax -- SET TERMINAL PROGRAM <CR>

Description -- The SET TERMINAL PROGRAM command places the console terminal in the program I/O mode of operation. This command returns control to a program that was previously interrupted by a ^P. SET TERMINAL PROGRAM returns the processor from console mode to program I/O mode.

```
    Response
    >>>SET TERMINAL PROGRAM
```

3.6.19 SET RELOCATION Command

Syntax -- SET RELOCATION: <data> <CR>

Description -- The SET RELOCATION command deposits data into the console's relocation register. The value <data> set in the relocation register is added to the effective address of all virtual and physical memory examines and deposits.

```
    Response
    >>> SET RELOCATION: 1000

    >>> EXAMINE 200       (With the relocation set to 1000 an examine
                          200 command examines the contents of
                          physical memory location 1200.)

             P  00001200  49

    >>>
```

3.6.20 SET/CLEAR SOMM Command

Syntax -- SET <blank> SOMM <CR>

Description -- The SET SOMM command sets the Stop On Microbreak Match (SOMM) enable (on the console interface board). The CPU clock is stopped if SOMM is enabled and the contents of the microbreak match register (ID register 21), match the contents of the CPU micro-PC.

The CLEAR SOMM command disables the Stop On Microbreak Match.

```
    Response
    >>>SET SOMM

    >>>
```

3.6.21 SHOW Command (SH)

Syntax -- SHOW <CR>

Description -- The SHOW command displays the following at the console terminal:

a. The default settings for data length, address type, and radix of address and data inputs and outputs

b. The terminal fill character count

c. The CPU status including the run halt state and current clock mode setting.

```
    Response
    >>>SHOW

             CPU HALTED,SOMM CLEAR,STEP=NONE,CLOCK=NORM
             RAD=HEX,ADD=PHYS,DAT=LONG,FILL=00,REL=00000000

    >>>
```

3.6.22 START Command (S)

There are two formats used with the START command.

```
Syntax --  START <blank> <address> <CR>
       or  START/WCS <blank> <address> <CR>
```

Description -- The START command of the first format performs the equivalent of the following sequence of console commands:

1. Initialize the CPU.

2. Deposit the <address> data in the PC.

3. Issue a CONTINUE instruction to the CPU.

The START command of the second format performs the equivalent of the following sequence of commands:

1. Deposit the <address> data in the micro-PC.

2. Start the CPU clock in the normal mode of operation.

```
    Response
    >>>START 100

    or

    >>>START/WCS FF
```

3.6.23 TEST Command (T)

Syntax -- TEST [/COM] <CR>

Description -- The TEST command enables the microdiagnostic monitor program. Microdiagnostic execution begins immediately if no /COM qualifier is entered. If microdiagnostic testing is completed and no errors are detected, the console I/O mode is re-entered. A microdiagnostic diskette must be loaded in the RX01.

With a [/COM] qualifier the microdiagnostic monitor enters its command mode (MIC>) and waits for an operator command.

```
    Response
    >>>TEST

    ZZ-ETKAB
    01,02,03,04,
    NO. OF WCS MODULES  = 0002
    05,06,07,08,09,0A,0B,0C,0D,0E,0F,10,11,12,13,14,15,16,17,
    18,19,1A,1B,1C,1D,1E,1F,20,21,22,23,24,25,26,27,28,29,2A,2B,2C,2D,
    2E,2F,30,31,32,33,34,35,36,37,38,39,3A,3B,3C,3D,
    END PASS 0001
    MOUNT FLOPPY & TYPE "DI"
    MIC>DI
    3E,
    # MEM CTRLS= 00000001
    3F,40,
    4K CHIP 00000608
    41,42,
    CPU TR= 00000010
    43,44,45,46,47,48,49,4A,4B,4C,4D,
    CTRL 1 MAX ADR+1= 00040000
    4E,
    CTRL 1 MAX ADR+1= 00040000
    4F,50,
    NO FPA

    END PASS 0001

    MIC>
```

After running the first microdiagnostic test program, the operator is instructed to mount the diskette containing the second diagnostic test. A microdiagnostic prompt (MIC>) appears. The second hits a trap and responds with a microdiagnostic prompt. The operator returns to the console command program by typing RET or RET/DI. RET causes the console to perform a boot operation; RET/DI returns the console prompt (>>>).

3.6.24 UNJAM Command (U)

Syntax -- UNJAM <CR>

Description -- This command initiates a clearing (unjam) of the SBI.

```
    Response
    >>>UNJAM

    >>>
```

3.6.25 WAIT Command (WA)

Syntax -- WAIT <blank> DONE <CR>

Description -- The WAIT command is executed from an indirect command file. When executed, further execution of the command file is suspended until one of the following occurs:

a. A DONE signal is received from a program running in the CPU. On receipt of DONE, the console resumes execution of the command file.

b. The console prints <@EXIT> and aborts execution of the remainder of the console command file if the CPU halts and a DONE signal is not received.

c. The operator enters a CTRL C. The console aborts execution of the remainder of the command file.

```
    Response
    WAIT DONE
    ^||C

    >>>
```

3.6.26 Indirect (@) Command

Syntax -- @<filename> <CR>

Description -- This command causes the console to open the file specified by <filename> and begin executing console commands from the file. Execution continues until one of the following events occurs:

a. A WAIT DONE command is read from the file (refer to WAIT command).

b. The end of the indirect file is reached. The console prints <@EOF> and an <input prompt>.

c. The operator enters a CTRL C (^| |C). Execution of the indirect file is aborted.

Response

```
>>> @BOOT.HLP

! BOOT.HLP - VAX-11/785 Bootstrap Help File
    .
    .
    .
<@EOF>
<EXIT>

>>>
```

An indirect command file may also be entered using the following sequence. Note the lack of prompt characters once this routine has been entered.

```
>>>@ESCAA

HALT         ! HALT THE CPU

             CPU HALTED
INIT         ! INITIALIZE

             INIT SEQ DONE
UNJAM        ! AND CLEAR THE SBI

LO ESCAA.EXE/ST:200

             LOAD DONE, 0000B400 BYTES LOADED
!            RH780 MASSBUS DIAGNOSTIC

LO ESSAA.EXE/ST:FE00

             LOAD DONE, 0000D000 BYTES LOADED
START 10000

<@EOF>
<@EXIT>

DIAGNOSTIC SUPERVISOR.  ZZ-ESSAA-4.02-417 21-JUL-1978 01:33:44.92

DS>
>>>H
             HALTED AT 00016B04
>>>
```

3.7 CONSOLE COMMANDS PERFORMED WITH CPU RUNNING

Most console commands have an impact on the running of the CPU, so the CPU must be halted before executing these commands. However, some console commands do not require interaction with the CPU, and they may be executed with the CPU running. These commands include:

a. SHOW

b. HELP

c. the SET commands

d. EXAMINE/V BUS

e. HALT

f. CLEAR

g. WAIT

If any other console command is entered while the CPU is running, the command is rejected and the following error message is displayed on the console terminal:

```
? CPU NOT IN CONSOLE WAIT LOOP, COMMAND ABORTED

>>>
```

3.8 COMMENTS WITHIN COMMANDS

Comments, preceded by an exclamation mark (!), may be used in any command line. When the console detects an exclamation mark, all remaining text in that command line is ignored. A comment may begin in any character position within a command line, including the first. (The ! is considered a part of the comment.)

Example

```
>>> ! THIS IS A VALID COMMENT

>>>
```

The above comment follows directly after the console prompt.

```
>>> EXAMINE 1234 ! THIS IS ALSO A COMMENT

>>>
```

This comment is made after a command, prior to the carriage return.

3.9 CONTROL AND SPECIAL CHARACTERS

There are a number of control characters and special characters that cause an action to occur within the VAX-11/785 when entered independently. These form an integral part of the console command language. A list of the control and special characters is contained in Table 3-7.

Table 3-7 Control and Special Characters Descriptions

Character	Description
CTRL C (↑C)	When a CTRL C is typed any repetitive console operation is halted. This includes the following: a. Executions resulting from a repeat command b. Successive operations resulting from a /next qualifier c. Delays resulting from a wait command d. Successive steps resulting from a next command e. Aborts further execution of an indirect command file after current instruction is completed. The console responds with an <input prompt>.
CTRL O (↑0)	When a CTRL O is entered, the console terminal output is suppressed or enabled (on a toggle basis). Console terminal output is always enabled at the next console terminal <input prompt>.
CTRL U (↑U)	When a CTRL U is typed before a line terminator, all characters typed since the last line terminator are deleted from temporary storage. The console echoes: U <CR> <LF>
DELETE	Typing DELETE (the DELETE key) deletes the last character typed on an input line. Only characters typed since the last line terminator can be deleted. Several characters can be deleted in a sequence by typing successive deletes. The first delete echoes as a backslash (\) followed by the character which has been deleted. Subsequent deletes cause only the deleted character to be echoed. The next character typed that is not a delete causes another backslash (\) to be printed, followed by the new character.
CTRL P (↑P)	When a CTRL P is entered the console is placed in the console I/O mode. (unless the console mode switch is in the LOCAL/DISABLE or REMOTE/DISABLE position).
CARRIAGE RETURN (CR)	When a carriage return is entered a console command line is terminated.

3.10 COMMAND QUALIFIERS AND DEFAULTS

Qualifiers are entered as part of a command line and further define the basic console command. Defaults are supplied by the console software when the operator does not exercise the option of entering a qualifier. The defaults can be changed by using the set default console command.

The following paragraphs describe the qualifiers and defaults for address and data arguments; qualifiers and defaults for address type, data length, and numeric argument defaults; and a qualifier to operate on multiple sequential addresses.

3.10.1 Qualifiers For Address Type

Qualifiers for address type are used within a command line. These qualifiers specify the type of address argument as virtual, physical, or register address. The qualifiers for the respective types are:

```
/VIRTUAL
/PHYSICAL
/IDBUS
/VBUS
/GENERAL
/INTERNAL
/CONSOLE
```

Example
To examine virtual address 1234, type:

```
>>>EXAMINE/VIRTUAL 8000E3FC
```

The system responds by displaying the contents of that address

```
P 0002E3FC 0000ABCD

>>>
```

Some register addresses have mnemonic names that are unique. Therefore, the /GENERAL qualifier need not be specified when mnemonic addresses such as PC or SP are referenced.

Example
To examine the PC, the operator enters either of the following statements:

```
>>>EXAMINE/GENERAL PC
```

or

```
>>> E PC
```

The system responds by displaying the contents of the PC.

```
G 0000000F 00000001

>>>
```

If a console command references a nonexistent virtual address, the execution of that command is aborted. An error message is then typed at the console terminal.

3.10.2 Defaults For Address Types

The console applies a default for address type to any command that requires an address argument and does not contain an address qualifier. The default is set by using the SET DEFAULT command.

Example

```
>>> SET DEFAULT VIRTUAL

>>>
```

After executing this command the console defaults to virtual addressing for any console command that requires an address argument but does not have an address qualifier.

With virtual default set, the command EXAMINE 1234 results in the contents of virtual address 1234 being typed on the terminal.

3.10.3 Qualifiers For Data Length

Qualifiers for data length are used within a command line to specify the length of the data quantity associated with the command. The qualifiers for data length are:

BYTE
WORD
LONG
QUAD

Example

```
>>>EXAMINE/BYTE 1233

P  00001233  FF

>>>
```

The data (FF) in the byte at address 1233 is typed at the terminal.

Since the general and processor registers are longword quantities, all register references default to longword data length, regardless of the current setting of the data length default.

3.10.4 Defaults For Data Length

The console applies a default for data length to any command that references data and does not contain a data length qualifier. The default can be set through the SET DEFAULT command.

Example

The command

```
>>>SET DEFAULT WORD

>>>
```

results in the console defaulting to word data length.

```
>>>EXAMINE 1234
```

The system responds by displaying the contents of the word that has its first byte at address 1234.

```
P 00001234 FFFF

>>>
```

All general and processor registers are longword quantities. Therefore, all register references default to longword data length, regardless of the current setting of the data length default.

3.10.5 Local Radix Override

The current default radix can be overridden by including a local radix override as a prefix to any numeric argument. A local radix override is one of the following:

%O (percent sign O) for octal arguments

```
>>>EXAMINE %O7654

P 00000FAC FFFF

>>>
```

%X (percent sign X) for hexadecimal arguments

```
>>>EXAMINE %X7654

P 00000073124 177777

>>>
```

A local radix override must appear as the two leftmost characters of the numeric argument it modifies, and it must not be separated by spaces or tabs from that argument.

Defaults For Radix -- The console allows setting the radix default to octal or hexadecimal for address and data arguments. The console does not allow a decimal radix.

Example

```
>>>SET DEFAULT OCTAL

>>>
```

The default radix for address and data arguments is set to octal.

3.10.6 Default Address Facility

Every time an EXAMINE or DEPOSIT command is executed, the address of the memory location following the location referenced by the command is computed. The address of the next memory location is the <default address>. An EXAMINE command that does not specify an address references the next address by default. The <default address> is computed as follows:

> <default address> = <address used by last EXAMINE or DEPOSIT command> + n
>
> where n = 1 for byte references\
> 2 for word references\
> 4 for longword references\
> 8 for quadword references

The following example shows a sequence of console commands and the value taken by the computed default <address>. Note that the next address is data length dependent, since a byte reference advances the default address by 1, while a longword reference advances the default address by 4.

Example (all numbers are octal)

```
Command                            Default address after execution

>>>EXAMINE/BYTE 1234

P 00001234 78                                 1235
>>>EXAMINE/WORD (uses default address)

P  00001235  3456                             1237
>>>EXAMINE/LONG (uses default address)

P 00001237 FFFFFF12                           1243
>>>EXAMINE R0

G 00000000 00000010                           R1
>>>EXAMINE

G 00000001 00002000                           R2
>>>EXAMINE R11

G 0000000B 04000000                           R12
>>>EXAMINE

G 0000000C 00001C00                           R13
>>>EXAMINE PC
```

```
G 0000000F 00000001                           R0
>>>EXAMINE

G 00000000 00000010                           R1

>>>
```

Note that the <default address> following a PC reference is R0.

3.10.6.1 Specifying Defaults For Address -

The symbol (+) can be used as an address argument in a DEPOSIT or EXAMINE command to represent the <default address>. This symbol permits depositing to (or examining) successive locations without typing the address argument for commands after the first deposit.

Example
To load a program starting at address 12345, the following DEPOSIT commands can be used:

```
>>>DEPOSIT     12345          987654

>>>DEPOSIT     +              12345

>>>DEPOSIT     +              6543

>>>
```

Each DEPOSIT command, after the first, writes the <data> into the next successive memory location. This same technique can be used with the EXAMINE command.

```
>>>EXAMINE 12345

               P               00012345 00987654
>>>EX +

               P               00012349 00012345
>>>EX +

               P               0001234D 00006543
>>>
```

3.10.6.2 Last Address Notation -

The last address argument referenced (virtual, physical, register) by an EXAMINE or DEPOSIT command is denoted by an asterisk (*). The <last address> may be used as an argument to an EXAMINE or DEPOSIT command by typing an asterisk in place of the address argument.

Example

```
>>>EXAMINE 1234

P 00001234 12345678
```

The contents of location 1234 is displayed.

If the next command issued is:

```
>>>DEPOSIT * FFFF
```

The number FFFF is loaded into location 1234.

```
>>>EXAMINE *

P 00001234 0000FFFF

>>>
```

Examines and deposits of general and processor registers replace the <last address> with the register address. Mnemonic register names are translated into register addresses by the console.

3.10.7 NEXT Qualifier

Syntax -- Command/NEXT [:<COUNT>]

The <count> argument specifies the number of additional executions of the command to be performed after the initial execution of the command. The default value for <count> is one. The radix for <count> is octal or hexadecimal, depending on the radix set previously in the SET DEFAULT command.

Example

The command

```
>>>EXAMINE/BYTE 12345/NEXT:2

P 00012345 54
P 00012346 76
P 00012347 98
```

is evaluated by the console as follows:

1. Initially the console applies default values.

2. The command, with applied defaults, is executed. The contents of location 12345 are displayed, and the <default address> is updated to 12346.

3. The qualifier /NEXT is now evaluated. The console repeats the operation specified by the command the number of times indicated by the <count> argument. Each execution of the command uses the <default address> for its address argument, then updates the <default address>. In this example, locations 12345 and 12346 are displayed and the final value of the <default address> is 12347.

Example

If the command

```
              >>>EXAMINE/NEXT:2

              P 00012348 01234500
              P 0001234C 00654300
              P 00012350 FFFFFF00
```

is entered (following the command in the previous example) the contents of locations 12348, 1234C, and 12350 are displayed. The default is longword. Since the EXAMINE command does not contain an address argument, the initial execution of the command uses the current <default address> (which was 12347 following the command in the previous example).

When the qualifier /NEXT is used to examine or deposit multiple general registers, the next register after PC is R0.

Example

The command

```
              >>>EXAMINE/NEXT:5 R13

              G 0000000D F0F0F0F0
              G 0000000E 8C000000
              G 0000000F 00000001
              G 00000000 00000010
              G 00000001 00002000
              G 00000002 00000800

              >>>
```

results in the contents of R13, SP, PC, R0, R1, and R2 being displayed (in that order).

```
>>>EXAMINE/BYTE 12345/NEXT:2

P 00012345 54
P 00012346 76
P 00012347 98

>>>EXAMINE/WORD 12345/NEXT:2

P 00012345 7654
P 00012347 0098
P 00012349 2345

>>>
```

The preceding commands are used to examine the contents of the same location.

3.11 COMMAND REPEAT FACILITY

The command repeat facility is primarily a maintenance aid. Commands are executed repeatedly so that CPU logic involved with the commands can be tested with an oscilloscope.

Example

```
>>>REPEAT EXAMINE 1234

P 00001234 0000FFFF
P 00001234 0000FFFF
P 00001234 0000FFFF
P 00001234 0000FFFF
P 00001234 0000FFFF
P 00001234 0000FFFF
P 00001234^||C 0000FFFF
```

This results in the content of location 1234 being repeatedly examined and displayed.

Once initiated, repeated execution continues until terminated by the entry of a CTRL C (^||C) on the console keyboard.

3.12 COMMAND LINK FACILITY

The console's command link facility provides a means for multiple commands to be linked into a single executable list of commands. Once a list of linked commands is constructed, the list can be executed once or repeatedly.

3.12.1 LINK Command Operation

The LINK command puts the console in the link mode. The desired commands are then entered. The LINK command is entered only as the first command of a command string. Commands to be linked are entered one to a line, with each command line terminated by a <CR>. The console then returns a link mode input prompt (<<<) requesting the next command. The link operation is terminated by entering a CTRL C (^||C) on the console.

As the command string is entered, the commands are stored on a dedicated sector on the diskette (RX01). When the command string is executed, the string is treated as an indirect command file (the command is retrieved, parsed, and executed, then the next command is retrieved, parsed, etc.).

Once the input of a list of linked commands has been terminated, no further commands can be added to the commands already linked. The linked commands are executed by entering a PERFORM command. The console does not execute commands being linked until a PERFORM command is issued.

The command string can be executed once or repeatedly. When the PERFORM command is entered after the CTRL C (^||C) the linked commands are executed once. However, when PERFORM is entered prior to and after the CTRL C (^||C), the first PERFORM command becomes a part of the command list and the linked command list is repeated until a second CTRL C (^||C) is entered.

If a linked command is entered incorrectly, the console outputs an error message when the command containing the syntactic error is executed. Typing a CTRL U (^||U) will reject the current command line.

3.12.2 Link Facility Usage

```
Syntax -- LINK <CR>

             COMMAND <CR>

             COMMAND <CR>

             <^||C>
```

Response -- Dependent on command list.

Example

It is desired to repeatedly examine the contents of a device register after a CPU initialize. The sequence of commands is as follows.

```
    >>> LINK                            The link mode is entered.

    <<< INITIALIZE                      An initialize command is
                                        entered.

    <<< EXAMINE/LONG 20002008           Examine command is entered
                                        into string (examining MBA
                                        status register).

    <<< PERFORM                         Perform is entered prior to
                                        linking termination.

    <<< ^C                              The linking command is
                                        terminated.

    >>> PERFORM                         Initiates execution.

    >>> P 20002008 00070000             Command string executed
                                        continuously.

    >>> P 20002008 00070000
```

3.13 CONSOLE I/O MODE ESCAPE

The console I/O mode escape sequence causes the console to move from console I/O mode to program I/O mode. To escape to program I/O enter

```
            >>> SET TERMINAL PROGRAM
```

Console commands START, CONTINUE, and NEXT also enable program I/O mode.

The program I/O mode escape sequence causes the console to change from program I/O mode to console I/O mode. To escape to console I/O mode enter

```
            ^||P
```

The CTRL P (^||P) is not recognized by the console if the console power switch on the console panel is in the REMOTE DISABLE or LOCAL DISABLE position.

3.14 REMOTE CONSOLE ACCESS COMMAND SET

A set of commands is included in the console command language for accessing the console from a remote terminal. The command set for remote access has the following:

1. Talk mode -- provides communication between local and remote terminal operators.

2. Copy control -- permits suppressing or enabling typeout on the local terminal while a remote operator is in control.

3. Transfer control -- permits transferring control of the console between the local and remote operators.

4. Echo control -- provides control of the echo of characters to the remote terminal while in talk mode.

5. Error message suppressor -- a method of suppressing lost carrier error messages caused by a loss of carrier on the remote line.

A printout of the remote console command help file is contained in Appendix L.

3.14.1 Enable Talk Mode Command

Syntax -- ENABLE <blank> TALK <CR>

Description -- The ENABLE TALK command places the console into talk mode. While in talk mode, characters typed on the remote keyboard are typed on the local terminal, and vice versa. The console does not echo characters back to the originating keyboard, unless the talk mode echo feature is enabled. Console commands are not recognized while in talk mode.

Talk mode is terminated by typing the escape character, CTRL P (^||P), on either the local or remote keyboard. When talk mode is terminated, console I/O mode is enabled. When the talk mode is entered, the console enables the remote serial interface and sends the data terminal ready signal to the data set. All terminal I/O to the CPU is disabled while in talk mode.

Response

```
>>> ENABLE TALK
.
.
.
^||P

>>>
```

3.14.2 ENABLE/DISABLE ECHO Command

Syntax -- (ENABLE ! DISABLE) <blank> ECHO <CR>

Description -- With the ENABLE ECHO command the console echoes characters input from either the remote or local keyboards while in talk mode. The DISABLE ECHO command suppresses the echo of characters typed from the local or remote keyboards.

ENABLE and DISABLE ECHO commands do not have any effect until talk mode is entered. A DISABLE ECHO is entered each time the console control panel rotary keyswitch is put in the LOCAL or LOCAL/DISABLE position, and upon power up of the console.

Response

```
>>> ENABLE ECHO
.
.
.
>>> DISABLE ECHO
```

3.14.3 ENABLE/DISABLE LOCAL COPY Command

Syntax -- (ENABLE ! DISABLE) <blank> LOCAL <blank> COPY <CR>

Description -- With the ENABLE LOCAL COPY command the local terminal prints a copy of all output directed to the remote terminal. The DISABLE LOCAL COPY command disables the local terminal from getting a copy of output directed to the remote terminal.

Local copy disable is entered each time the console keyswitch is turned to the LOCAL, REMOTE, or LOCAL/DISABLE position. Local copy enable is entered each time the console control panel rotary keyswitch is placed in the REMOTE/DISABLE position.

Response

```
>>> ENABLE LOCAL COPY

>>> DISABLE LOCAL COPY
```

3.14.4 ENABLE LOCAL CONTROL Command

Syntax -- ENABLE <blank> LOCAL <blank> CONTROL <CR>

Description -- An ENABLE LOCAL CONTROL command, entered at the remote terminal (when the console keyswitch is in the REMOTE position), transfers control of the console to the local terminal. This permits the local operator to take control of the console and the CPU, while the remote link is maintained. The remote operator can regain control of the console by typing CTRL P (^||P).

An ENABLE LOCAL CONTROL command entered from the local terminal has no effect. Local control is enabled when the console control panel rotary keyswitch is placed in the LOCAL or LOCAL/DISABLE position.

Response

```
            >>> ENABLE LOCAL CONTROL
            .
            .
            .
            ^||P

            >>>
```

3.14.5 ENABLE/DISABLE CARRIER ERROR Command

Syntax -- (ENABLE ! DISABLE <blank> CARRIER <blank> ERROR <CR>

Description -- With the ENABLE CARRIER ERROR command the message ?CARRIER LOST is printed each time a loss of carrier on the remote line is detected.

Response

```
>>> ENABLE CARRIER ERROR
.
.
.
?  CARRIER LOST

>>>
```

The DISABLE CARRIER ERROR command causes the console to inhibit the carrier lost message. An ENABLE CARRIER ERROR is automatically done whenever the console keyswitch is placed in the LOCAL or LOCAL/DISABLE position.

3.15 CONSOLE ERROR MESSAGES

All console error messages are prefixed by a question mark to distinguish them from informational messages. A description of the console error messages is contained in Appendix E. To obtain a set from your VAX-11/785 enter

```
>>>@ERROR.HLP
```

CHAPTER 4

DIAGNOSTIC SUPERVISOR

4.1 GENERAL

The VAX diagnostic supervisor controls the operation of most diagnostic programs used on VAX-11/785 systems. This chapter explains methods for booting the diagnostic supervisor, details the setup for running diagnostics, and includes instructions for disk formatting.

This chapter is intended as an introduction to the diagnostic supervisor. It is by no means an exhaustive treatment of the subject. For more detailed information on the diagnostic supervisor, see VAX Diagnostic System User's Guide (EK-VX11D-UG) and DS780 Diagnostic System Technical Description (EK-DS780-TD).

4.2 BOOTING THE DIAGNOSTIC SUPERVISOR

4.2.1 Environments

The diagnostic supervisor can run in two environments. In the first, user mode, the VAX/VMS operating system must be running. The other environment, standalone mode, requires that VMS not be running on the system. Whether the diagnostic supervisor is booted in user mode or standalone mode depends upon the diagnostic that will be run.

The environment necessary for a specific diagnostic is defined by the diagnostic's level. The diagnostic levels that require the diagnostic supervisor are 3, 2R, and 2. Table 4-1 lists the environment for each of these levels. The level of any diagnostic can be found in Section 2.1 of EVNDX.

Table 4-1 Diagnostic Supervisor Environments

LEVEL	ENVIRONMENT
3	Stand-alone
2R	User Mode
2	Either

4.2.2 Boot Procedures

4.2.2.1 Stand-alone Mode -

The diagnostic supervisor can be booted from either the system disk or the diagnostic supervisor diskette when it is to run in stand-alone mode.

Type the following to boot the diagnostic supervisor from the system disk.

```
    >>> B SRn

    where  S  designates the diagnostic supervisor
           R  designates the type of disk
           n  is the number of the disk drive
```

To boot the diagnostic from the diagnostic supervisor diskette, place diskette RX7 in the drive and type the following.

```
    >>> B
```

The DEFBOO.CMD file on diskette RX7 loads and starts the diagnostic supervisor.

4.2.2.2 User Mode -

The diagnostic supervisor is booted from the system disk when it is to run in user mode. To boot the diagnostic supervisor with VMS running, log into the field service account or SET DEFAULT to SYS$MAINTENANCE and run ESSAA.

```
    USERNAME: FIELD
    PASSWORD: SERVICE

    $RUN ESSAA

        or, from any account

    $SET DEFAULT SYS$MAINTENANCE
    $RUN ESSAA
```

Table 4-2 is a summary of methods for booting the diagnostic supervisor.

Table 4-2 Diagnostic Supervisor Boot Procedures

LOAD DEVICE	STAND-ALONE MODE	USER MODE
System disk	>>> B SRn . . . DS>	$SET DEFAULT SYS$MAINTENANCE $RUN ESSAA . . . DS>
Diskette	>>> B . . . DS>	not used

4.3 RUNNING DIAGNOSTICS

After the diagnostic supervisor has been booted, some setup work must be done before running a diagnostic. This involves the diagnostic supervisor ATTACH and SELECT commands.

4.3.1 ATTACHing A Device

The diagnostic supervisor keeps track of devices, drivers, system addresses, TR and BR interrupt priority levels by creating a parameter table from information received from the operator. This table is created when the operator issues ATTACH commands to the diagnostic supervisor.

The diagnostic supervisor can access a device only when it knows about each interface between the device and the CPU. Table 4-3 lists the commands used to attach the CPU, various interfaces, and several disk devices.

NOTE

> If a <CR> is typed after ATTACH, the operator will be prompted for the information needed. All information in Table 4-3 is given in this format. The alternative is to include everything on one line, with spaces separating each piece of information.

Table 4-3 ATTACH Commands

CPU

DS>	ATT
Device type?	KA785
Device link?	SBI
Device name?	KA0
G-floating instructions?	Yes
H-floating instructions?	Yes
WCS last address?	1FFF
Accelerator type?	0/1 (0=no accelerator)

CONTROLLERS

	UNIBUS (DW780)	MASSBUS (RH780)
DS>	ATT	ATT
Device type?	DW780	RH780
Device link?	HUB	HUB
Device name?	DWn	RHn
TR?	3	8
BR?	4	9

UNIBUS DISK ADAPTER
(UDA50)

DS>	ATT
Device type?	UDA50
Device link?	DW0
Device name?	DUA
UDAIP?	772150
Vector?	154
BR?	5
Burst_rate?	5

Table 4-3 ATTACH Commands (Cont)

	DEVICES				
	UNIBUS			MASSBUS	
	RA60	RA80	RA81	RM05	RP07
DS>	ATT	ATT	ATT	ATT	ATT
Device type?	RA60	RA80	RA81	RM05	RP07
Device link?	DUA	DUA	DUA	RH0	RH0
Device name?	DUA0	DUA0	DUA0	DRA1	DRA0

When attaching UNIBUS disks, make attachments in this order:

CPU
DW780
UDA50
disk

When attaching MASSBUS disks, make attachments in this order:

CPU
RH780
disk

4.3.2 SELECTing A Device

A device must be selected before it can be tested. For example, to run a CPU diagnostic, type at the DS> prompt:

```
    DS> SELECT KA0
    DS> SET TRACE     !Do this to see the titles of tests being run
    DS> RUN EVKA*
```

*	EVKAB	VAX Architectural Instr
	EVKAC	VAX Floating-Point Inst
	EVKAD	VAX Compatibility Mode Inst
	EVKAE	VAX Privileged Architectural Instr

The device name from the ATTACH command (see Table 4-3) is used as the name in the SELECT command.

4.3.3 Summary

The actions required to run a diagnostic are summarized in the following list.

1. Determine the diagnostic to be run.

2. Check in EVNDX for the diagnostic's level.

3. Boot the diagnostic supervisor. The boot procedure used depends on the diagnostic's level.

4. Attach the CPU.

5. Attach the device to be tested and its interfaces.

6. Select the device.

7. Run the diagnostic.

4.4 DISK FORMATTERS

Table 4-4 lists disk formatters and their levels.

Table 4-4 Disk Formatters

NAME	LEVEL	FORMATS THESE DISKS
EVRAC	2	RM05 and RP07
EVRLB	3	RA80, RA81, and RA60

To run a formatter, do the following.

1. Determine the correct environment (Table 4-4) and run the diagnostic supervisor.
2. Attach the device to be formatted and its interfaces.
3. Select the device.
4. Run the formatter.

NOTE

> Formatting will destroy data on the disk. Be sure that formatting is desired. DIGITAL maintenance personnel will format the disk, if the customer desires.

CHAPTER 5

PERIPHERAL DEVICES OPERATING INFORMATION

5.1 GENERAL

This chapter contains brief descriptions of the console and peripheral equipment typically used in VAX-11/785 systems. The description for each includes, as applicable, equipment specifications, operating controls and indicators, operating procedures, and customer care and maintenance.

NOTE

> Refer to Table 1-2 for a list of manuals containing more detailed information about the following equipment.

5.2 LA120 CONSOLE TERMINAL

The LA120 is a smart bidirectional 7x7 dot matrix impact printer. It contains a 1000-character input buffer and has a maximum print rate of 180 characters per second.

5.2.1 LA120 Specifications

Table 5-1 lists the specifications for the LA120.

Table 5-1 LA120 Specifications

Printer	
Printing technique	Impact dot matrix, smart bidirectional
Print matrix (width by height)	7 by 7
Maximum print speed	180 CPS
Horizontal slew speed	60 IPS
Single linefeed time	33 ms
Vertical slew speed	7.5 IPS
Paper feed	Pin-feed, tractor drive
Paper type	Fanfold, up to six parts (see paper requirements)
Forms length	1 to 168 lines
Vertical pitch (lines per inch)	2,3,4,6,8,12
Horizontal pitch (characters per inch)	
180 CPS	10,12,13.2,16.5
90 CPS	5,6,6.6,8.25
Maximum line length (varies with horizontal pitch)	
5 CPI	66 columns
6 CPI	79 columns
6.6 CPI	87 columns
8 CPI	108 columns
10 CPI	132 columns
12 CPI	158 columns
13.2 CPI	174 columns
16.5 CPI	217 columns
Margins	Left, right, top, bottom
Tabs	217 horizontal, 168 vertical, from keyboard or line
Forms storage	True nonvolatile memory (no batteries)
Positioning commands	Horizontal and vertical, absolute and relative
Character set	ASCII upper/lowercase set
National character sets	
Standard	United States United Kingdom
Optional	Finland Sweden Norway Denmark Germany France

Table 5-1 LA120 Specifications (Cont)

APL character set	Optional
Other printer features	Paper out and cover open interlocks, manual and automatic last character view, selectable auto new line, self-test, status message, 4-digit numeric display used as column counter and to set parameters, factory stored form setup (10 CPI, 6 LPI, 66 lines per form tab stops every eight columns, etc.)
Keyboard Specifications	
Keyboard	Typewriter style with multi-key rollover
Selectable auto LF	Standard
Optional numeric keypad	18 keys including 4 function keys
Feature selection	Keyboard entry to nonvolatile memory
Other keyboard features	Local form feed key, local line feed key, auto repeat on all alphanumeric keys, and selectable keyclick
Communication Specifications	
Data transfer	Serial, asynchronous
Baud rates (BPS)	50,75,110,134,134.5,150,300,600, 1200,1800,2400,4800,7200,9600.
Split speeds (BPS)	600 or 1200 receive, with 75 or 150 transmit; 2400 or 4800 receive, with 300 or 600 transmit.
Parity	Odd, even, or none (8th bit mark or space transmitted, or data bits only)
Input buffer	1024 characters standard, 4096 characters optional
Interface	Full EIA standard (includes auto answer/disconnect)
Physical Specifications	
Dimensions	
Width	69.9 cm (27.5 in)
Height	85.1 cm (33.5 in)
Depth	61.0 cm (24.0 in)
Weight	
Uncrated	46.4 kg (102 lb)
Shipping	63.7 kg (140 lb)

Table 5-1 LA120 Specifications (Cont)

Physical Specifications (Cont)	
Power	
Transformer power supply	
Voltage	87 to 128 V
Frequency	60 Hz + 1 Hz
Switcher power supply	
Voltage	90–128 V or 180–256 V
Frequency	47–63 Mhz
Input current	4.2 A max. at 115 V
Heat dissipation – printing	440 W max.
Temperature	
Operating	10° to 40°C (50° to 104°F)
Nonoperating	–40° to 66°C (–40° to 151°F)
Relative Humidity	
Operating	10 to 90 percent with a maximum wet bulb temperature of 28°C (82°F) and a minimum dewpoint of 2°C (36°F), noncondensing
Nonoperating	5 to 95 percent, noncondensing

Paper Requirements	
General	Continuous, fanfold, pin-feed forms
Width	7.6 to 37.8 cm (3 to 14-7/8 in)
Hole spacing	12.7 mm ± 0.25 mm (0.500 in ± 0.010 in) non-accumulative over 5 cm (2in)
Hole diameter	3.81 to 4.06 mm (0.150 to 0.160 in)
Forms thickness	
Single part	15 lb paper minimum, 0.25 mm (0.010 in) card stock maximum
Multipart	Up to 6 parts (see notes), 0.50 mm (0.020 in) maximum

5.2.2 LA120 Controls And Indicators

The LA120 keyboard and control panel contains a four-digit numeric display, several control keys that do not transmit printing characters, and seven indicator LEDs. See Figure 5-1.

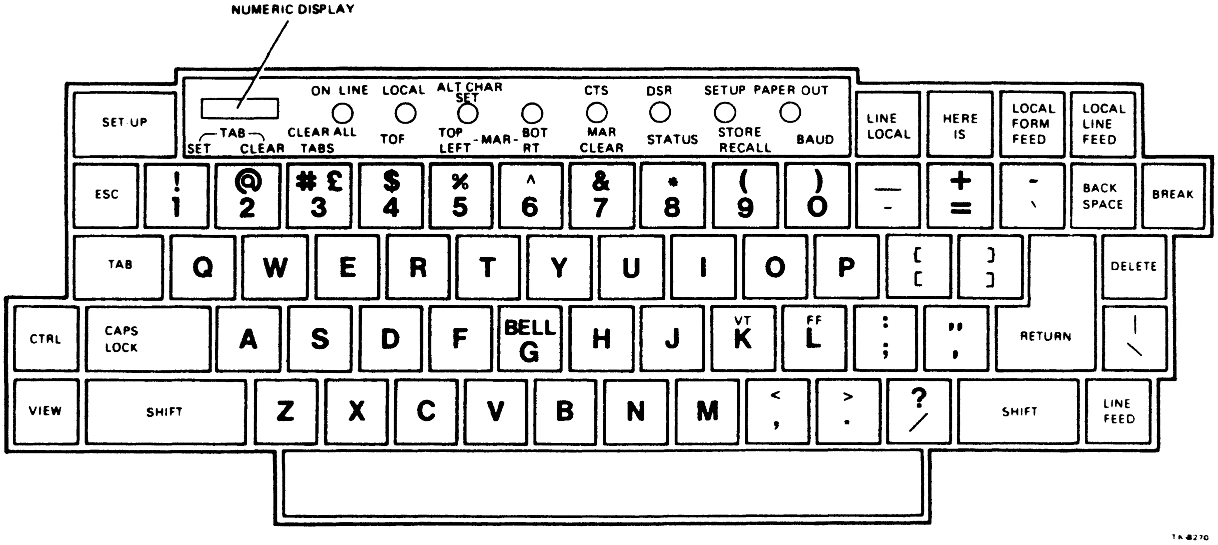


Figure 5-1 LA120 Keyboard

Table 5-2 LA120 Controls

Key	Function
SHIFT	When used with a character producing key, the key's uppercase character is transmitted. (The lowercase character is normally transmitted.)
CTRL	When used with a character producing key, the key's control character is transmitted. Use of the CTRL key is generally expressed as follows. CTRL/K = Press and hold the "CTRL" key and press the letter "K" key on the keyboard.
CAPS LOCK	When set, causes the alphabet keys to transmit uppercase characters regardless of the SHIFT key. Does not affect the number or symbol keys.
LINE/LOCAL	Changes the LA120 to the ON LINE mode or LOCAL mode as displayed by the indicators.
LOCAL LINE FEED	Advances the paper one line at a time.
LOCAL FORM FEED	Advances the paper to the top-of-form position, or to the top margin (if the top margin is set).
VIEW	Allows the operator to view the last character printed if the last-character-view feature is disabled.

Table 5-3 LA120 Indicators

Indicator	Function
ON LINE	The LA120 is on-line. The terminal transmits keyboard characters to the CPU and receives printer characters. Characters typed on the keyboard are transmitted back to the printer by the program.
LOCAL	The LA120 keyboard is locally connected to the printer. The terminal is off-line and does not receive or transmit characters.
ALT CHAR SET	The alternate character set is enabled (in terminals optionally equipped with a foreign language character set).
CTS *	Data transmission is enabled (clear to send).
DSR *	The modem is in data mode (data set ready).
SETUP	Blinks when the LA120 is in the SETUP mode.
PAPER OUT	Blinks when the LA120 is disabled by one of the following conditions. • The printer is out of paper. • The cover is open. • The print head is jammed.
NUMERIC DISPLAY	During normal operation, indicates the horizontal column position of the print head. In SETUP mode, the vertical line position is also indicated by pressing the SHIFT key.

* Used in terminals connected to a modem.

5.2.2.1 SETUP Functions -

Place the terminal in LOCAL mode to avoid transmitting characters while making changes in SETUP mode.

In SETUP mode, the numeric display shows the current column number. When the SHIFT key is pressed, the display shows the current vertical line number.

The following procedure describes the two methods of entering and exiting SETUP mode.

1. To enter SETUP mode, press and hold the CTRL key and press the SETUP key. Release both keys. The SETUP light flashes, indicating that the terminal is in SETUP mode.

   Make the desired changes.

   To exit SETUP mode, press the SETUP key and release it. The SETUP light stops flashing.

2. To enter SETUP mode, press and hold the SETUP key. The SETUP light flashes.

   Make the desired changes.

   Release the SETUP key to exit SETUP mode. The SETUP light stops flashing.

Most of the keys on the console keyboard have SETUP functions. The functions for the top row of keys and some other commonly used features are briefly described in Table 5-4.

Table 5-4 LA120 SETUP Functions

Key	Name	Function
1	TAB SET	Sets horizontal tab at the current column. When used with SHIFT, sets vertical tab on the current line.
2	TAB CLEAR	Clears horizontal tab at the current column. When used with SHIFT, clears vertical tab on the current line.
3	CLEAR ALL	Clears all tabs, horizontal and vertical.
4	TOF	Resets the vertical line counter to line 1. Top-of-form (TOF) defines the first line after the perforation when fanfold paper or forms are loaded. If the top margin is set, the paper moves to the top margin (the first printable line) when the form feed command is sent by the computer or LOCAL FORM FEED is pressed. To avoid having to set top-of-form each time the printer is turned on or the paper is changed, press LOCAL FORM FEED before turning off the system or the printer.
5	TOP/LEFT	Sets the left margin at the current column. When used with SHIFT, sets the top margin on the current line.

Table 5-4 LA120 SETUP Functions (Cont)

Key	Name	Function
6	BOT/RT MAR	Sets right margin at the current column. When used with SHIFT, sets bottom margin on the current line.
7	MAR CLEAR	Clears left and right margins. When used with SHIFT, clears top and bottom margins.
8	STATUS	Prints a message that shows the state of the currently selected SETUP features. (See the *LA120 User Guide* for a description of the letter key functions given by the printout.)
9	STORE/RECALL	Cancels all temporary SETUP features and recalls the permanently stored features. When used with SHIFT, permanently stores the currently selected SETUP features.
0	BAUD	Press once to view the current transmit and receive baud rate in the numeric display. Pressing the key again changes the baud rate until the desired rate is indicated in the numeric display. This value should not be changed because the CPU responds only to a fixed rate. The factory setting is for 300 baud. The CPU is switch selectable to 1200, 2400, or 9600 baud but should not be changed except by authorized service personnel.
G	Volume	Press once to view the current setting in the numeric display. Press again to set the desired bell and keyclick volume as indicated in the numeric display. 1 = High volume 2 = Low volume
K	Keyclick	Press once to view the current setting in the numeric display. Press again to enable or disable the keyclick feature as indicated in the numeric display. 1 = Enabled 0 = Disabled
Z	View Last Character	Press once to view the current setting in the numeric display. Press again to enable or disable the last character view feature as indicated in the numeric display. 1 = Enabled 0 = Disabled

5.2.3 LA120 Operating Instructions

5.2.3.1 Loading Paper Or Forms -

The LA120 accepts sprocket-fed, multiple-part paper or forms ranging in width from 3 to 14-7/8 inches. The general procedure for loading paper or forms is as follows.

1. Turn the printer off, then open the cover.

2. Loosen the left and right tractor adjusting screws.

3. Position the left tractor feed pins approximately two inches from the left sideplate and tighten the left tractor adjusting screw. This provides optimum margins for 132-column paper. It may be necessary to readjust the left pins when using printed forms.

4. Open the printer cover. Open both tractor covers and move the print head adjustment lever away from the paper (Figure 5-2).

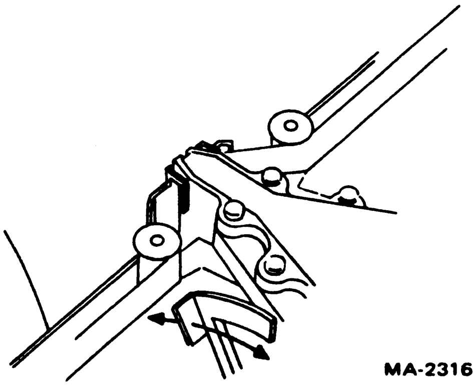


Figure 5-2 LA120 Print Head Adjustment

5. Place the paper on the floor between the legs of the LA120 (Figure 5-3) and align the leading edge of the paper so it is parallel to the paper slot. Align the left sprocket holes of the paper with the left tractor.

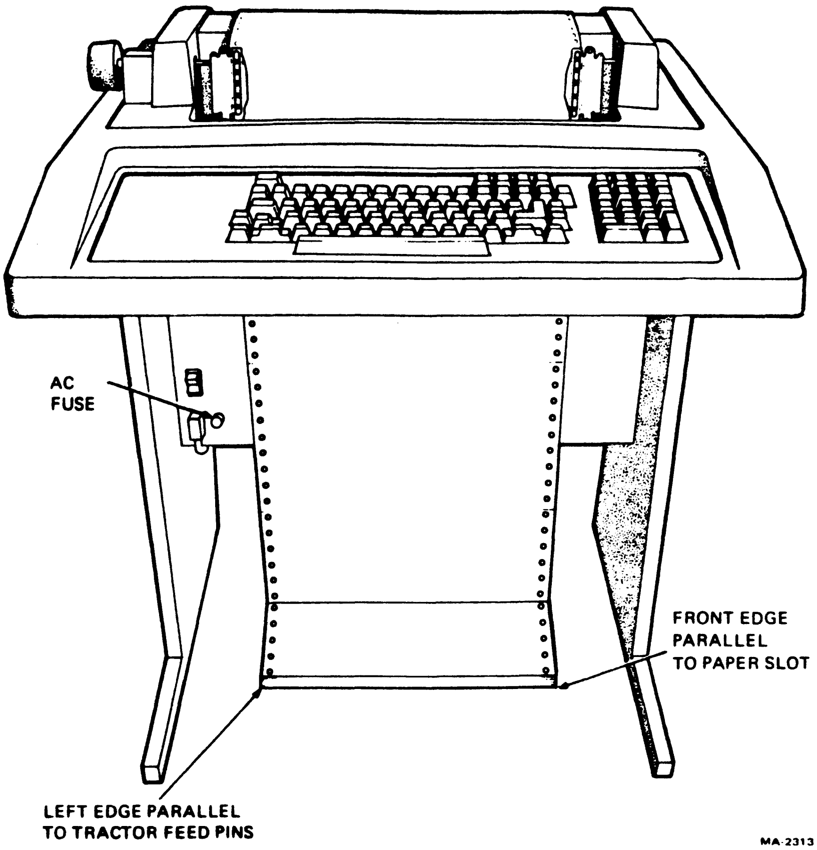


Figure 5-3 LA120 Paper Alignment

6. Feed the paper up through the paper slot. Align the left sprocket holes over the feed pins. Close the left tractor cover (Figure 5-4).

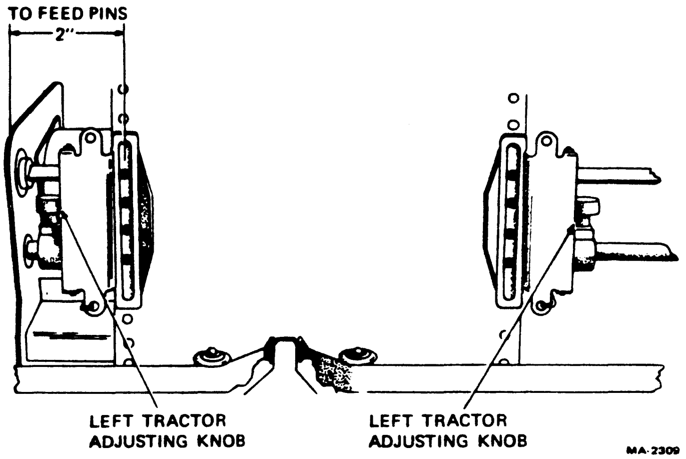


Figure 5-4 LA120 Tractor Adjustment

7. Making sure that the paper is straight, align the right sprocket holes over the feed pins and close the right tractor cover. If the paper pulls against the pins or bows in the middle, loosen the right tractor adjusting knob and adjust the right tractor.

CAUTION

Do not overtighten the tractor adjusting knobs.

8. Reset the print head adjustment lever for the thickness of the paper or forms.

9. Set top-of-form (Figure 5-5). Press and hold the SETUP key and press the number 4 (TOF) key to reset the line counter to 1. Using the paper adjust knob, set the paper perforation one character height above the line indicator on the print head. The paper adjust knob is pushed in to make fine vertical adjustments.

NOTE

To avoid having to set top-of-form each time the printer is turned on or the paper is changed, press LOCAL FORM FEED before turning off the system or printer or opening the cover.

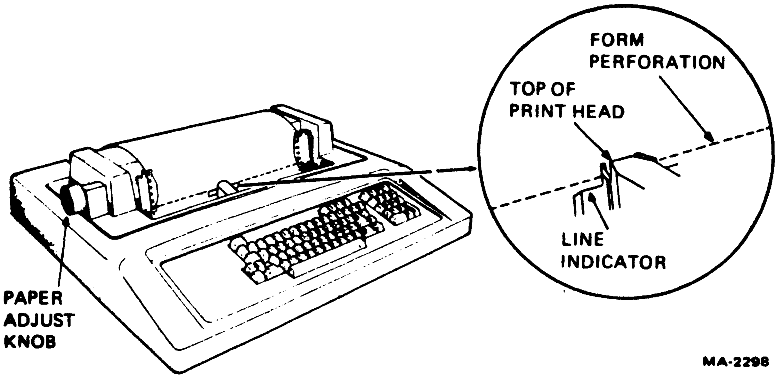


Figure 5-5 Setting Top-of-Form on LA120

NOTES

1. Multipart forms with 3- or 4-prong margin crimps on both margins are recommended.

2. Stapled forms are not recommended and may cause damage to the print head, tractors, or other parts of the machine, voiding the warranty.

3. Multipart forms may have only one card part. The card must be the last part.

4. Multipart carbonless forms of up to six parts may be used. Ribbon must be used on the top copy; first-surface impact paper is not recommended.

5. Dot or line glue margins are acceptable if the glue is on one margin only. Line glue on both margins prevents air from escaping, resulting in poor impressions.

6. Split forms with each side containing a different thickness or number of sheets are not recommended.

5.2.4 LA120 Customer Care And Maintenance

5.2.4.1 Power Switch, Fuse, And Cover Interlock -

The console terminal is plugged into a switched ac outlet of the system power control. The normal position for the ac power switch is ON to allow the terminal to turn on and off with the system. A safety interlock is provided to prevent printer operation when the cover is opened. The ac power switch and line fuse are located in the lower left-hand corner of the front panel below the keyboard. The line fuse is a 250-volt slo-blo, rated at 3 amps.

5.2.4.2 Ribbon Installation -

The LA120 ribbon provides from six to eight hours of continuous printing. When the print contrast becomes too light, the ribbon may be turned over for one or two more hours of printing. It should then be replaced as follows.

1. Turn the printer off and open the top cover.

2. Move the carriage adjustment lever away from the paper (Figure 5-6).

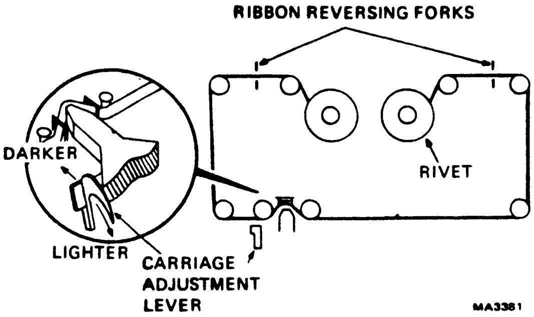


Figure 5-6 LA120 Ribbon Installation

3. Wind the old ribbon on one spool and remove it.

4. Save the empty spool and discard the old ribbon.

5. Attach the end hook of the new ribbon to the empty spool.

6. Wind ten turns of ribbon on the empty spool, making sure that the ribbon rivet is on the empty spool.

7. Place the full spool on the left spool shaft (which turns freely) and turn it clockwise until it drops into position.

8. Thread the new ribbon around the print head and guides and through the ribbon reversing forks (Figure 5-6).

9. Check that the ribbon rivet is on the empty spool or is between the empty spool and the ribbon reversing fork.

5.2.4.3 Print Impression Adjustment -

1. Open the printer top cover.

2. Move the carriage adjustment (Figure 5-6) lever toward the paper and make light contact with the paper or form.

3. Manually move the print head to the side to check for smudging or paper rippling.

4. Close the cover and type several characters.

5. If smudging or rippling occurs, open the cover and move the head adjustment lever slightly away from the paper and check it again. Repeat steps 3 and 4.

CAUTION

Only DIGITAL recommended ribbons (part no. 36-12153-01) should be used in the LA120. Using ribbons other than that specified may damage the print head and will void the warranty.

5.2.4.4 Error Display -

When the LA120 is turned on, it runs several internal tests. Any error results are viewed in the numeric display as shown in Table 5-5. If the error cannot be cleared, an authorized service engineer should be notified.

Table 5-5 LA120 Error Display

Display Indicates	**Causes**	**Corrective Action**
0 (flashing)	Error at ROM address 0	Call for service
1 (flashing)	Error at ROM address 2048	Call for service
2 (flashing)	Error at ROM address 4096	Call for service
3 (flashing)	Error at ROM address 6144	Call for service
4 (flashing)	Error at ROM address 8192	Call for service
5 (flashing)	Error at ROM address 10240	Call for service
6 (flashing)	Reserved for future options	
7 (flashing)	RAM diagnostic failure	Call for service
8 (flashing)	Microprocessor failure	See note 1
9 (flashing)	Nonvolatile memory failure	See notes
8888 (constant)	Cover open, or paper out indication	Close cover, Reload paper

NOTES

1. *Turn LA120 off then back on. If an error indication reappears, record indication and call for service.*
2. *If the original problem was a flashing 9, check the stored SET-UP feature to ensure that it has not been affected. The self-test is an additional test (Part 2, operator's chapter) which can be initiated by the operator. The test will help determine if the problem is in the printer or in some other portion of the communication system.*

If you are unable to turn the printer on or if the printer appears to be faulty, refer to the operator's troubleshooting table. This table describes those things an operator can check prior to requesting service.

5.2.4.5 Operator Troubleshooting -

Common problem symptoms, their possible causes, and corrective actions are listed in Table 5-6.

Table 5-6 LA120 Operator Troubleshooting

Symptom	Possible Cause and Corrective Action
LA120 does not turn on when printer power switch set to ON	AC power cord is not plugged into wall outlet or front of printer. Plug in this cord. Power is not coming from the wall outlet. Check outlet with a known working electrical device (such as a lamp). If no power, call your electrician. AC line fuse blown; turn printer off and have the fuse replaced. (See Part 3 for fuse location.)
Characters do not print	Printer out of paper; load paper. (See Part 3 for paper loading.) Printer cover open or ajar. Close cover. Print head too far from paper; readjust print head adjustment lever. (See Part 3 for adjustment.) Data set unplugged; plug it in. Incorrect communication setup.
Light print	Print head too far from paper; adjust print head adjustment lever. Ribbon out of ink; turn ribbon over or replace ribbon. (See Part 3 for ribbon replacement.)
Paper does not advance	Paper not loaded properly; check that the tractor covers are closed and the feed holes are properly aligned. Feed holes torn; reload paper. If paper pulls against the tractor pins or bows in the middle, readjust the right tractor.

NOTE

Turn the ribbon over after 5 to 6 hours of continuous printing. Ribbon can be turned over only once; then it must be replaced.

Table 5-6 LA120 Operator Troubleshooting (Cont)

Symptom	Possible Cause and Corrective Action
Paper tearing on multipart forms	Print head exerting too much pressure on the paper; readjust the print head adjustment lever. Tractor incorrectly adjusted. If the paper pulls against the tractor pins or bows in the middle, readjust the right tractor. Paper not straight in printer; re-align paper.
Print head jam or print head does not move	Paper or print head jam; clear jam and perform reloading paper/form procedure in section 3 of this chapter.
No keyboard or printer	Printer cover open or ajar when printer is turned on (normally indicated by flashing 8888 and PAPER OUT light); close cover.
Garbled or double characters.	Incorrect communication setup. Ensure that your communication setup is compatible with the equipment at the other end of the line.

5.3 RA60 DISK DRIVE

The RA60 disk drive is a removable media drive with a storage capacity of 205 million bytes of 16-bit word data or 208 million bytes of 18-bit word data. The drive operates on the standard disk interface (SI).

The RA60 is sold with slides and is rack mountable. Up to three disk drives may be mounted into each cabinet.

5.3.1 RA60 Specifications

Table 5-7 RA60 Performance Specifications

Characteristics	Specifications
Data storage capacity	
in 16 bit word format	205 megabytes
in 18 bit word format	208 megabytes
Head switch latency	
Minimum	10 ms
Maximum	14 ms
Seek latency	
One cylinder seek latency	6.67 ms
Maximum seek latency	70.00 ms
Average seek latency	41.67 ms
Rotational latency	
Disk speed	3600 RPM ± 5%
Maximum latency	16.66 ms
Average latency	8.33 ms
Maximum start time	35 sec
Data transfer rate	
Bit rate	15.84 megabits/sec ± 5%
Bit cell period	63.13 nsec ± 5%
Unrecoverable data error rate	1 in 10**12 bits
Recoverable data error rate	1 in 10**7 bits
Corrected data error rate	1 in 10**11 bits
Recoverable seek error rate	1 misseek in 10**6

Table 5-8 RA60 Physical Specifications

Characteristics	Specifications (RA60 Drive Only)
Height	26.52 centimeters 10.44 inches
Width	48.26 centimeters 19 inches
Depth	85.09 centimeters 33.75 inches
Weight (drive only)	68.95 kilograms 152 pounds
Height	106.30 centimeters 41.80 inches
Width	54.20 centimeters 21.30 inches
Depth	14.20 centimeters 36.00 inches
Weight (cabinet only)	90.70 kilograms 200 pounds
Shipping weight one drive in cabinet	176.42 kilograms 389 pounds
Minimum door entrance width required for shipping crate	76.2 centimeters 30 inches
Minimum service clearance required	
Front	91.4 centimeters 36 inches
Rear	91.4 centimeters 36 inches
Left side	No side access needed
Right side	No side access needed

Table 5-9 RA60 Environmental Limits Specifications

Characteristic	Specification
Operating ranges:	
Temperature	16°C (50°F) to 40°C (104°F)
Temperature rate of change	20°C/hour (36°F/hour)
Relative humidity	8% to 80% with a maximum wet bulb temperature of 26°C (78°F) and a minimum dew point of 2°C (36°F)
Altitude	2438 meters 8000 feet
Air quality requirement	Less than 5 million particles of 0.5 micron per cubic foot of air
Nonoperating ranges: shipping and storage	
Temperature	–40°C (–40°F) to +66°C (+150°F)
Relative humidity	8% to 80% with no condensation
Altitude	9175 meters 30,000 feet

Table 5-10 RA60 AC Power Specifications

Characteristic	Specification
Voltage/Current	
RA60-CA/CE	Nominal: 120 volts ac/7.5 amperes
RA60-CD/CJ	Nominal: 240 volts ac/3.75 amperes
Voltage range/Frequency range	
RA60-CA/CE	90-128 volts ac/60 ± 1 hz
RA60-CD/CJ	180-256 volts ac/50 ± 1 hz
Input power	
RA60-CA/CE	600 watts typical/800 watts worst case
RA60-CD/CJ	600 watts typical/800 watts worst case
Power factor	Greater than 0.75
Maximum start current	190 amps RMS at 120 VAC for ½ ac cycle 95 amps RMS at 240 VAC for ½ ac cycle

5.3.2 RA60 Controls And Indicators

5.3.2.1 Operator Control Panel -

The RA60 operator control panel is located on the front of the disk drive. See Figure 5-7.

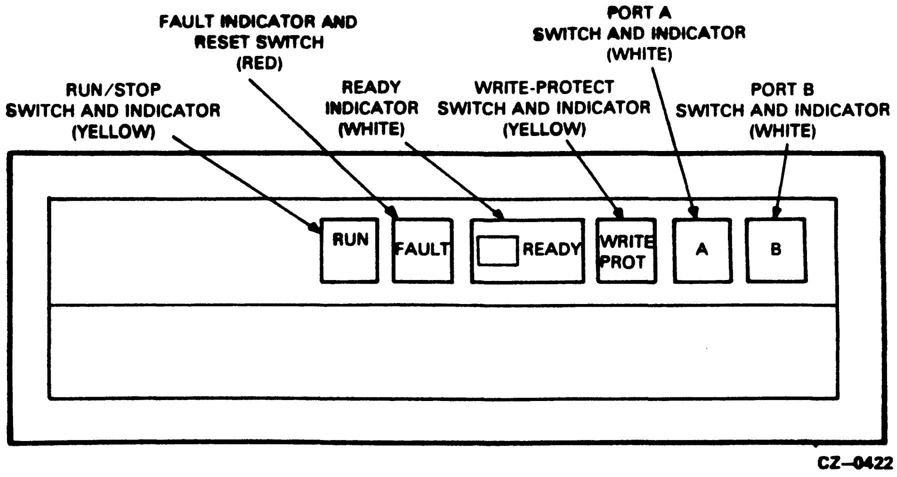

FAULT INDICATOR AND RESET SWITCH (RED)
PORT A SWITCH AND INDICATOR (WHITE)
RUN/STOP SWITCH AND INDICATOR (YELLOW)
READY INDICATOR (WHITE)
WRITE-PROTECT SWITCH AND INDICATOR (YELLOW)
PORT B SWITCH AND INDICATOR (WHITE)
RUN
FAULT
READY
WRITE PROT
A
B
CZ-0422

Figure 5-7 RA60 Front Panel Controls and Indicators

RUN/STOP Switch and Indicator

The RUN/STOP switch is a double-action push button. When placed in the RUN position (button in), the switch requests that the disk spin up and load the heads. When placed in the STOP position (button out), the heads are unloaded off the disk and the spindle motor is stopped. The RUN/STOP light always reflects the physical state of the disk spindle. When the light is on, it indicates that the disk spindle is spinning. When the light is off, it indicates that the disk spindle is stopped.

If the drive is on line and operating when the switch is placed in the STOP position, the drive will complete all outstanding write operations commanded by the controller before spinning down.

If the RUN/STOP switch is in the IN position when power is applied to the drive, it must be reset by releasing it and pushing it in again to start the spin-up sequence.

FAULT Switch and Indicator

The FAULT switch and indicator performs a dual function. When the FAULT light is off, the FAULT switch can be pushed to perform a lamp test. When the FAULT light is on, it indicates that an error condition exists within the RA60.

The FAULT button is a momentary contact switch. When pushed the first time, the switch causes the drive to go off-line. In this state, the drive assumes control of all front panel indicator lights which blink to display a binary code indicating the type of fault. Table 5-11 shows the hex fault codes that can be displayed by the front panel indicators.

The blinking fault code will remain displayed until an attempt is made to clear the fault by pushing the FAULT switch a second time. If successful, the FAULT light will go off and the drive will return to the available state. If unsuccessful, the FAULT light will remain on until the drive is repaired.

Table 5-11 RA60 Fault Identification Codes

Description of Error	RUN STOP	FAULT	READY	WRT PROT	A	B	HEX Code
Microcode error (not sent to front panel)	*	off	off	off	off	off	00
Heads home switch fault	*	off	off	off	off	on	01
Front panel fault	*	off	off	off	on	off	02
Long spin-up time fault	*	off	off	off	on	on	03
Bad servo samples fault	*	off	off	on	off	on	05
Bad velocity fault	*	off	off	on	on	off	06
Linear mode fault	*	off	off	on	on	on	07
Retry on seek fault	*	off	on	off	off	off	08
Lost servo samples fault	*	off	on	off	off	on	09
Heads home switch would not open	*	off	on	off	on	off	0A
Master processor fault	*	on	off	off	off	off	10
SDI fault	*	on	off	off	off	on	11
Invalid SDI level 1 command	*	on	off	off	on	on	13
Master processor fault	on	on	on	on	on	on	3F

* The run/stop switch may be on or off depending on the state of the drive.

UNIT/READY Plug and Indicator

The plastic cap labeled READY is both an indicator cap and a removable unit address plug. The READY light will come on after the RUN/STOP switch is pushed in and the drive has completed a successful spin-up sequence. The drive is ready to accept read and write commands whenever the READY light is on. The READY light will go off when the drive performs seek operations.

The plastic READY cap is cam-coded to distinguish one drive from another in multi-drive arrangements. This unit address plug determines the logical number that this drive presents to the host. Only numbers between 0 and 251 may be programmed into this unit address plug to satisfy the requirements of the Mass Storage Control Protocol.

WRITE PROT Switch and Indicator

The WRITE PROT switch is a double action push button with a built-in indicator light. Push in the WRITE PROT switch to place the drive in write protect mode and turn on the WRITE PROT light. In this mode, the write circuits of the disk drive are disabled.

Release the WRITE PROT switch (OUT position) to enable the write circuits and turn off the WRITE PROT light.

A and B Port Select Switches

The RA60 has two port select switches. These double-action switches (A and B) have built-in lights that indicate which drive port is presently on line to the controller. See Table 5-12.

Table 5-12 RA60 A and B Port Select Switches

PORT SELECT SWITCH		
A	B	PORT AVAILABILITY
IN	IN	Dual-port programmable
IN	OUT	A
OUT	IN	B
OUT	OUT	Neither port available

5.3.3 RA60 Operating Instructions

5.3.3.1 Power On And Power Off -

Power to the disk drive is controlled by two circuit breakers. One circuit breaker is on the power controller at the base of the cabinet, and it controls power to all drives in that cabinet. The other circuit breaker is on the back of the disk drive. These two circuit breakers should normally be left in the ON position after drive installation. If both of these circuit breakers are on, the drive will spin up whenever the RUN/STOP switch is pushed. See Figure 5-8.

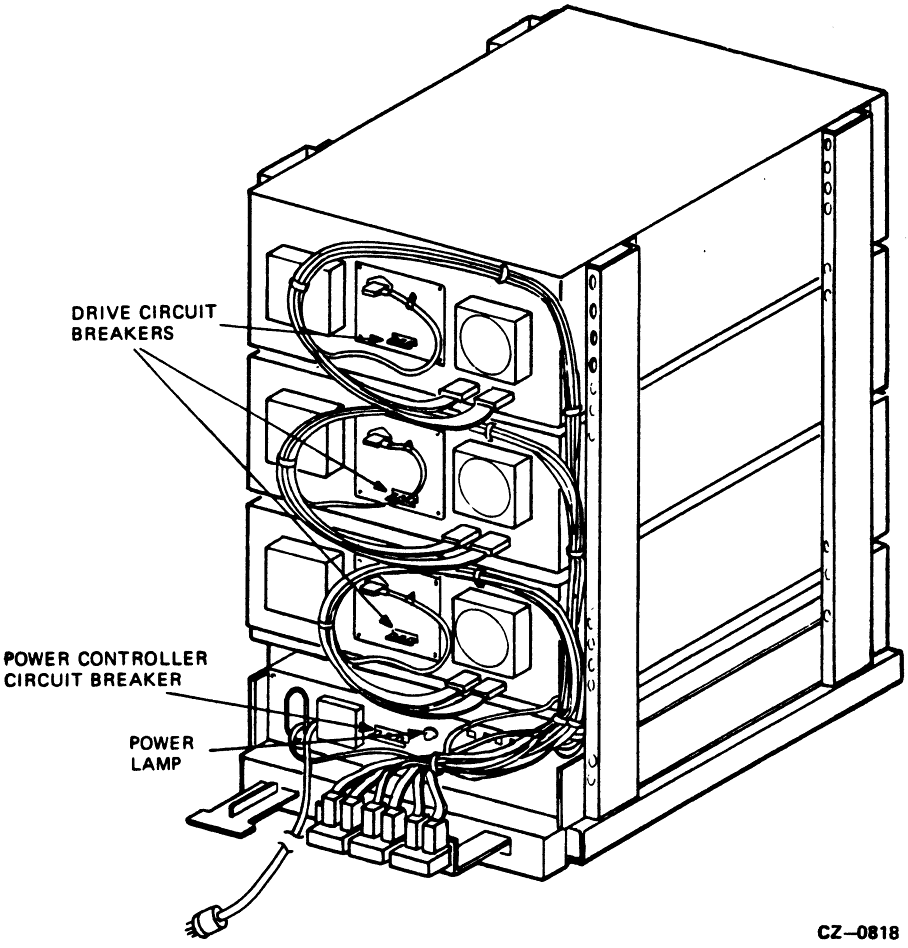


Figure 5-8 RA60 Drive Circuit Breakers

5.3.3.2 Disk Pack Loading And Unloading -

The RA60 must be loaded with an RA60 disk pack and then be brought to the ready state before information can be stored or read. The procedures for loading and unloading the disk pack are as follows.

Loading the Disk Pack

1. Apply power by turning on the drive circuit breaker located at the back of the drive.

2. Check that the RUN light is off (RUN/STOP switch is not pushed in).

3. Push the release button to raise the front cover. Drive power must be on to operate the cover release button.

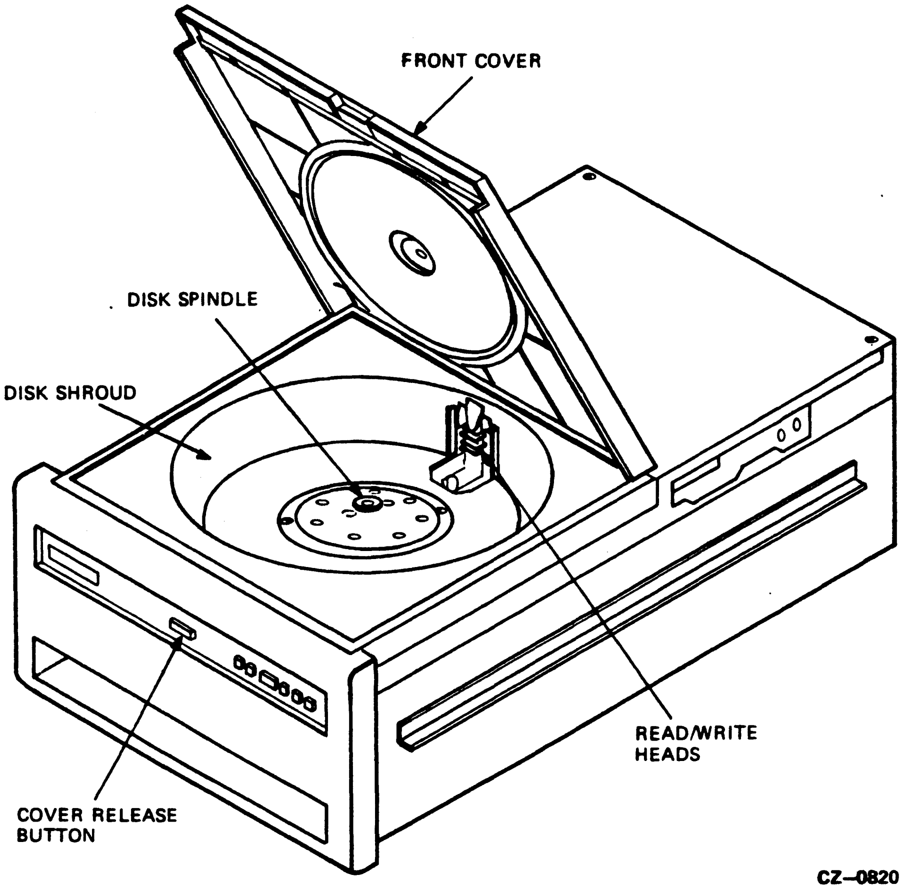


Figure 5-9 Gaining Access to the RA60 Disk Spindle

4. Look into the disk well to make sure that the read/write heads are retracted from the disk shroud area. Failure to do this could cause damage to the heads or the disk pack or both.

5. Remove the bottom dust cover from the disk pack canister by squeezing together the two slide buttons on its underside. Avoid placing fingerprints on the exposed disks inside the canister.

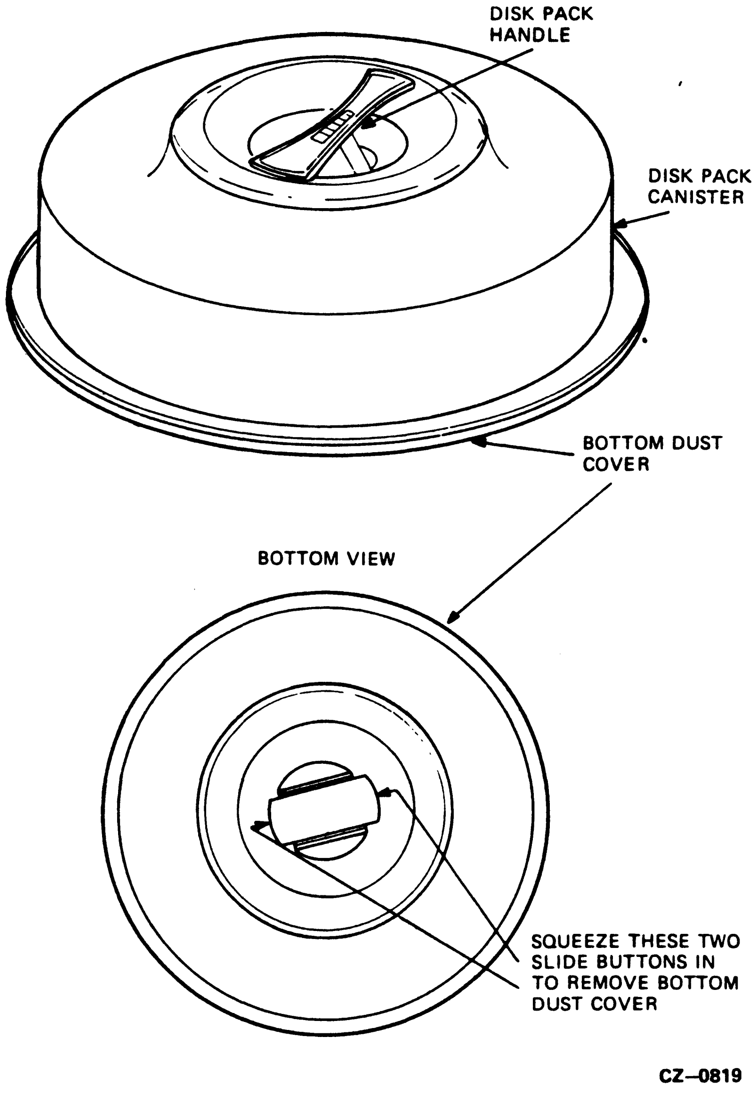


Figure 5-10 RA60 Disk Pack Storage Canister

6. Lifting the canister by its handle, set the pack onto the disk spindle.

7. Using one hand to turn the handle while holding the disk pack cover stationary with the other, rotate the pack handle clockwise until it stops turning. Give one last, firm twist to the handle and lift the storage canister off the pack.

8. Close the front cover of the disk drive.

9. Push in the RUN/STOP switch and wait for the READY light to come on.

Unloading the Disk Pack

1. Release the RUN/STOP switch and wait for the RUN light to go off.

2. Push in the release button and raise the front cover.

3. Place the storage canister over the disk pack so that the post protruding from the center of the disk pack contacts the canister handle.

4. Using one hand to turn the handle and the other to hold the disk pack stationary, rotate the canister handle counterclockwise until the pack is free of the spindle.

5. Using the canister handle, lift the pack out of the disk drive and close the front cover.

6. Install the bottom dust cover on the base of the storage canister by pressing it on gently until it locks into place.

5.4 RA80 DISK DRIVE

The RA80 is a rack-mountable disk drive using non-removable media. It has a data storage capacity of 121 megabytes. Each RA80 disk drive offers dual-port capability. The RA80 disk drive operates on the standard disk interface (SI) bus, which connects the drive to the controller. As many as four drives can be connected to an SI controller. The RA80 disk drive is shown in Figure 5-11.

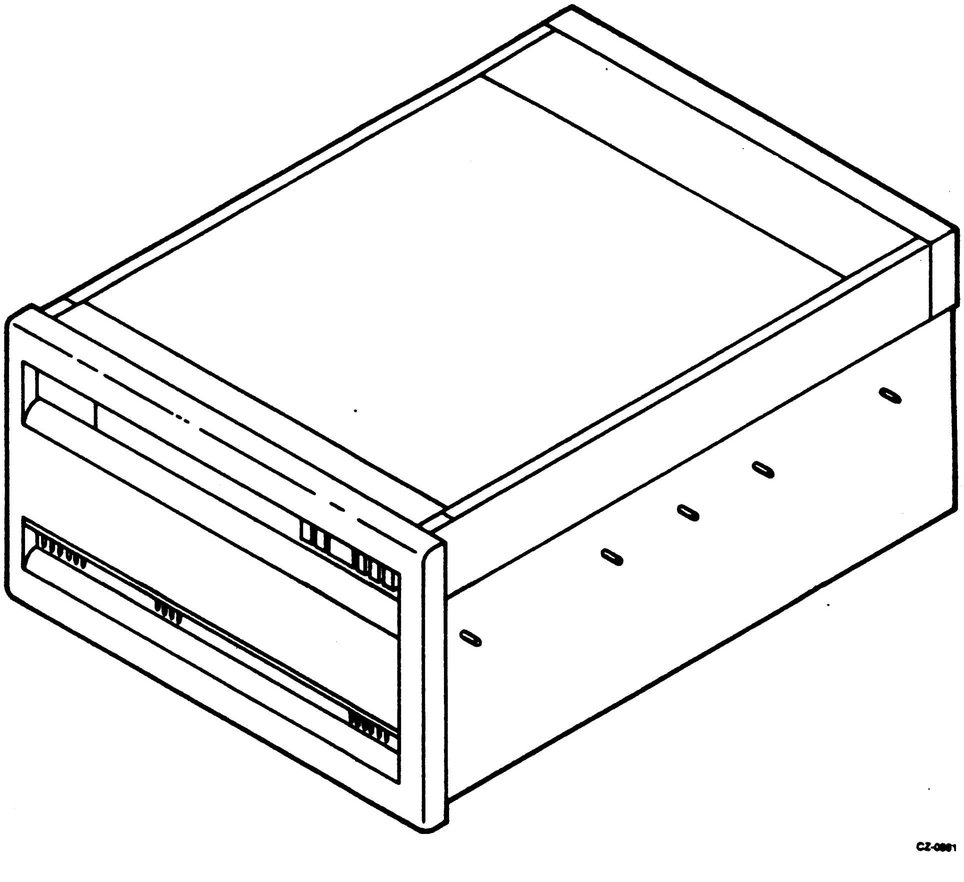


Figure 5-11 RA80 Disk Drive

5.4.1 RA80 Specifications

Table 5-13 RA80 Specifications

Characteristic	Specification
Data Storage Capacity (Single drive)	
HDA word format	16-bit
Storage capacity	121 MB
Bits/inch (inner track)	6339
Tracks/inch	478
Head Disk Assembly (HDA)	
Disk recording method	Modified frequency modulation (MFM)
Number of disks	4
Disk surfaces	7 data, 1 servo
Number of heads	14 data, 1 servo
Heads per disk surface	2
Number of tracks	561 per head
Logical cylinders	546 per head
Number of tracks used for revector and format control tables	14 per head
Number of sectors	31 (0 through 30)
Number of logical blocks	236,964 LBNs
Maximum Seek Times	
All cylinders	50 ms
One cylinder	6 ms
Average seek	25 ms
Latency	
Speed	3600 r/min + 4.3% - 3.5%
Maximum latency	16.6 ms
Average latency	8.33 ms
Single Start/Stop Time	
Start (maximum)	40 seconds
Start (typical)	15 seconds
Inhibit between stop and restart	3 minutes

Table 5-13 RA80 Specifications (Cont)

Characteristic	**Specification**
Data Rates	
Bit cell time	103.3 ns
Bit rate	9.677 MHz
Word rate (16-bit)	1.65 microseconds
RA80 Environmental Limits	
Temperature	
Operating	10° C to 38° C (50° F to 101° F) with a temperature gradient of 20° C/hour (36° F/hour)
Non-operating	-40° C to +60° C (-40° F to +140° F)
Relative humidity	
Operating	10% to 90% (non-condensing) with a maximum wet bulb temperature of 28° C (82° F) and a minimum dew point of 2° C (36° F)
Shipping and storage	10% to 90% with no condensation
Heat dissipation	2460 Btu/hour
Altitude	
Operating	Sea level to 2,400 meters (8,000 feet) Note: Maximum allowable operating temperatures are reduced by a factor of 1.8° C/1000 meters (1° F/1000 feet) for operation above sea level.
Non-operating and shipping	300 meters (1000 feet) below sea level to 3,600 meters (12,000 feet) above sea level (actual or effective by means of cabin pressurization)

Table 5-13 RA80 Specifications (Cont)

Characteristic	Specification
Physical Characteristics	
Drive	
Height	26 cm (10.38 inches)
Width	44 cm (17.50 inches) 48 cm (18.88 inches) including bezel
Depth	67 cm (26.50 inches) 71 cm (28.12 inches) including bezel
Weight	66.7 Kg (147 pounds) approximately
Electrical	
Voltages available (single-phase)	120 Volts ac + 8, -21; 60 Hz ± 1 Hz
	220-240 Volts ac + 16, -36; 50 Hz ± 1 Hz
Input Power (maximum)	
120 Volts ac, 60 Hz	720 Watts
220-240 Volts ac, 50 Hz	720 Watts
Starting current for:	
120 Volts ac, 60 Hz	31 A (peak surge for five seconds)
220-240 Volts ac, 50 Hz	17A (peak surge for five seconds)
Running current for:	
120 Volts ac, 60 Hz	7.5 A (measured at 117 V)
220-240 Volts ac, 50 Hz	3.75 A (estimated)
Power factor	0.8
Line cord length (from cabinet)	(8.5 feet)
Plug type	
120 Volts ac, 60 Hz	NEMA L5-30P
220-240 Volts ac, 50 Hz	NEMA 6-15P

5.4.2 RA80 Controls And Indicators

5.4.2.1 Operator Control Panel -

The operator control panel is located on the front of the disk drive (Figure 5-12).

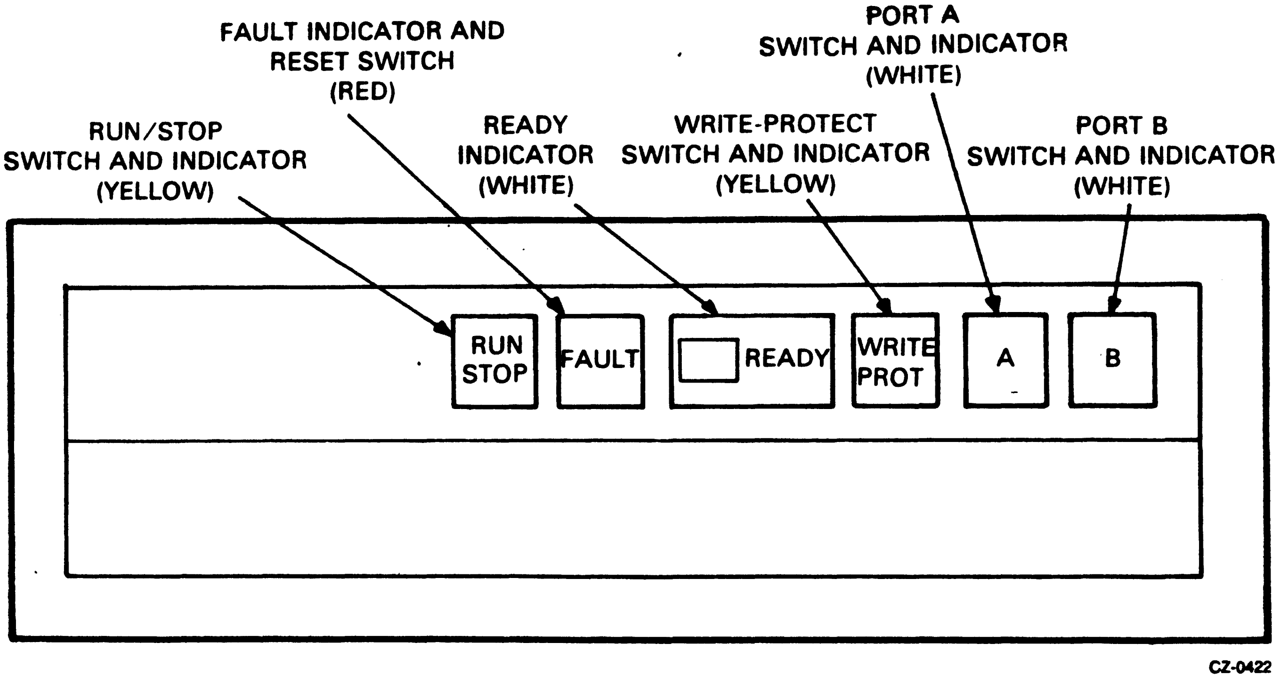

FAULT INDICATOR AND RESET SWITCH (RED)
PORT A SWITCH AND INDICATOR (WHITE)
RUN/STOP SWITCH AND INDICATOR (YELLOW)
READY INDICATOR (WHITE)
WRITE-PROTECT SWITCH AND INDICATOR (YELLOW)
PORT B SWITCH AND INDICATOR (WHITE)
RUN STOP
FAULT
READY
WRITE PROT
A
B
CZ-0422

Figure 5-12 RA80 Controls and Indicators

RUN/STOP Switch and Indicator

The RUN/STOP switch is a two-position push button. In the RUN (button in) position, a spin-up sequence begins if all power has been turned on and no faults are detected. A spin-up attempt will be made following power interruptions as long as the switch remains in this position.

NOTE

> A three-minute delay is recommended between each stop and restart to prevent opening the thermal circuit breaker associated with the spindle motor.

In the STOP (button out) position, power to the spindle motor is turned off. As soon as spindle rotation stops, the RUN/STOP indicator goes off.

FAULT Switch and Indicator

The FAULT light indicates an error condition within the drive, and the FAULT reset switch is used to correct errors.

The FAULT indicator lights whenever a device-unsafe or error condition occurs within the disk drive. If a fault occurs during a spin-up attempt, the spin-up will be aborted.

The FAULT indicator and five other front-panel indicators also provide a lamp display code that identifies various faults and errors as they occur.

READY Indicator

The READY indicator lights following a successful spin-up sequence, indicating that the drive is up to speed and ready to receive commands. The READY indicator goes off during seek operations.

The numbered switch cap over the READY indicator lamp also serves as a unit address plug which distinguishes one drive unit from another in multi-drive arrangements.

WRITE PROT Switch and Indicator

The WRITE PROT switch is a two-position push button with a built-in indicator lamp. Pushing this switch in places the disk drive in the write-protect mode and lights the indicator lamp. In this mode, the write circuits of the disk drive are disabled and no data can be written.

Releasing the push button enables the write circuitry and turns off the WRITE PROT indicator. The WRITE PROT switch is inoperable when the drive is placed off-line.

A and B Port Select Switches

The RA80 drive is equipped with two port select push buttons. These switches (A and B) have built-in lamps, which indicate the port through which the drive is being accessed. See Table 5-14.

Table 5-14 RA80 A and B Port Select Switches

PORT SELECT SWITCH A	B	PORT AVAILABILITY
IN	IN	Dual-port programmable
IN	OUT	A
OUT	IN	B
OUT	OUT	Neither port available

5.4.2.2 Circuit Breakers -

A circuit breaker is located on the rear panel of each disk unit (Figure 5-13). Open the rear door to gain access to the circuit breaker. This circuit breaker controls the ac power to the dc power supply, the cooling fans, and the spindle motor. The circuit breaker must be in the ON position before the drive will operate.

Another circuit breaker is located on the rear of the power controller (Figure 5-13). This circuit breaker controls power to all assemblies within the RA80 cabinet.

The local/remote switch also controls power to the drive. It should always be turned to the local position (Figure 5-13).

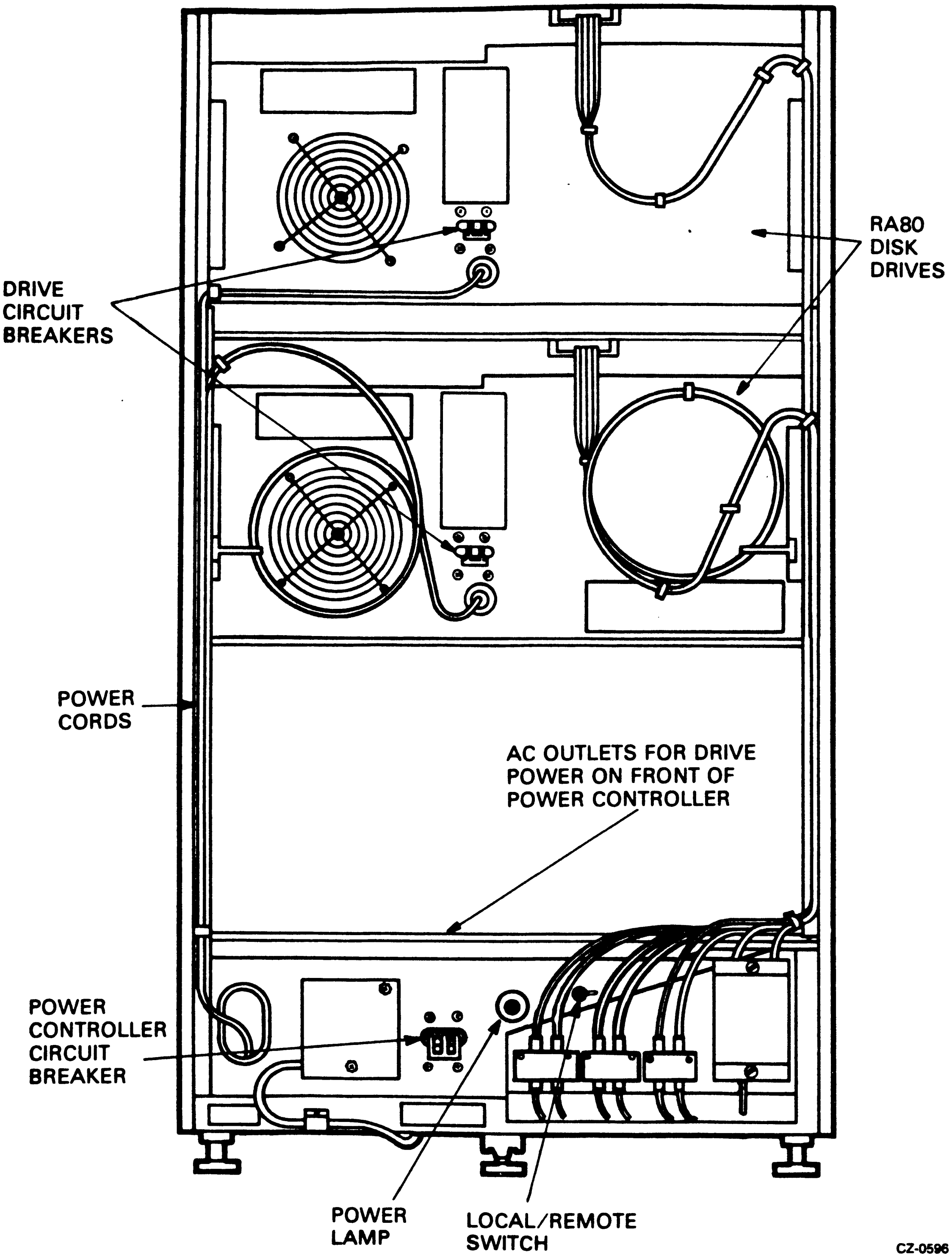
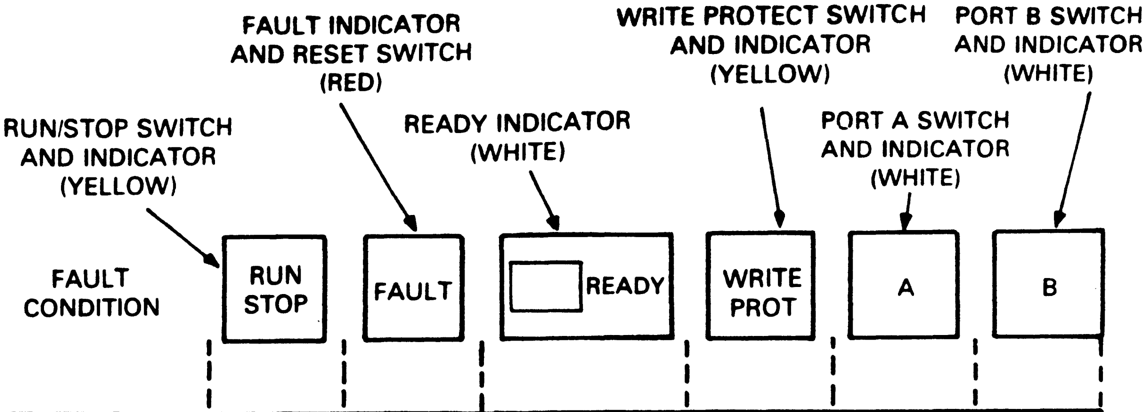


Figure 5-13 RA80 Circuit Breakers

5.4.2.3 Front Panel Fault Indicators -

In addition to their usual functions, the six front panel controls and indicators provide a visual code that identifies generic faults within the RA80. Figure 5-14 shows the faults that can occur and the indicator lamp combinations that identify each particular fault.

When the FAULT indicator light comes on, pressing the FAULT switch once causes all of the indicator lamps to light momentarily as a means of verifying that the lamps are functioning. Pushing the FAULT switch a second time stores the identity of the fault in the microprocessor and clears the fault.


FAULT INDICATOR AND RESET SWITCH (RED)
WRITE PROTECT SWITCH AND INDICATOR (YELLOW)
PORT B SWITCH AND INDICATOR (WHITE)
RUN/STOP SWITCH AND INDICATOR (YELLOW)
READY INDICATOR (WHITE)
PORT A SWITCH AND INDICATOR (WHITE)
RUN STOP
FAULT
READY
WRITE PROT
A
B

FAULT CONDITION	RUN STOP	FAULT	READY	WRITE PROT	A	B
SPIN-UP		ON				ON
R/W DIAGNOSTIC	•	ON			ON	
READ/WRITE UNSAFE	•	ON			ON	ON
SPINDLE MOTOR INTERLOCK	•	ON		ON	ON	
SPINDLE MOTOR SPEED	•	ON		ON	ON	ON
DRIVE DISABLED BY "DD" BIT	•	ON	ON			ON
HDA OR SERVO BOARD OVERTEMPERATURE	•	ON	ON		ON	
MICROCODE	•	ON	ON		ON	ON
SERVO DIAGNOSTIC	•	ON	ON	ON		ON
INITIAL RECAL	•	ON	ON	ON	ON	
MICROPROCESSOR HARDCORE TEST	ON	ON	ON	ON	ON	ON

*THE INDICATOR STATE WILL BE THE SAME AS IT WAS BEFORE THE FAULT SWITCH WAS PUSHED

CZ-0428

Figure 5-14 RA80 Operator Control Panel Fault Indicators

5.4.3 RA80 Operating Instructions

5.4.3.1 Spin-Up -

Push in the RUN/STOP button on the drive front panel to begin the spin-up cycle. The RUN/STOP indicator will immediately light and the READY indicator will light as soon as the drive has completed its spin-up cycle.

5.4.3.2 Spin-Down -

Release the RUN/STOP button on the drive front panel to spin-down the disks. The READY and RUN/STOP indicators will go out. Power is still applied to the drive until the circuit breaker is switched off.

5.4.3.3 Removing RA80 Power -

To remove power from the RA80, release the RUN/STOP button to spin down the disks. Wait until the RUN/STOP indicator goes off. Then switch off the circuit breaker at the rear of the disk drive and unplug its ac cord from the power controller.

5.4.4 RA80 Customer Care And Maintenance

The air filter in the logic chassis assembly should be inspected and cleaned every six months.

5.4.4.1 Filter Removal -

1. Pull down the access door on the front bezel of the drive (Figure 5-15).

CAUTION

> Placing tools on the open access door or leaning on it can damage the hinge mechanism.

2. Grasp the filter material and pull it outward until it is free of the bezel.

5.4.4.2 Filter Cleaning -

1. Wash the foam air filter in warm water using a mild detergent.
2. Rinse the filter with clear water, and allow it to dry.

CAUTION

A wet filter can cause damage to the RA80 drive.

5.4.4.3 Filter Replacement -

1. Slide the filter into place inside the bezel.

2. Close the access door.

NOTE

If it becomes necessary to replace the existing filter, order DIGITAL Part No. 7422816-00.

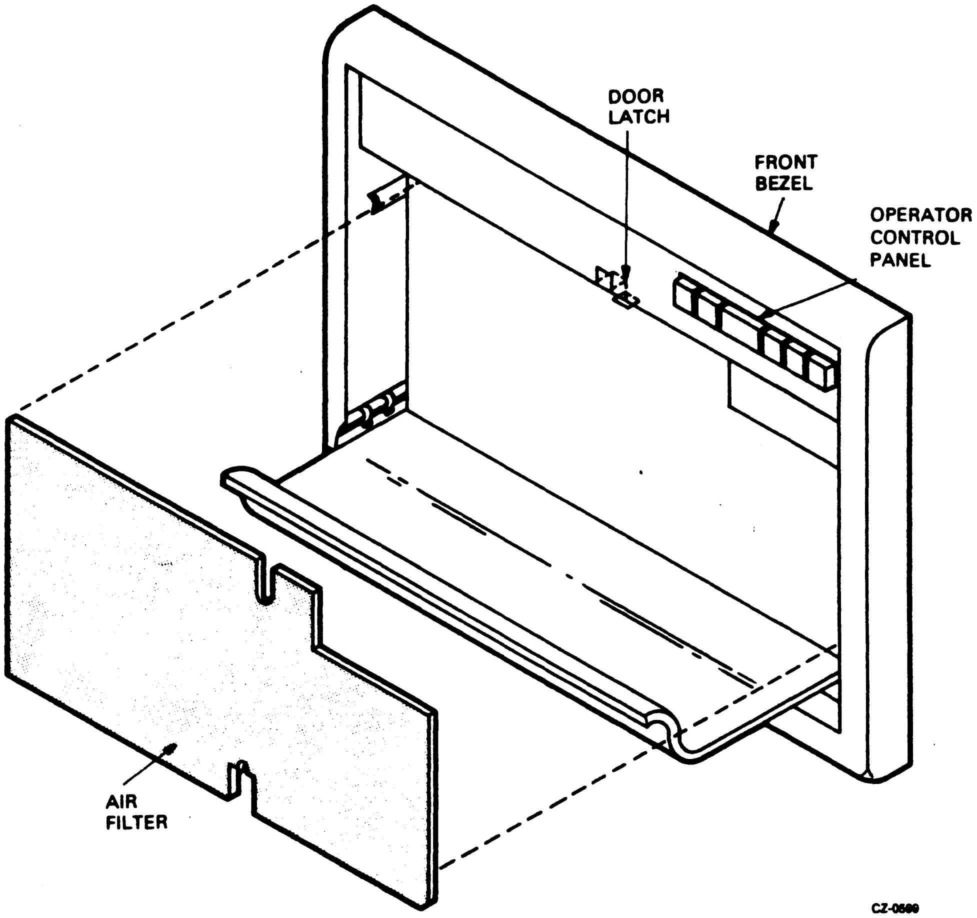


Figure 5-15 RA80 Air Filter Location

5.5 RA81 DISK DRIVE SUBSYSTEM

The RA81 is a random-access, moving-head disk drive with non-removable media. The RA81 has a data storage capacity of 456 megabytes in 16-bit word format. The RA81 connects to the controller via the standard disk interconnect (SI). Additional disk drives can be connected to a controller to increase the data storage capacity. The RA81 can also be connected in a dual-port arrangement permitting time-shared access by two controllers.

5.5.1 RA81 Specifications

Table 5-15 RA81 Specifications

Characteristic	Specification
Data storage capacity	
(Single drive)	
16-bit word format	456 MB
Media type	Winchester double density
Head disk assembly(HDA)	
Number of disks	4
Disk surfaces	7 data, 1 servo
Number of heads (groups)	14 data, 1 servo
Heads per disk surface	2
Cylinders per head	1258 (numbered 0 through 1257)
Sectors per track	
16-bit word format	52
Servo system	Dedicated and embedded
Performance characteristics	
Disk encoding method	3/2
Linear bit density	11,400 bpi, 8550 flux changes per inch at inner group
Bytes per track (unformatted)	36,352
Transfer rate	17.4 megabits per second
Bit cell period	57 ns (nominal)
Read initialization time	
Write-to-read recovery	10 microseconds (excluding data separator sync)

Table 5-15 RA81 Specifications (Cont)

Characteristic	Specification
Read-to write recovery	1 microsecond
Data separator synchronization	10 microseconds
Rotational latency	
Rotational speed	3600 rpm, +2.5%, −3.5%
Average rotational latency	8.33 ms
Maximum rotational latency	16.66 ms
Head switch latency (switch on a given cylinder)	6 ms maximum
Average seek	28 ms maximum
One cylinder	7 ms maximum
Maximum seek	50 ms (1258 cylinders)
Disk rotation start/stop times	
Start time	20 seconds maximum
Stop time	20 seconds maximum
Maximum start/stop frequency	6 (3 minutes between cycles with drive powered up and ready during one of the 3-minute cycles)
Physical characteristics (Drive only)	
Width (nominal)	44.5 centimeters (17.50 in.)
Depth (nominal)	67.3 centimeters (26.50 in.)
Height (nominal)	26.4 centimeters (10.38 in.)
Weight (approx.)	67.1 kilograms (148 pounds)
Environmental limits	
Temperature	
Operating	10 to 40 degrees C (50 to 104 degrees F) with a temperature gradient of 20 degrees C/hour (36 degrees F/hour)
Non-operating	−40 to +60 degrees C (−40 to +140 degrees F)
Relative humidity	
Operating	10% to 85% (non-condensing) with a maximum wet bulb temperature of 28 degrees C (82 degrees F) and a minimum dew point of 2 degrees C (36 degrees F)

Table 5-15 RA81 Specifications (Cont)

Characteristic	Specification
Shipping and storage	10% to 85% with no condensation
Heat dissipation	644 watts (nominal)
Altitude	
Operating	Sea level to 2400 meters (8000 feet) above sea level. Note: Maximum allowable temperatures are reduced by a factor of 1.8 degrees C/1000 meters (1 degree F/1000 feet) for operation at higher altitude sites
Non-operating and shipping	300 meters (1000 feet) below sea level to 9000 meters (30000 feet) above sea level (actual or by means of cabin pressurization)
Power	
Voltage requirements (Single phase)	
120 volt, 60 Hz drives	90-128 volts ac, 60 plus or minus 1 Hz
220-240 volt, 50 drives	184-256 volts ac, 50 plus or minus 1 Hz
Current requirements	
Starting current for 120 volt ac drives	35 ampere peak surge for 4 seconds
Running current for 120 volt ac drives	7.8 amperes rms at 120 volts
Starting current for 220-240 volt ac drives	18 ampere peak surge for 4 seconds
Running current for 220-240 volt ac drives	3.5 amperes rms at 240 volts
Power requirements	
Starting power for 120 volt ac drives	4200 watts at 120 volts

Table 5-15 RA81 Specifications (Cont)

Characteristic	Specification
Running power for 120 volt ac drives	644 watts at 120 volts
Starting power for 220-240 volt ac drives	4320 watts at 240 volts
Running power for 220-240 volt ac drives	496 watts at 240 volts
Power factor	0.9
Line cord length (from cabinet)	259 cm (8.5 feet)
Plug type	
120 volts,60 Hz	Hubbel No.2611 NEMA No.L5-30P DEC No.12-11193 (See Figure 2-1)
240 volts, 50Hz	Hubbel No.2321 NEMA No.L6-20P DEC No.12-11192 (See Figure 2-1

5.5.2 RA81 Controls And Indicators

5.5.2.1 Operator Control Panel -

The RA81 operator control panel is located on the front of the disk drive. See Figure 5-16.

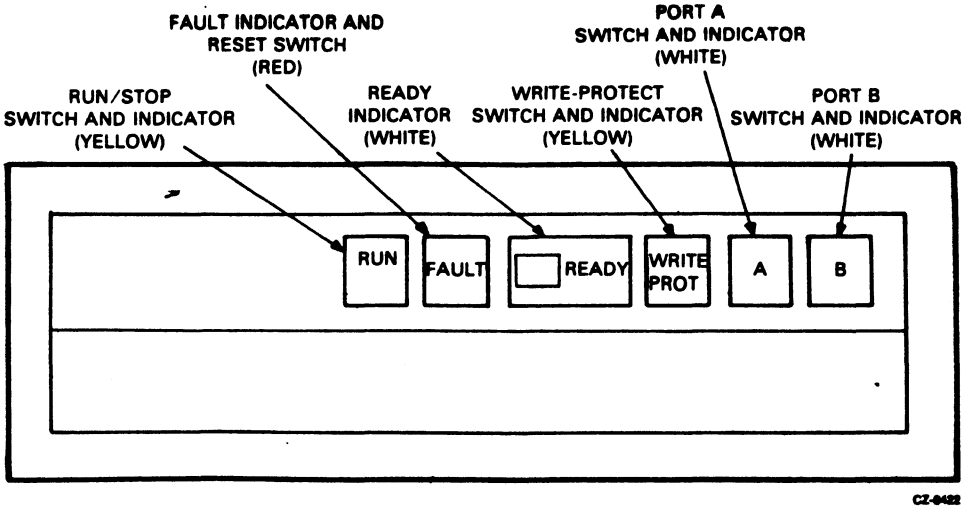

FAULT INDICATOR AND RESET SWITCH (RED)
PORT A SWITCH AND INDICATOR (WHITE)
RUN/STOP SWITCH AND INDICATOR (YELLOW)
READY INDICATOR (WHITE)
WRITE-PROTECT SWITCH AND INDICATOR (YELLOW)
PORT B SWITCH AND INDICATOR (WHITE)
RUN
FAULT
READY
WRITE PROT
A
B
CZ-0422

Figure 5-16 RA81 Front Panel Controls and Indicators

RUN/STOP Switch and Indicator

The RUN/STOP switch is a double-action push button. In the RUN (button in) position, the switch requests a drive spin-up and permits initialization of the drive for operation. In the STOP (button out) position, the heads are moved off of the data bands and the spindle motor is stopped.

If the drive is on-line and operating when the switch is placed in the STOP position, the drive will complete all outstanding write operations commanded by the controller before spinning down.

The RUN/STOP indicator light always reflects the state of the drive spindle. The indicator is on while the spindle is turning and off when the spindle has stopped.

If the RUN/STOP switch is in the RUN position on power up, spin-up will occur by the controller or by manually toggling the RUN/STOP switch.

NOTE

> A minimum three-minute delay is recommended between each start and restart to prevent opening the thermal circuit breaker associated with the spindle motor.

FAULT Switch and Indicator

The FAULT light indicates an error condition within the drive, and the FAULT reset switch is used to correct errors.

When the disk drive detects an error, the FAULT light is turned on. The drive stays in its current operating state until the condition is corrected. Some faults are corrected automatically and the light will go off. If the fault condition is not corrected, the light will stay on and the FAULT button will have to be pushed. If a fault occurs during a spin-up try, the spin-up will be aborted.

The FAULT button is a momentary contact switch. When pushed the first time, the switch causes the drive to enter an off-line state (relative to the controller) and to assume control of all front-panel indicator lights in order to display a blinking error code. Refer to Table 5-16 for identification of faults and indicator light combinations. The drive stays off-line as long as the fault code is being displayed. When the FAULT switch is pushed the second time, the drive tries to clear the error condition, stores the error code, enters the available operating state, and returns the front-panel indicators to their normal service. Up to 16 error codes can be stored for later use in troubleshooting the drive.

Table 5-16 RA81 Front-Panel Fault Identification Codes

FAULT CONDITION	**RUN/ STOP**	**FAULT**	**RDY**	**WRITE PROT**	**A**	**B**
Spin error		on				on
Microprocessor fault		on			on	
SDI error		on			on	on
Master/slave error		on		on		
Servo fine positioning error		on		on		on
Servo coarse positioning error		on		on	on	
Spindle motor interlock error		on		on	on	on
Servo or HDA overtemp error		on	on			

Table 5-16 RA81 Front-Panel Fault Identification Codes (Cont)

FAULT CONDITION	RUN/ STOP	FAULT	RDY	WRITE PROT	A	B
R/W unsafe error		on	on			on
R/W command error		on	on		on	
Servo/HDA interlock error		on	on		on	on
Servo/ microprocessor interlock error		on	on	on		
Read/write microprocessor interlock error		on	on	on		on
Control panel/ microprocessor interlock error		on	on	on	on	
Personality/ microprocessor interlock error		on	on	on	on	on
Index error	on	on				
Write and bad embedded data	on	on				on
Drive disabled by DD bit	on	on			on	
Write enable and write protect asserted error	on	on			on	on
Servo diagnostic test error	on	on		on		
R/W diagnostic test error	on	on		on		on
Diagnostic idle loop test error	on	on		on	on	
DC low	on		on	on	on	on
Microprocessor hardcore test error	on	on	on	on	on	on

UNIT/READY

The READY indicator is preceded by a successful (fault-free) spin-up sequence and indicates that the drive is in a read/write ready condition. The READY indicator will only come on when the RUN/STOP switch is in the RUN position and the RUN light is on. The READY indicator goes off during seek operations.

The numbered switch cover over the READY indicator light is also a logic plug which distinguishes one drive from another in multi-drive arrangements. The switch cover has a unique cam-coded configuration that corresponds to the drive unit number. The cam-coded switch sets up a unit code corresponding to the drive number when plugged into the mating cam-encoded switch located behind the READY indicator light. Refer to Figure 5-17. Drive selection is accomplished when the address on the drive select lines matches the code established by the switch cover.

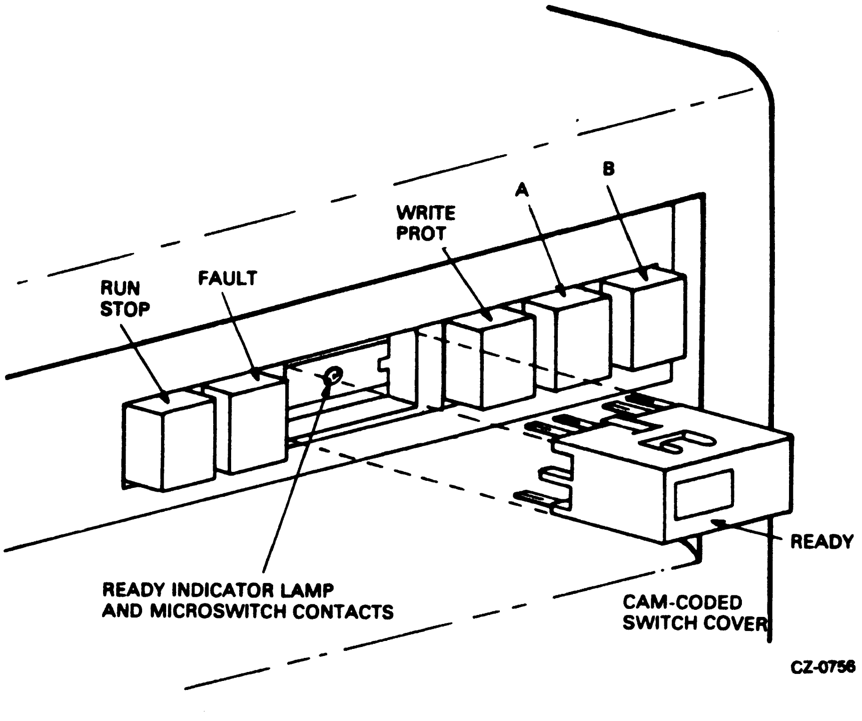


Figure 5-17 RA81 Cam-Coded Switch Cover

WRITE PROT

The WRITE PROT switch is a double-action push button with a built-in indicator light. Pushing in on this switch places the drive in the write protect mode and turns on the indicator light. The write circuits of the drive are disabled in this mode.

Releasing the push button enables the write circuits and turns off the WRITE PROT indicator light. When the disk is not read/write ready, it is always write protected.

A and B Port Select Switches

The RA81 disk drive has two port select push button switches. These double-action switches (A and B) have built-in lights that indicate the port through which the drive is being accessed. If the drive is equipped with dual-port abilities, any one of three modes can be selected with the A and B switches. See Table 5-17.

Table 5-17 RA81 A and B Port Select Switches

PORT SELECT SWITCH A	PORT SELECT SWITCH B	PORT AVAILABILITY
IN	IN	Dual-port programmable
IN	OUT	A
OUT	IN	B
OUT	OUT	Neither port available

5.5.2.2 Circuit Breakers -

Each disk drive has a circuit breaker on the back panel as well as a breaker on the back of the power controller in the bottom of the cabinet. Access to these breakers is through the back door of the cabinet. The breakers on the drive control the application of ac power to the drive dc power supply, the cooling fans, and the spindle motor. This breaker must be on before the drive will operate.

The power controller breaker regulates the application of ac power to all assemblies within the RA81 cabinet and must be turned on before the drive breaker can be effective.

5.5.3 RA81 Operating Instructions

5.5.3.1 Drive Operation -

Spin-Up

Push in the RUN/STOP button on the drive front panel to spin-up the drive. The RUN indicator will light immediately and the READY indicator will come on as soon as the drive has completed the spin-up cycle.

Spin-Down

Release the RUN/STOP button (out position) on the drive front panel to stop the drive. The READY indicator will go off immediately and the RUN indicator will go off as soon as disk rotation stops.

Removing RA81 Power

Do a spin-down as described above. After the RUN indicator has gone off, turn the drive breaker on the back panel off and disconnect the cord going to the power controller. This will remove all power to the drive.

5.5.4 RA81 Customer Care And Maintenance

The air filter in the logic chassis assembly should be inspected and cleaned every six months.

5.5.4.1 Air Filter Removal -

To remove the air filter, proceed as follows:

1. Pull down the access door on the front bezel of the drive. Refer to Figure 5-18.

NOTE

> The hinge mechanism can be damaged by placing tools on the open access door or by leaning on it.

2. Grasp the filter material and pull it outward until it is free of the bezel.

5.5.4.2 Filter Cleaning -

1. Wash the foam air filter in warm water using a mild detergent.
2. Rinse the filter material with clear water and then allow it to dry.

CAUTION

A wet filter can cause damage to the RA81 drive.

5.5.4.3 Filter Replacement -

1. Slide the filter into place inside the bezel.
2. Close the access door.

NOTE

Replace the filter by ordering DIGITAL part number 7422816-00.

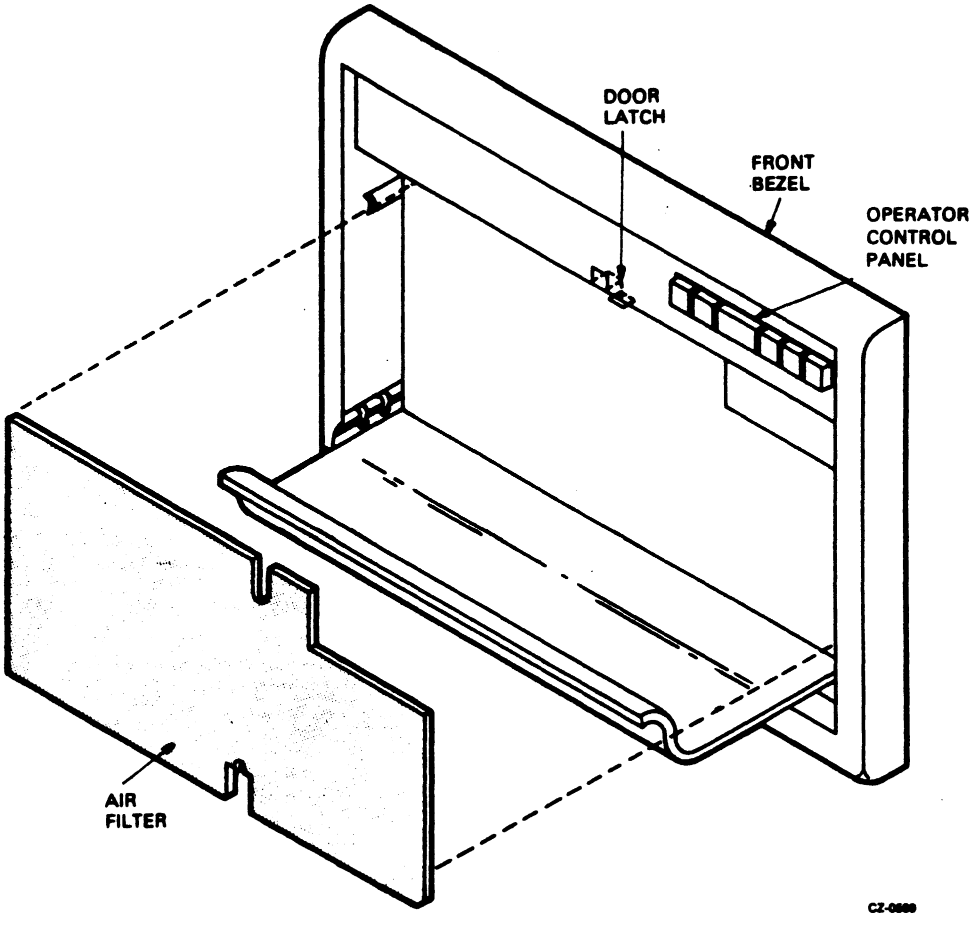


Figure 5-18 RA81 Air Filter Removal

5.6 RM05 DISK SUBSYSTEM

The RM05 is a removable media disk subsystem that has a formatted data storage capacity of 256 megabytes. The RM05 uses a 14-inch disk pack with 12 platters. Data is recorded in 16-bit words.

5.6.1 RM05 Specifications

Table 5-18 RM05 Specifications

Characteristic	Specification
Seek time	
Maximum seek (822 cylinder)	55 ms
One cylinder seek (maximum)	6 ms
Average seek	30 ms
Seek to the same cylinder	4 μs
Latency	
Speed	3600 r/min
Maximum latency	17.3 ms
Average latency	8.33 ms
Start/stop time	
Start (maximum)	35 s
Start (typical)	25 s
Stop (with power) (maximum)	35 s
Stop (with power) (typical)	25 s
Stop (without power)	120 s
Heads	
Servo head	1
Read/write heads	19
Data rates	
Bit cell time	103.3 ns
Word rate	1.65 μs
Number of addressable registers in RM05 adapter	16
Error detection/correction	32-bit ECC/sector
Time for error correction	4.47 ms, maximum

Table 5-18 RM05 Specifications (Cont)

Characteristic	Specification
Environmental limits	
Temperature	
Operating:	15.0° to 32.2° C (59° to 90° F) with a maximum gradient of 6.7° C (12° F) per hour.
Non-operating:	−40° to 70.0° C (−40° to 158° F) with a maximum gradient of 20° C (36° F) per hour.
Relative humidity	
Operating:	20 to 80 percent (providing there is no condensation)
Non-operating:	5 to 95 percent (providing there is no condensation)
Altitude	
Operating:	305 m (1000 ft) below sea level to 2000 m (6500 ft) above sea level
Non-operating:	305 m (1000 ft) below sea level to 4572 m (15,000 ft) above sea level
Electrical	
Voltages available (Drive)	208 V (+14.6, −29.0), 60 Hz 230 V (+14.5, −32.0), 60 Hz 220 V (+15.0, −25.0), 50 Hz 240 V (+17.0, −27.0), 50 Hz
Voltages available (Adapter)	120 V (+8.0, −8.0), 60 Hz 220 V (+15.0, −25.0), 50 Hz 240 V (+17.0, −27.0), 50 Hz
Start current	208 Vac, 60 Hz @ 39.0 A rms max 230 Vac, 60 Hz @ 40.0 A rms max 220 Vac, 50 Hz @ 40.0 A rms max 240 Vac, 50 Hz @ 41.0 A rms max

Table 5-18 RM05 Specifications (Cont)

Characteristic	Specification
Line current	
Disk and carriage in motion	208 Vac, 60 Hz @ 8.0 A rms max 230 Vac, 60 Hz @ 7.2 A rms max 220 Vac, 50 Hz @ 9.5 A rms max 240 Vac, 50 Hz @ 8.7 A rms max
Disk not in motion	208 Vac, 60 Hz @ 2.0 A rms max 230 Vac, 60 Hz @ 1.8 A rms max 220 Vac, 50 Hz @ 2.5 A rms max 240 Vac, 50 Hz @ 2.3 A rms max
Adapter	120 Vac, 60 Hz @ 2.1 A rms max 220 Vac, 50 Hz @ 1.3 A rms max 240 Vac, 50 Hz @ 1.4 A rms max
Line cord length	366 cm (12 ft)
Disk cartridge type	RM05P
Weight	
RM05 drive and cabinet	249 kg (550 lbs)
Adapter cabinet	54 kg (120 lbs)
With one adapter	91 kg (200 lbs)
With two adapters	127 kg (280 lbs)
AC plug types	
RM05 drive	
120 volt 60 Hz	NEMA L6-20P
208 volt 50 Hz	Not shipped
Adapter	
120 volt 60 Hz	NEMA 5-15P
240 volt 50 Hz	NEMA 6-15P

Table 5-19 RM05P Disk Pack Specifications

Characteristic	Specification
Disk diameter	35.56 cm (14 in)
Number of disks	12 (the upper and lower disks are not used for recording)
Number of recording surfaces	19 read/write and one read-only servo surface
Cylinders per disk pack	823
Total number of tracks	15,637 per disk pack
Tracks per cylinder	19
Tracks per inch	384
Bad sector file	Cylinder 822, track 18
Environmental requirements	
Temperature range	
Operating:	10° to 57° C (50° to 135° F); temperature change rate not to exceed 0.1° C (0.2° F) per minute
Non-operating:	−40° to 65° C (−40° to 150° F); temperature change rate not to exceed 14° C (25° F) per hour
Relative humidity	
Operating and non-operating:	8 to 80 percent
Wet bulb reading	
Operating:	25° C (78° F), maximum
Non-operating:	30° C (85° F), maximum
Altitude	
Operating:	Sea level to 3,050 m (10,000 ft)
Non-operating:	Sea level to 12,190 m (40,000 ft)
Stray magnetic fields	
Operating and non-operating:	Not to exceed 50 oersteds

5.6.2 RM05 Controls And Indicators

The operator controls and indicators for the RM05 disk subsystem are on the front panel and inside the rear door on the power supply control panel. The controls and indicators for the RM adapter are on the front panel and inside the rear door on the adapter power supply.

5.6.2.1 Operator Control Panel -

The operator control panel is on the front of the disk drive cabinet and contains four controls and four indicators (Figure 5-19).

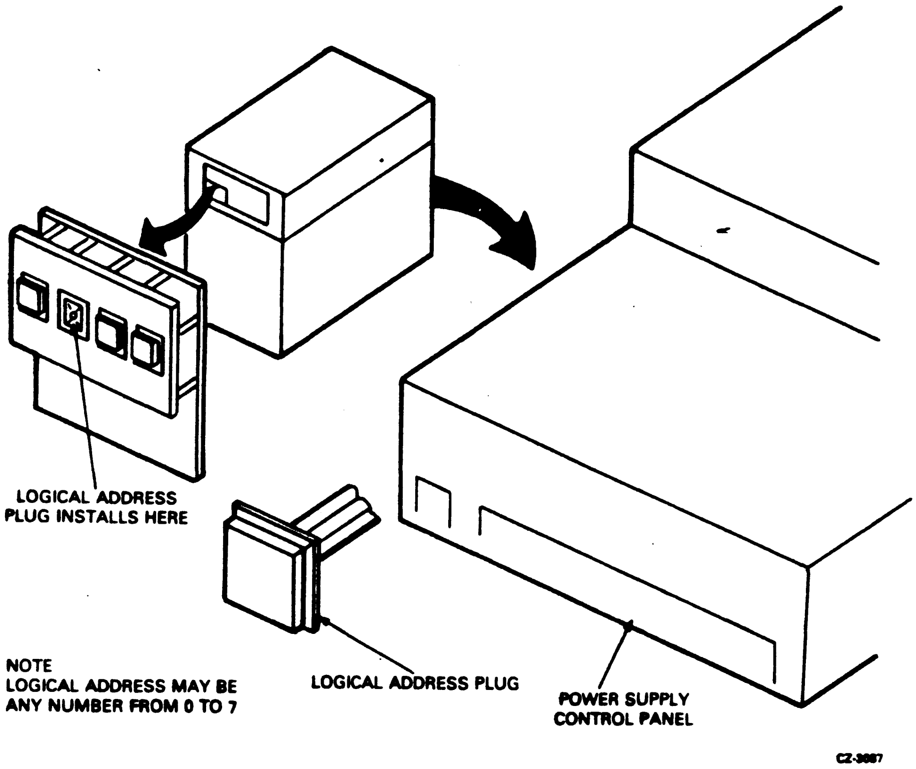


Figure 5-19 RM05 Controls and Indicators

START Switch

The START switch is a two-position push button. Pressing the START switch when the drive is in the power-off condition (disk pack not spinning) lights the START indicator and initiates the power-up sequence provided the conditions listed below are met.

- Disk pack is installed.

- Pack access cover is closed.

- All power supply circuit breakers are on.

START Indicator

This indicator lights when the START switch is depressed from the OFF position provided the three conditions mentioned above are met. The light indicates that power is on.

READY Indicator

This indicator lights when the disk is up to speed, the heads are loaded, and no fault condition exists.

Logical Address Plug

This plug determines the logical address of the drive. The address can be changed to any number from 0 to 7.

FAULT Indicator

This indicator lights when any of the faults listed below exist in the drive.

- Write fault
- More than one head selected
- Read and write gates selected simultaneously
- Read or write and off cylinder
- Voltage fault

Once the fault no longer exists, the FAULT indicator can be turned off by any of the following methods.

- Pushing the FAULT button on the operator panel
- Issuing a fault clear signal from the controller
- Issuing an initialize signal from the controller

FAULT Clear Switch

This switch is a momentary-contact push button used to clear the FAULT indicator. Depressing the FAULT clear switch turns off the FAULT indicator only if the fault condition no longer exists.

Write PROTECT Indicator

This indicator lights when the drive is in the write protect mode. Data cannot be written on the pack when the drive is in this mode.

Write PROTECT Switch

This switch is a two position push button. Pressing the switch to light the write PROTECT indicator disables the drive write circuits and prevents them from writing data on the pack. Pressing the switch again enables the write circuits and turns off the light.

5.6.2.2 Power Supply Control Panel -

The power supply control panel is located inside the rear door of the drive cabinet, at the bottom of the cabinet. The panel contains the MAIN AC circuit breaker, the LOCAL/REMOTE switch, an HOURS elapsed time meter, and several test points and circuit breakers. The panel is shown in Figure 5-19.

MAIN AC Circuit Breaker

This circuit breaker controls application of site ac power to the drive. Closing this breaker immediately applies power to the blower and elapsed time meter.

HOURS Elapsed Time Meter

This meter records the accumulated ac power-on time. The meter starts when the MAIN AC circuit breaker is closed.

LOCAL/REMOTE Switch

This switch determines whether the drive can be powered up from the drive controls (LOCAL) or from the controller (REMOTE). In the LOCAL position, drive power-up sequence starts when the START switch is pressed. In the REMOTE position, the drive power-up sequence starts when the START switch is pressed and sequence power ground is received from the controller.

Other Circuit Breakers

The other circuit breakers on the power supply control panel control application of associated voltages to the drive and provide circuit overload protection.

5.6.2.3 Port-Select Switches (Dual-Port Option) -

The port-select switches are located on the upper left corner of the front panel of the RM adapter cabinet. Figure 5-20 shows the RM adapter cabinet. Depressing the port A switch allows only the controller connected to part A to access the drive. Depressing the port B switch allows only the controller connected to port B to access the drive. With neither or both port switches depressed, either controller can access the drive depending upon the logic of the RM adapter.

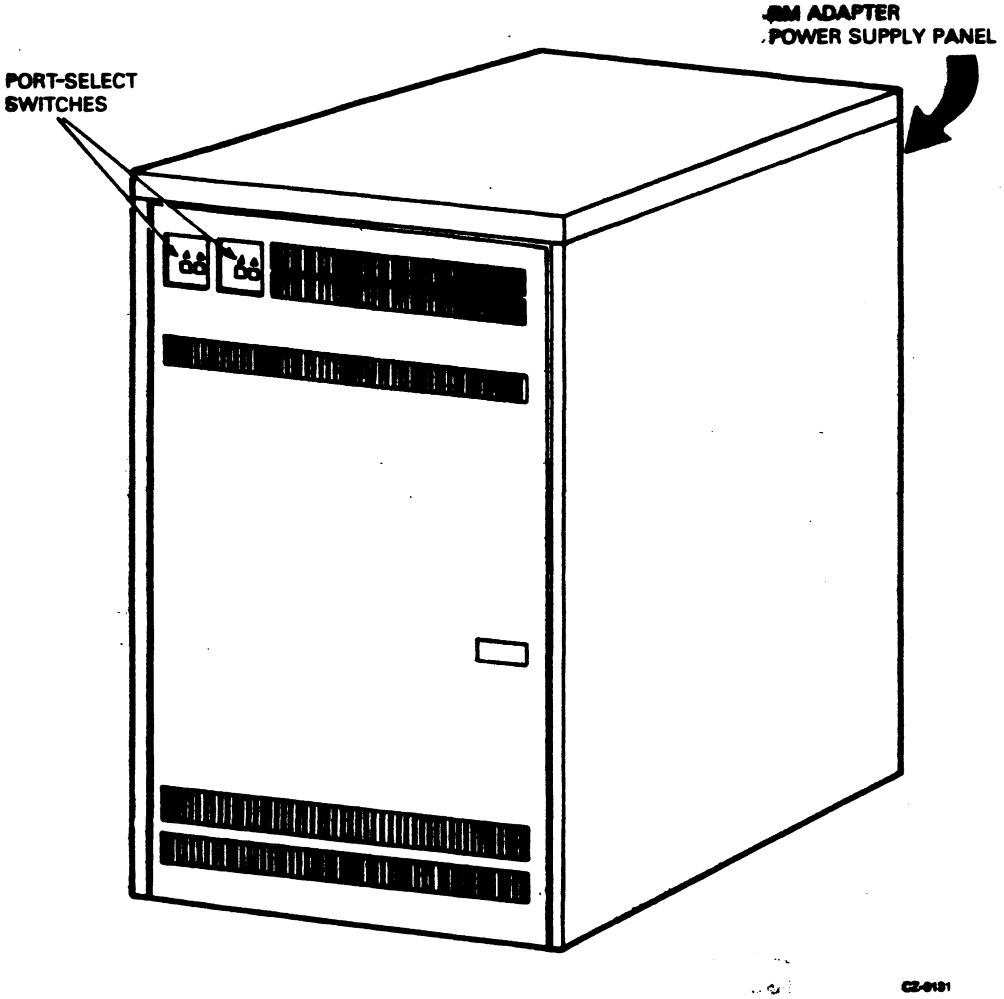


Figure 5-20 RM05 Adapter Cabinet

5.6.2.4 Adapter Power Supply Controls -

The adapter power supply controls are inside the adapter cabinet and can be reached through the rear door of the cabinet. Figure 5-20 shows the location of the adapter power supply, and Figure 5-21 shows the controls.

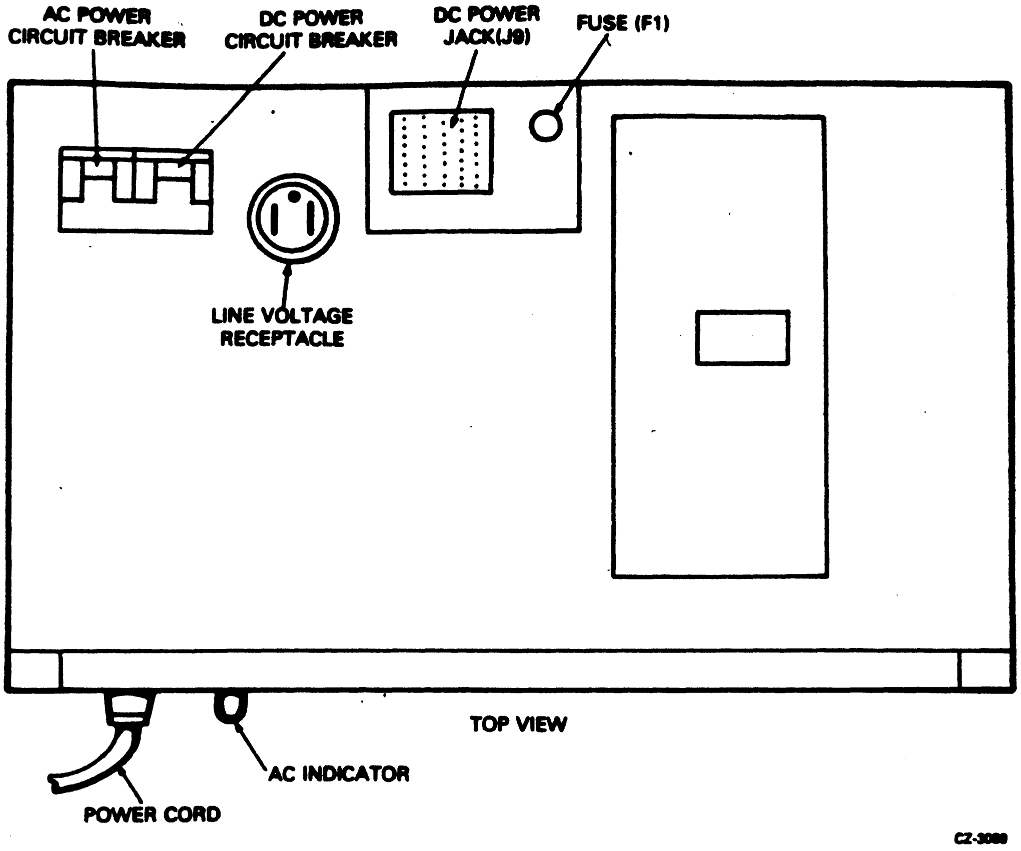


Figure 5-21 RM05 Adapter Power Supply Controls

AC Power Circuit Breaker

This circuit breaker controls ac power to the adapter. When in the ON position, it routes ac power to the adapter. When a second adapter is installed in the upper part of the cabinet, the circuit breaker on the lower adapter also controls power to the upper adapter. The upper adapter is plugged into the lower adapter.

DC Power Circuit Breaker

This circuit breaker applies ac power to the adapter dc power supply if the ac circuit breaker is on. The dc power supply provides the dc voltages needed by the adapter circuits.

-15 V Supply Fuse (Fl)

This fuse protects the transformer from circuit overloads in the -15 V supply.

AC Indicator

This light, located next to the ac power cord, is lit whenever the ac power cord is plugged into a source of ac power. It is a warning indicator which shows that even though the drive is not energized, there is ac voltage present up to the ac circuit breaker terminal.

5.6.3 RM05 Operating Instructions

5.6.3.1 Disk Pack Loading And Drive Start-Up -

Power-On Procedure

1. Turn on the ac and dc power circuit breakers on the RM adapter power supply.
2. Turn on all circuit breakers on the drive power supply control panel.
3. Ensure that the drive blower starts to operate. Wait two minutes before loading a disk pack.

CAUTION

> Failure to allow the blower to operate for two minutes before installing the disk pack may cause damage to the disk pack or heads.

4. Set the LOCAL/REMOTE switch to the desired position.

Disk Pack Loading Procedure

CAUTION

> Ensure that no dust or other foreign particles are in the shroud area. Also, ensure that the blowers operate for at least two minutes before installing a disk pack. This will allow the purging of the blower system.

1. Check that the drive is powered on.
2. Raise the pack access cover and ensure that the heads are fully retracted (Figure 5-22).

CAUTION

> Heads that are not fully retracted may be damaged by or may damage the disk pack. The condition may indicate a servo problem. If the heads are not fully retracted, DO NOT push on them. Contact service personnel.

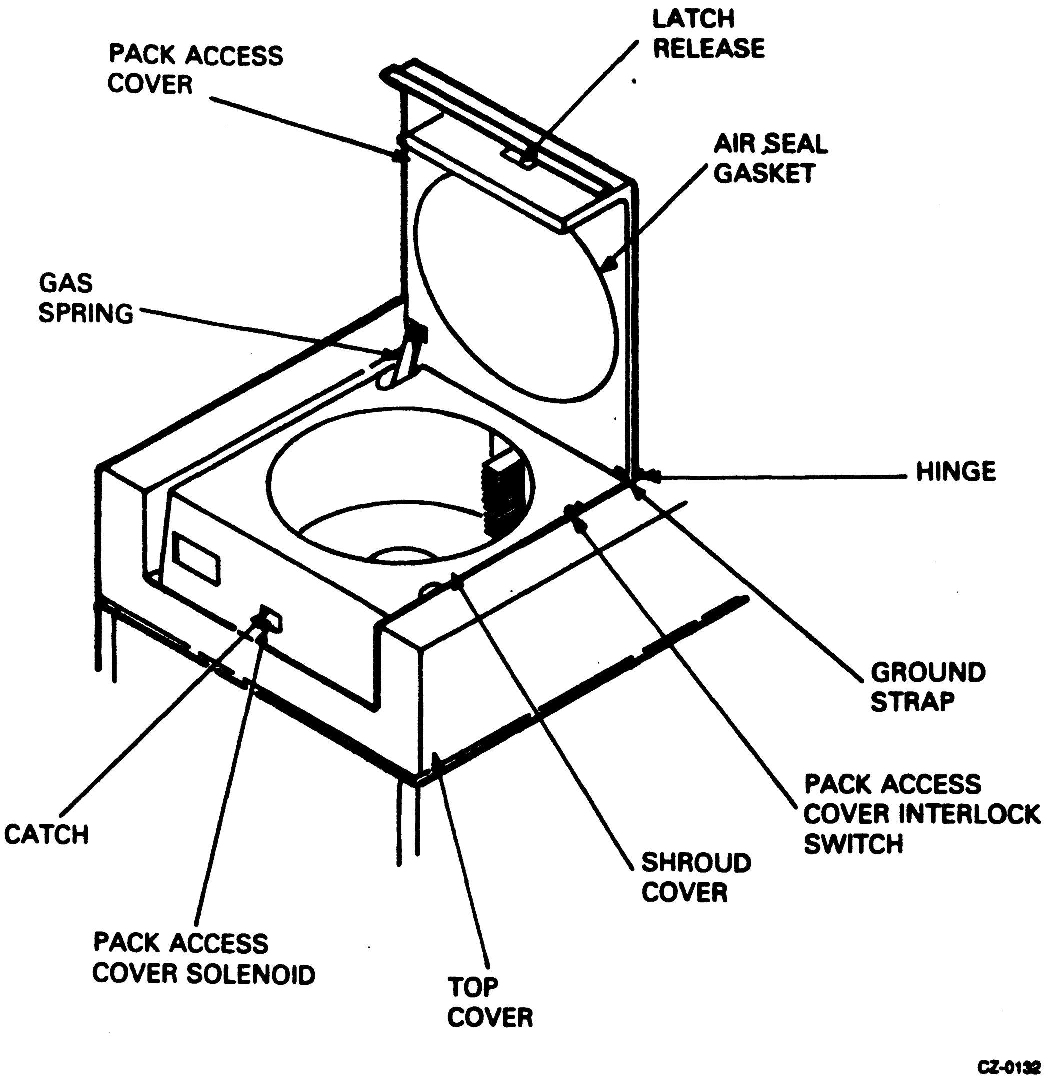


Figure 5-22 RM05 Pack Access

3. Disengage the bottom dust cover from the disk pack by squeezing the levers of the release mechanism in the center of the bottom dust cover (Figure 5-23). Set the cover aside in an uncontaminated storage area.

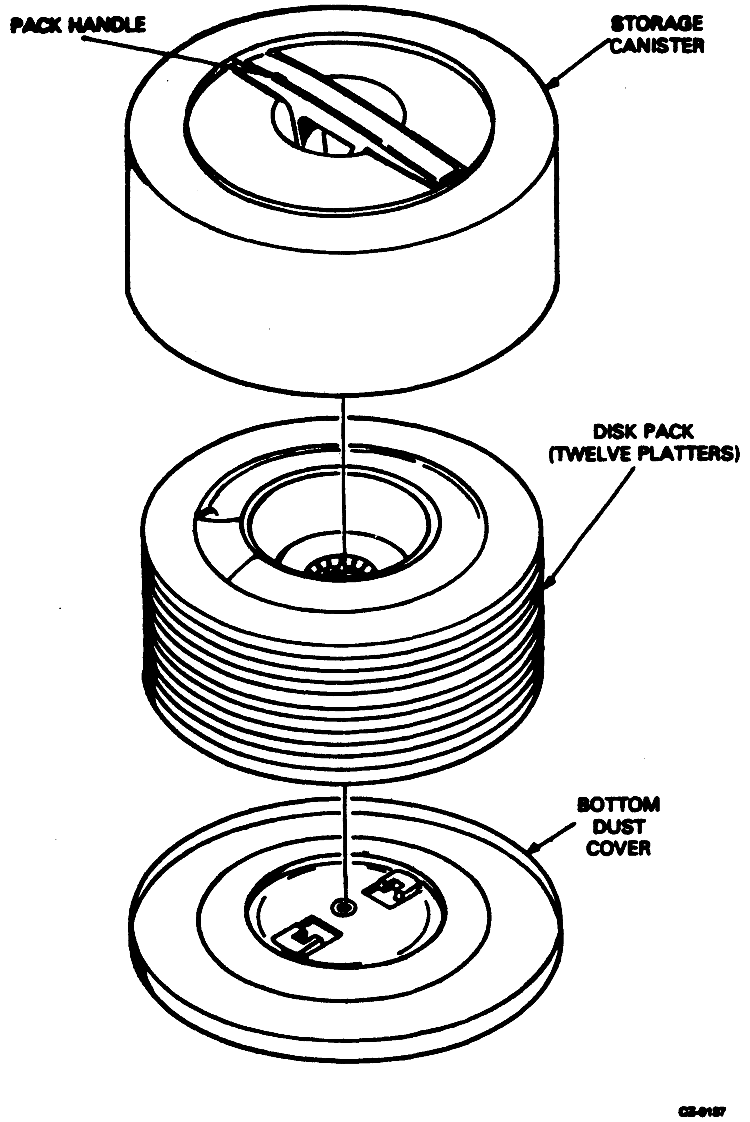


Figure 5-23 RM05P Disk Pack

4. Set the disk pack on the spindle. Avoid using force to place the disk pack on the spindle. Twist the pack handle clockwise until it is secured to the spindle lock-shaft.

5. Lift the top dust cover clear of the drive and store it with the bottom dust cover.

6. Close the pack access cover immediately to prevent dust and other foreign matter from contaminating the disk surfaces.

7. Press the START switch on the operator control panel. The START indicator should come on immediately.

Disk Pack Removal Procedure

1. Spin down the disk by pressing the START switch to unload the heads and stop the drive motor. The READY light will go out immediately and the START light will go out when the disk pack has stopped rotating.

2. Open the pack access cover when disk pack rotation has stopped and the cover has unlocked.

3. Check that the heads have fully retracted.

4. Place the top dust cover over the disk pack so the post protruding from the center of the disk pack fits into the dust cover handle.

5. Turn the dust cover handle counterclockwise until the disk pack is free of the spindle. Avoid using force when removing the disk pack from the spindle.

6. Lift the top cover and disk pack clear of the drive and close the pack access cover.

7. Place the bottom dust cover on the disk pack and store the pack in a proper storage area.

5.6.4 RM05 Customer Care And Maintenance

5.6.4.1 Disk Pack Storage -

To ensure maximum disk pack life and reliability, observe the following precautions.

- Store disk packs in a computer room environment.
- Allow two hours before use for adjustment to the computer room conditions if the disk pack has been stored in a different environment.
- Do not store disk packs in direct sunlight or in a dirty environment.
- Store disk packs flat, not on edge. They may be stacked two high. A pack tree is recommended.

5.6.4.2 Disk Pack Precautions -

- Ensure that both top and bottom plastic covers are on and locked together whenever a disk pack is not installed on a drive.
- Mark packs with a pen or a felt-tip marker that will not produce a loose residue. Never use a lead pencil.
- Do not attach labels to the disk pack. Attach them to the plastic covers. Labels will not remain attached to the pack when it is spinning and catastrophic head crashes may result.
- Do not clean disk surfaces. Cleaning should be done by trained service personnel only.
- Do not touch disk recording surfaces.

5.6.4.3 Disk Pack Handling -

The positive pressure filtration system of the drive eliminates the need for periodic inspection and cleaning of the disk pack. However, certain symptoms may indicate improper operating conditions for the pack. Immediately stop the drive and remove the disk pack if any of the conditions listed below occur.

- A sudden increase in error rates related to one or more heads
- An unusual noise such as pinging or scratching
- A burning odor
- Known or suspected contamination of the pack from dust, smoke, oil, or the like

CAUTION

> Do not operate the pack on another drive until determining that the media is not damaged or contaminated.
>
> Do not operate the drive with another pack before determining that the heads and shroud area are not damaged or contaminated.

Do not attempt to inspect or clean a disk pack. Call in trained service personnel for inspection of the disk pack.

5.7 RP07 DISK DRIVE

The RP07 disk drive is a fixed media, random access, mass storage device. It provides for storage and retrieval of data from any location on the rotating disks. The drive has a moving carriage which positions 32 read/write and one servo head over 17 disk surfaces. It has 630 cylinders available for customer use.

5.7.1 RP07 Specifications

Table 5-20 RP07 Specifications

OPERATION:	
Start-up time, (CB1 ON to heads on cylinder 000)	82 seconds ±3 seconds
Disk rotation	3633 RPM ±3%
Spindle stop time	20 sec
DISK STACK:	
Disk diameter	35.56 cm (14")
Number of disks	9
Data recording surfaces	16
Servo recording surface	1
READ/WRITE:	
Number of Read/Write heads	32
Cylinders per disk stack	2X(630 + 2 FE cylinders) = 1260 + 4 FE cylinders total
Tracks per cylinder	32
Tracks per inch	537 (nominal)
Total tracks per disk stack	20,160 + 64 FE Tracks
Data bytes per track max	
16 bit mode (50 sectors/TK)	
Sector capacity	256 words; 512 bytes
Track capacity	12,800 words; 25,600 bytes
18 bit mode (43 sectors/TK)	
Sector capacity	256 words; 576 bytes
Track capacity	11,008 words; 24,768 bytes
Stack capacity	
16 bit mode (50 sectors/TK)	258 megawords; 516 megabytes
18 bit mode (43 sectors/TK)	221 megawords; 499.3 megabytes

Table 5-20 RP07 Specifications (Cont)

Data transfer rate	
Interleaved sectors	1.3 megabytes/second max (128 word max buffer - 16 or 18 bit mode)
Non-interleaved sectors	2.16 megabytes/second max (16 bit mode) 1.92 megabytes/second max (18 bit mode)
Recording density--in bits per inch (linear)	11,139 BPI on inner track (nominal)

ENVIRONMENT:

Operating temperature range	15°C (59°F) to 32°C (90°F) with max change of 6.6°C (12°F) per hour
Operating humidity range	20%-80% with max wet bulb of 25°C (77°F)
Shipping temperature range	-40°C (-40°F) to +66°C (+151°F) with max change of 14°F/hour
Shipping humidity range	5% to 90%
Drive cooling (internal)	No external forced air is necessary
Heat Dissipation	7000 BTU/Hour (2.051 KW)
Altitude Range - Operating (Normal)	Sea level to 2400 M (sea level to 8000 feet)

POWER REQUIREMENT:

TYPE 3 Voltage Taps	FREQUENCY TOLERANCE	NOMINAL OR TAP VOLTAGE	VOLTAGE RANGE
60 Hz	±2%	200/208	170-220
60 Hz	±2%	220	184-236
60 Hz	±2%	240	201-256
50 Hz	±2%	200/208	170-220
50 Hz	±2%	220	184-236
50 Hz	±2%	240	201-256

Starting current surge	55 amps (max)
Running current	8 amps (max)

Table 5-20 RP07 Specifications (Cont)

PHYSICAL DIMENSIONS OF CABINET:	
Height	118.1 cm (46.5")
Width	57.15 cm (22.5")without side covers 62.23 cm (24.5") with one side cover 67.31 cm (26.5") with two side covers
Depth	83.82 cm (33.0")
Mounting	Adjustable levelers (casters also provided)
Weight	Approximately 181.44 kg (400 lbs) total
Floor loading	382 kg/m^2 (78 lbs per square foot)
RECOMMENDED CLEARANCES:	
Front	71 cm (28") minimum
Rear	71 cm (28") minimum
Side	0" without side covers
Side	31 cm (12") with side covers

Characteristics

ACCESSING:	
Access drive motor	Electromagnetic actuator
Access control	Microprocessor controlled. Closed-loop track following and positioning
Direct access motion:	
Coarse positioning mode	Initially moves heads to desired cylinder
Fine positioning mode	Maintains heads on desired cylinder
Access times:	
One cylinder seek	5 milliseconds (maximum)
Average seek	23 milliseconds
Maximum seek	46 milliseconds
Average rotational latency time	8.26 milliseconds
Maximum rotational latency time	16.52 milliseconds

5.7.2 RP07 Controls And Indicators

The RP07 operator panel (Figure 5-24), located at the front of the drive, enables communication between the operator and the drive.

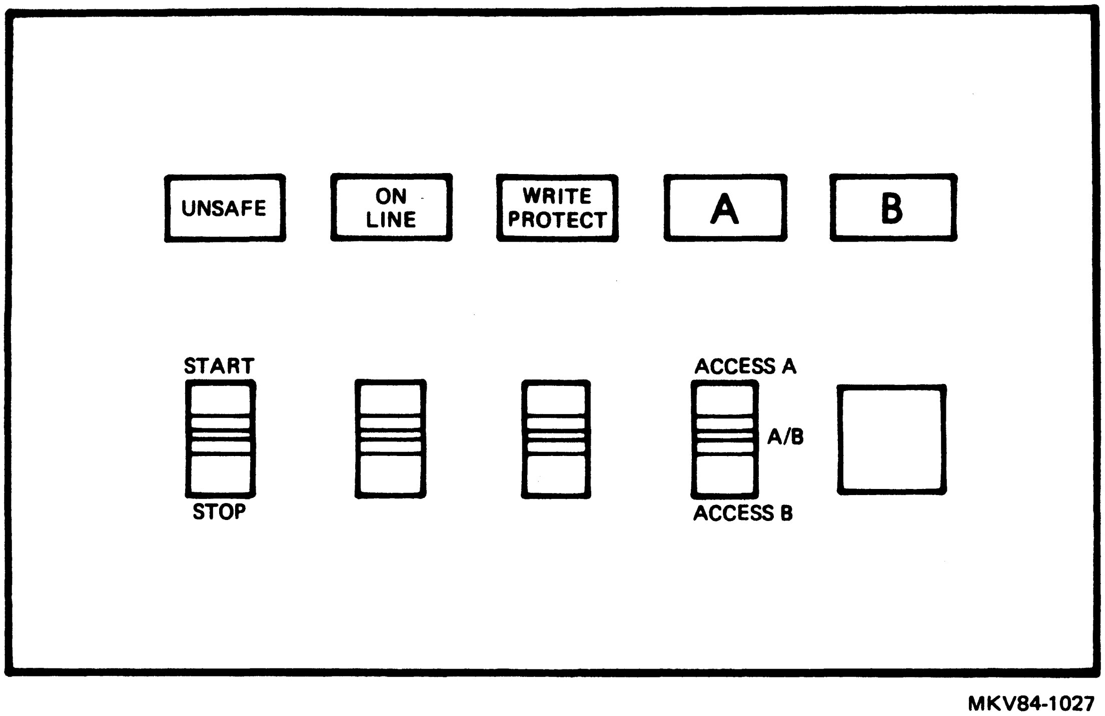


Figure 5-24 RP07 Operator Controls and Indicators

5.7.2.1 Controls -

START/STOP Switch

This is a two-position toggle switch. When placed in START position (up), it initiates the drive spindle motor start sequence. In the STOP position (down), it causes the heads to position correctly and removes power from the spindle motor.

Should a main power interruption occur and power be subsequently restored, the drive will cycle up if the START/STOP switch is still in the START position.

ONLINE Switch

The ONLINE switch is a two-position toggle switch. When this switch is in the ONLINE position (up), the RP07 drive is available to the VAX-11/785. Placing the switch in the down position causes the drive to be OFFLINE.

WRITE PROTECT Switch

This switch is a two-position toggle switch. When placed in the ON position (up), it sets the drive in a write protect mode only when a write command is not already in progress. If a write operation is in progress when the switch is moved to ON, write protection is inhibited until termination of the write command.

When the WRITE PROTECT switch is placed in the OFF (down) position, the write protect condition is immediately released.

ACCESS A, A/B, B Switch

This is a three-position toggle switch. The A (up) and the B (down) positions override the A/B (mid) position.

Two independent MASSBUS controllers may share a single RP07 drive. This dual controller option provides three states in which the drive can interface with the two controllers.

1. Connected to Controller A (up position)

   Only controller a can operate on the drive registers (except for the Attention Summary Pseudo Register and the Drive Type Register). On single access drives, "A" is forced true regardless of switch position.

2. Connected to Controller B (down position)

   Only controller B can operate on the drive registers (except for the Attention Summary Pseudo Register and the Drive Type Register).

3. Neutral (Programmable A/B - mid position)

   The drive can be accessed by either controller. Changing the position of the access switch after initialization produces no effect on the drive.

To change ports by using the ACCESS switch, do the following.

1. Place the ONLINE switch down.

2. Place the ACCESS switch in the desired position.

3. Place the ONLINE switch up.

5.7.2.2 Indicators -

Initiation of a start-spindle sequence illuminates all indicators on the operator control panel for one second as a lamp test.

UNSAFE Indicator

The UNSAFE indicator lights only when an unsafe error condition develops in the drive. The indicator goes out when the unsafe condition is reset.

An early temperature warning or an inadequate air flow will cause the UNSAFE indicator to blink until the condition is corrected or the drive shuts itself down.

ONLINE Indicator

The ONLINE indicator lights when the spindle-start sequence is completed and the drive is available to the VAX-11/785.

The ONLINE indicator goes out under any of the following conditions.

1. The ONLINE switch is placed to OFFLINE.

2. The START/STOP switch is in the STOP position.

3. The UNSAFE indicator is lighted.

4. The UNSAFE indicator is blinking.

WRITE PROTECT Indicator

The WRITE PROTECT indicator reflects the status of the switch.

ACCESS A, A/B, B Indicator

When the A indicator is lighted, it indicates that port A is being accessed by the controller. When the B indicator is on, port B is being accessed by the controller. Both indicators appearing to be lighted indicates that the drive is in the A/B state.

5.7.3 RP07 Operating Instructions

5.7.3.1 Start Up -

1. Determine if the disk is to be write protected, and place the WRITE PROTECT switch to the appropriate position.
2. Place the ACCESS A, A/B, B switch in the appropriate state.
3. Place the ONLINE switch to ONLINE.
4. Place the START/STOP switch in the up position to start.

NOTE

The spindle-start sequence completes in 30 seconds.

5.7.3.2 Shut Down -

1. Place the ONLINE switch to OFFLINE.
2. Place the START/STOP switch in the down position to stop. The spindle-stop sequence completes in 20 seconds.

5.7.4 RP07 Customer Care And Maintenance

WARNING

Hazardous voltages are present inside this equipment. Installation and servicing should be performed by a qualified and trained service representative. Bodily injury or equipment damage may result from improper servicing.

5.7.4.1 Precautions -

1. Keep pens, pencils, and all other foreign objects OFF the drive.

2. Do NOT place beverages on the drive.

CAUTION

> Beverages spilled on drive covers could filter down between covers and cause damage to drive circuitry.

3. If the drive is to be moved a long distance, or a short distance over a rough surface, it should be done only by an RP07 qualified service representative.

4. If, during drive operation, an unusual sound occurs, such as metal to metal scraping, tinging, or popping, IMMEDIATELY:

   - Place the drive OFFLINE.

   - Place the START/STOP switch in the STOP position.

   - Disconnect the main source of AC power.

   - Contact the service representative.

CAUTION

> Continued use of the drive under the circumstances listed above may result in additional damage to the drive components or assemblies or both.

5.7.4.2 Customer Preventive Maintenance Procedures -

Customer maintenance procedures for the RP07 drive consist of:

- Replacing the drive prefilter semiannually.
- External care and cleaning of drive service covers.

Prefilter Replacement

The prefilter (Figure 5-25) is a fiber type filter inserted in its housing in the blower assembly. It traps the larger airborne particles of contaminants, thus preventing them from circulating through the electronics library and absolute filter. The frequency with which the prefilter should be replaced depends upon the site environment. It is recommended that the prefilter be replaced every six months except when the site environment is less than computer room standard. The prefilter should then be replaced more often.

CAUTION

1. Do NOT use a replacement prefilter that is bent, broken, or obviously damaged.
2. Do NOT attempt to clean or reuse the prefilter. It is a throw-away item. Discard it after use.

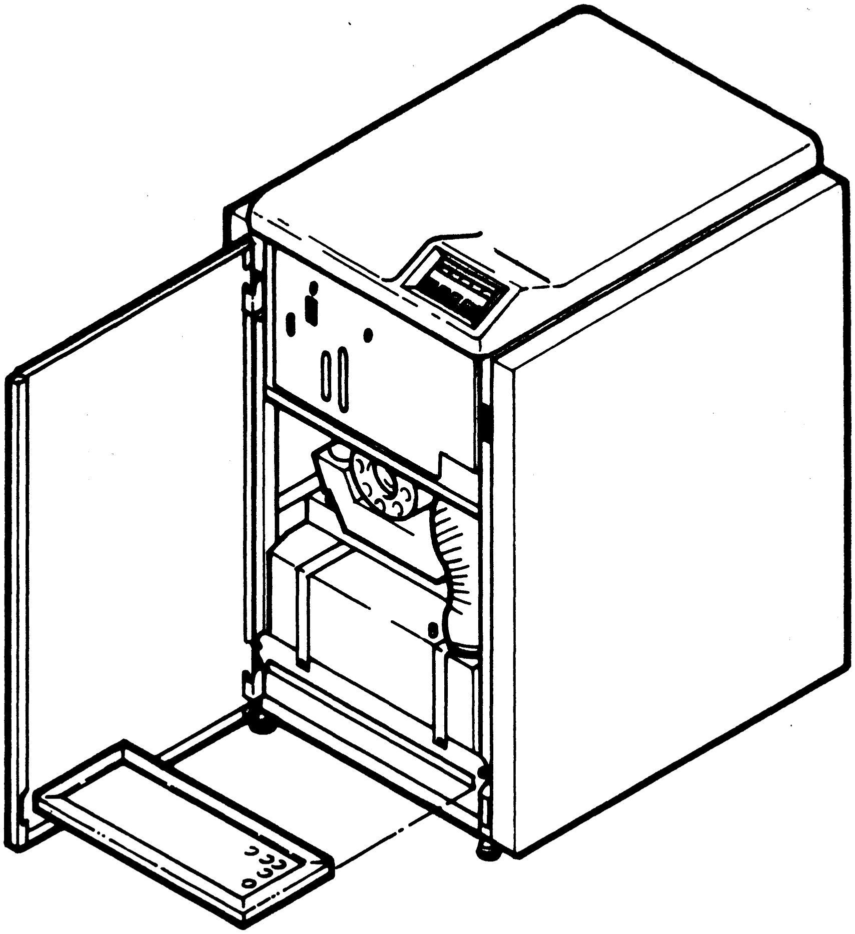


Figure 5-25 RP07 Drive Prefilter

To remove the prefilter from the RP07:

1. Open the drive front cover.

2. Slide the existing prefilter out of its housing in the blower assembly. Note the air flow arrow direction. Discard the used filter.

3. Slide the replacement prefilter into its housing in the blower assembly. Note the air flow arrow direction (arrow pointing up).

NOTE

> Insure that the replacement prefilter is installed correctly. The arrow should be pointing up toward the top of the drive, and the prefilter should be slid all the way into its housing.

4. Close the drive front cover.

Drive Covers

The RP07 drive has five service covers: front, top, rear, and two side covers. All drive service covers must be in place during drive operation, with the exception of the side covers.

NOTE

> Side covers may not always be used between drives when two or more drives are together in a string configuration.

The following procedures are recommended to insure and maintain quality and appearance of the RP07:

1. Remove fingerprints, dirt, film, or oily residue from the drive service covers with a soft lint-free cloth dampened with a cleaning solvent such as isopropyl alcohol.

WARNING

> Avoid excessive use of cleaning solvents near electrical wiring and components. Injury to personnel or damage to the equipment or both may result.

2. A damaged service cover should be replaced if it is damaged to such a degree that it will affect the operation of the drive.

5.8 TU77 MAGNETIC TAPE TRANSPORT

The TU77 magnetic tape transport records and reads data in 9-track nonreturn to zero (NRZI) or phase encoded (PE) format. Bit density is 800 bits per inch for the NRZI format and 1600 BPI for the PE format. The transport can read data in the forward or reverse direction. The read/write tape speed for both the forward and reverse directions is 125 inches per second. the nominal rewind time for a 731.5 m (2400 ft) reel is 65 seconds.

5.8.1 TU77 Specifications

Table 5-21 TU77 Specifications

Item	**Specification**
Tape (computer grade)	
Width	**12.6492 ± 0.0508 mm (0.498 ± 0.002 in)**
Thickness	**0.0381 mm (1.5 mil)**
Tape tension	**2.224 ± 0.139 Newtons (8.0 oz nominal)**
Reel diameter (autoload)	**266.7 mm (10.5 in) maximum (Note 1) and easy load cartridge #1 and #2™**
Recording modes	**1600 BPI PE** **800 BPI NRZI**
Magnetic head	**Dual stack (with erase head)**
Tape speed	**3.2 m/s (125 ips)**
Instantaneous speed variation	**±3 percent**
Long term speed variation	**±1 percent**
Rewind time (731.5 m) (2400 ft)	**65 seconds nominal (80 seconds maximum)**
Tape cleaner	**Dual-blade type connected to vacuum supply**
Interchannel displacement	
Read	**3.81 μm (150 μin) maximum (Note 2)**
Write NRZI	**5.72 μm (225 μin) maximum (Note 3)**
Write PE	**11.43 μm (450 μin) maximum**
Start time	**3.0 ± 0.3 ms**
Stop time	**3.0 ± 0.3 ms**
Start distance	**4.216 ± 0.508 mm (0.166 ± 0.02 in)**
Stop distance	**4.953 ± 0.508 mm (0.195 ± 0.02 in)**
Beginning of tape (BOT) and end of tape (EOT) detectors (Note 4)	**Photoelectric**
Tape creepage	**None**
Pneumatic interlock	**Tape motion disabled when vacuum is lost in vacuum column**
Load time	**No greater than 10 seconds without a retry, and 20 seconds with a retry for 10-1/2 inch reels**
Unload time	**Less than 7 seconds for 10-1/2 inch reels**
Write gap to read gap distance	**0.381 cm ± 0.013 cm (0.150 in ± 0.005 in)**
Weight	**288 kg (640 lbs) (master unit)**

™Easy Load #1 and #2 are trademarks of IBM.

Table 5-21 TU77 Specifications (Cont)

Item	Specification
Cabinet dimensions	
Height	152.4 cm (60.0 in)
Width	67.3 cm (26.5 in)
Depth (from face of front door to rear of cabinet)	81.9 cm (32.3 in)
Operating temperature (Electronics)	4.44° to 44.0°C (40° to 112°F) (Note 5)
Nonoperating temperature	−45.55° to 71.11°C (−50° to 160°F)
Operating altitude	0 to 2439 m (0 to 8000 ft) (Note 6)
Nonoperating altitude	15,240 m (50,000 ft) maximum
Power	
Volts ac	200, 210, 220, 230, 240, 250 (Note 7)
Frequency	50 ± 1 or 60 ± 1 Hz
Kilovolt Amp (KVA)	
Standby (loaded)	1.3 KVA maximum (Note 8)
Start/stop	2.1 KVA maximum (Note 8)
Electronics	All silicon

NOTES

1. 177.8 mm (7 in) and 216.0 mm (8.5 in) reels may be used but cannot be autoloaded.

2. This is the maximum displacement between any two bits of a character when reading a master tape using the read section of the read-after-write head.

3. This is the maximum displacement between any two bits of a character on a tape written with all ones using the write section of the read-after-write head.

4. The approximate distance from the detection area to the write head gap is 35.6 mm (1.40 in).

5. For data transfer, the operating temperature depends on the nature of the tape material.

6. Operation above 610 m (2000 ft), in 610 m (2000 ft) increments requires installation of high altitude pulleys and belts in the TU77 power pack.

7. Line variations must be within +10 percent.

8. Slave unit only. Add 625 W for master units.

5.8.2 TU77 Controls And Indicators

The TU77 controls and indicators are on the transport control panel. The controls and indicators are shown in Figure 5-26 and listed in Tables 5-22 and 5-23.

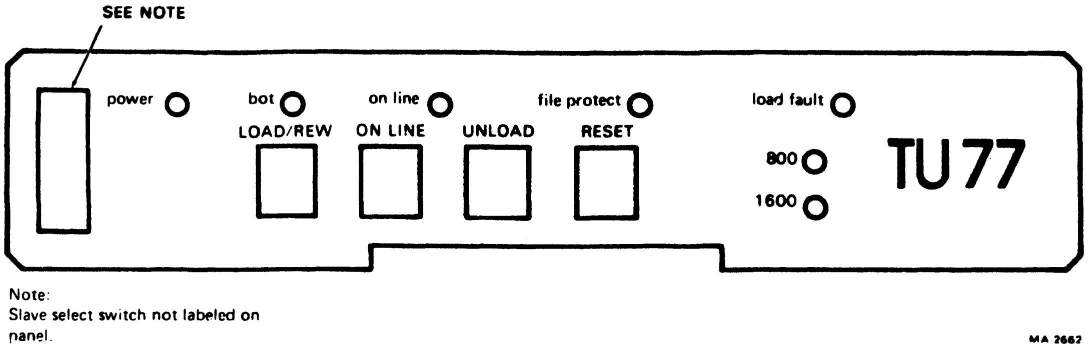


Figure 5-26 TU77 Control Panel

Table 5-22 TU77 Controls

Control	Function
Slave Select switch (unlabeled)	Selects the address (slave number 0 to 3) of the tape transport.
LOAD/REW	Pressing and releasing the LOAD/REW button will initiate one of three sequences. 1. With no tape in path, a load sequence is initiated. 2. With tape in path but not tensioned, a mid-reel load sequence is initiated. In a mid-reel load sequence the tape will load and run in the reverse direction to BOT. 3. With tape in path and tensioned, and the transport off-line, the tape rewinds to BOT. If the tape is already at BOT or if the transport is on-line, no action occurs.
ON LINE	Pressing and releasing the ON-LINE button will change the transport from off-line to on-line. Pressing and releasing the button again will change the transport from on-line to off-line.
UNLOAD	If the TU77 is off-line, pressing and releasing the UNLOAD button causes the tape to rewind and unload. If the tape is already at BOT, it will unload. If the TU77 is on-line, the UNLOAD button has no effect.
RESET	Pressing and releasing the RESET button terminates all functions and clears a load fault.

Table 5-23 TU77 Indicators

Indicator	Function
power	Indicates presence of dc and secondary ac power.
bot	Indicates tape is at BOT.
on line	Indicates the TU77 is on-line. The transport will return to off-line if any of the following occur. 1. ON LINE button is pressed. 2. An external rewind unload command is received. 3. Vacuum column interlock is broken. 4. AC power is lost. 5. RESET button is pressed.
file protect	Indicates that a reel of tape without a write-enable ring has been loaded onto the transport.
load fault	Lamp flashes when a load fault has occurred; there are two causes. 1. The autoload sequence has failed to load a tape from a 267 mm (10-1/2 in) reel after two trys. 2. A load sequence has failed to load tape from a 216 or 178 mm (8-1/2 or 7 in) reels.
800	Indicates the tape transport is set to read or write at 800 BPI (NRZI mode).
1600	Indicates the tape transport is set to read or write at 1600 BPI (PE mode).

5.8.3 TU77 Operating Instructions

5.8.3.1 Power -

Power is applied to the TU77 when the VAX-11/785 is turned on. The 861 power control is remotely enabled from the VAX-11/785 and applies ac power from the power source to the TU77 cabinet.

5.8.3.2 Auto-load/Manual Load Selection -

Figure 5-27 shows various conditions and interlock settings that affect the selection of an auto-load or manual load procedure.

Loading of a 216 or 178 mm (8-1/2 or 7 in) reel requires the manual procedure. If a 267 mm (10-1/2 in) reel in a cartridge is loaded, the cartridge will be engaged. The interlock bypasses the auto-load interlock and causes the auto-load procedure to be used.

A full reel of tape is needed for an auto-load sequence. The last turn of tape must be between 1.59 and 0.64 cm (5/8 and 1/4 in) from the outer edge of the reel (Figure 5-28). Large 276 mm (10-1/2 in) reels not containing sufficient tape to load automatically must be loaded manually.

If the tape is not in a cartridge, the autoload procedure is followed if the transport front door is closed. When the front door is open, the auto-load procedure will be used only if the auto-load interlock is pulled out. Otherwise, the manual procedure must be followed.

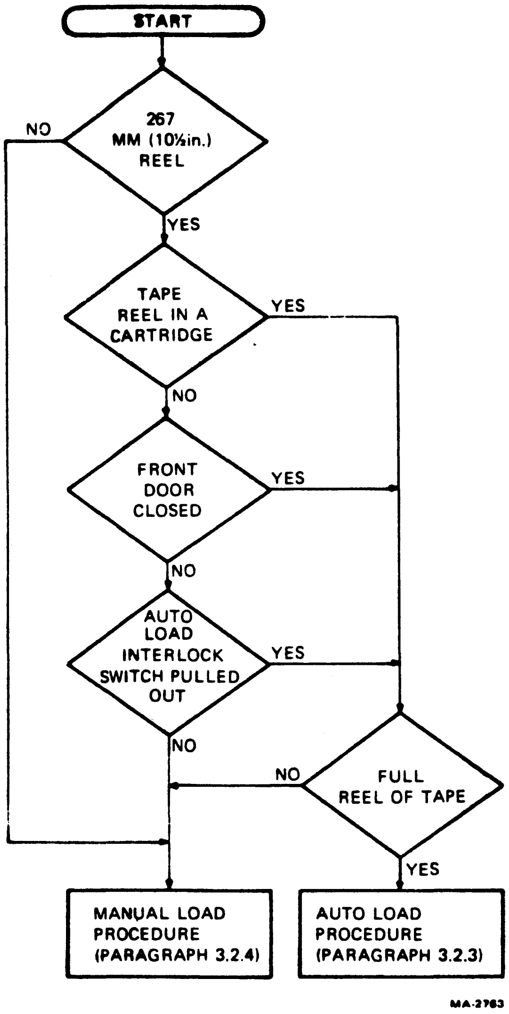


Figure 5-27 TU77 Auto-load/Manual Load Selection

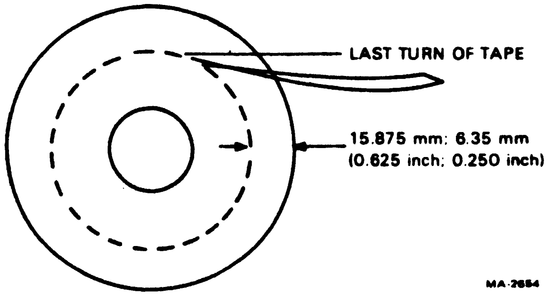


Figure 5-28 TU77 Tape Content Limit

5.8.3.3 Auto-load -

1. Check that the tape path is clean.
2. Check that the tape reel is 267 mm (10-1/2 in) and that it has a full reel of tape.
3. Check that the end of tape has a clean edge and is not bent, torn, or frayed. If necessary, trim the end with the tape crimper (P/N 47-00038).

NOTE

> Do not crimp the tape unnecessarily. Each crimping shortens the tape leader and could eventually lead to failures in the auto-load sequence.

4. Check that a static charge is not causing the end of tape to stick to the reel. The end of tape must be free of the reel to accomplish an auto-load.
5. If a write operation will be performed, install a write enable ring onto the rear flange of the tape reel.
6. Place the supply reel in position on the upper hub, rotate until it slips easily into place, and press the reel-retaining actuator. The reel should be positioned so that the tape will unwind if the reel is turned clockwise.
7. Carefully close the buffer door, making sure the door is closed securely.
8. Close the transport front door.

NOTE

> The transport front door must be shut to close the auto-load interlock and enable the auto-load sequence. If you want to leave the door open, override the interlock by pulling the switch out.

9. Check that power is applied to the tape transport (power indicator on).

10. Follow the instructions and check for the events specified in Figure 5-28. If the tape fails to load properly, the figure contains some operator troubleshooting hints you can try before calling maintenance personnel. Refer to Figure 5-30 to identify tape path components.

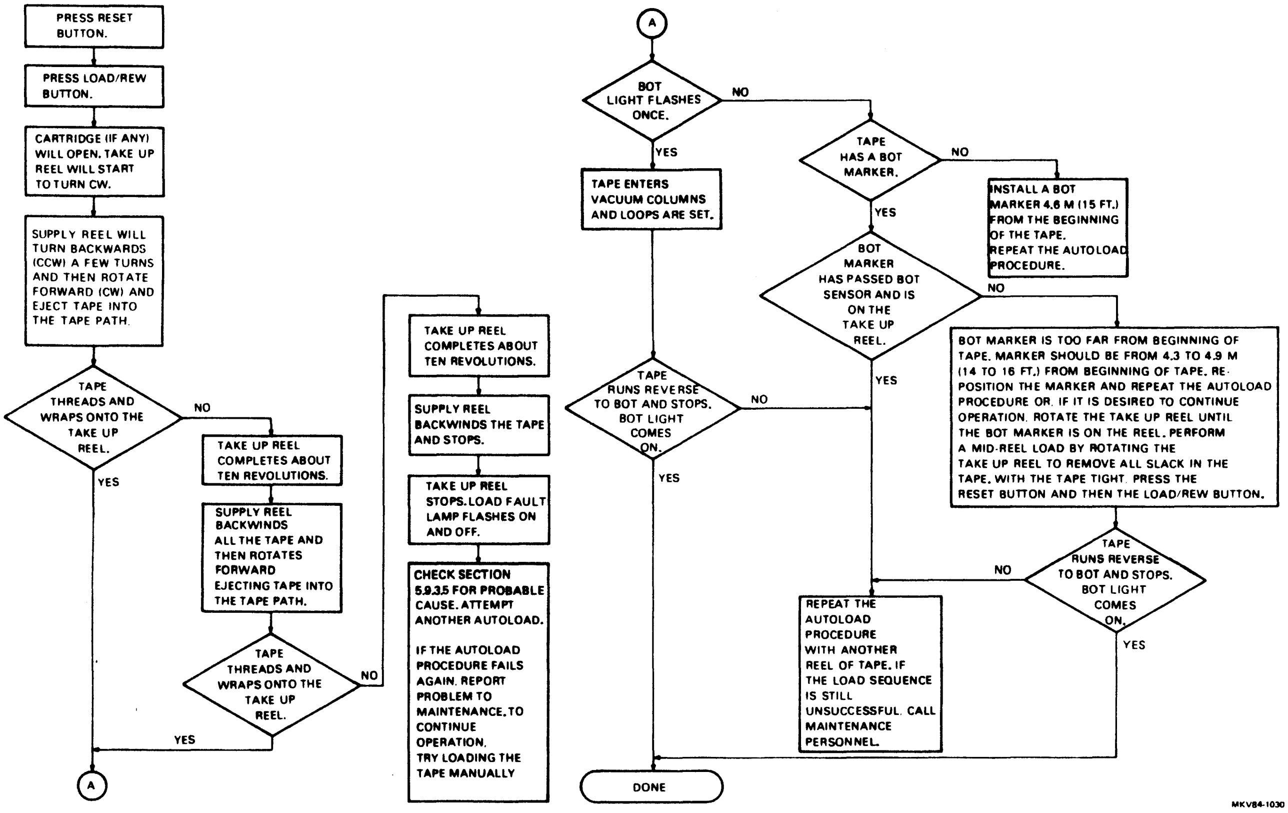

PRESS RESET BUTTON.
PRESS LOAD/REW BUTTON.
CARTRIDGE (IF ANY) WILL OPEN. TAKE UP REEL WILL START TO TURN CW.
SUPPLY REEL WILL TURN BACKWARDS (CCW) A FEW TURNS AND THEN ROTATE FORWARD (CW) AND EJECT TAPE INTO THE TAPE PATH.
TAPE THREADS AND WRAPS ONTO THE TAKE UP REEL.
NO
YES
TAKE UP REEL COMPLETES ABOUT TEN REVOLUTIONS.
SUPPLY REEL BACKWINDS ALL THE TAPE AND THEN ROTATES FORWARD EJECTING TAPE INTO THE TAPE PATH.
TAPE THREADS AND WRAPS ONTO THE TAKE UP REEL.
NO
YES
A
TAKE UP REEL COMPLETES ABOUT TEN REVOLUTIONS.
SUPPLY REEL BACKWINDS THE TAPE AND STOPS.
TAKE UP REEL STOPS. LOAD FAULT LAMP FLASHES ON AND OFF.
CHECK SECTION 5.9.3.5 FOR PROBABLE CAUSE. ATTEMPT ANOTHER AUTOLOAD.
IF THE AUTOLOAD PROCEDURE FAILS AGAIN, REPORT PROBLEM TO MAINTENANCE. TO CONTINUE OPERATION, TRY LOADING THE TAPE MANUALLY
A
BOT LIGHT FLASHES ONCE.
NO
YES
TAPE ENTERS VACUUM COLUMNS AND LOOPS ARE SET.
TAPE RUNS REVERSE TO BOT AND STOPS. BOT LIGHT COMES ON.
NO
YES
DONE
TAPE HAS A BOT MARKER.
NO
YES
INSTALL A BOT MARKER 4.6 M (15 FT.) FROM THE BEGINNING OF THE TAPE. REPEAT THE AUTOLOAD PROCEDURE.
BOT MARKER HAS PASSED BOT SENSOR AND IS ON THE TAKE UP REEL.
NO
YES
BOT MARKER IS TOO FAR FROM BEGINNING OF TAPE. MARKER SHOULD BE FROM 4.3 TO 4.9 M (14 TO 16 FT.) FROM BEGINNING OF TAPE. RE-POSITION THE MARKER AND REPEAT THE AUTOLOAD PROCEDURE OR, IF IT IS DESIRED TO CONTINUE OPERATION, ROTATE THE TAKE UP REEL UNTIL THE BOT MARKER IS ON THE REEL. PERFORM A MID-REEL LOAD BY ROTATING THE TAKE UP REEL TO REMOVE ALL SLACK IN THE TAPE. WITH THE TAPE TIGHT, PRESS THE RESET BUTTON AND THEN THE LOAD/REW BUTTON.
TAPE RUNS REVERSE TO BOT AND STOPS. BOT LIGHT COMES ON.
NO
YES
REPEAT THE AUTOLOAD PROCEDURE WITH ANOTHER REEL OF TAPE. IF THE LOAD SEQUENCE IS STILL UNSUCCESSFUL, CALL MAINTENANCE PERSONNEL.
MKV84-1030

Figure 5-29 TU77 Auto-load Sequence

5.8.3.4 Manual Load -

1. Check that the tape path is clean.

2. If a write operation will be performed, install a write enable ring onto the rear flange of the tape reel.

3. Place the supply reel in position on the upper hub. Rotate the reel until it slips into place, and press the reel-retaining actuator. The reel should be positioned so that the tape will unwind if the reel is turned clockwise.

4. Place the tape leader between thread block 1 and the air bearing (Figure 5-30).

NOTE

Ensure that there is no tape slack or sag between the supply reel and thread block 1.

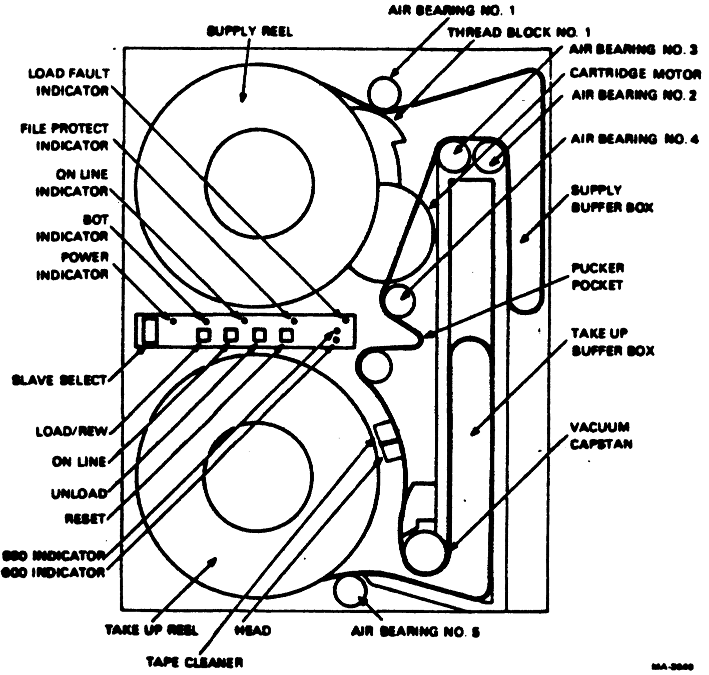


Figure 5-30 TU77 Tape Path and Controls

5. Carefully close the buffer door, making sure the door is closed securely.

6. If loading a 216 or 178 mm (8-1/2 or 7 in) reel, close the transport front door. If the reel is 267 mm (10-1/2 in), leave the transport door open; this disables the auto-load sequence that would normally occur with a 267 mm reel.

7. Check that power indicator is on, indicating that power is applied to the tape transport.

8. Follow the instructions and check for the events specified in Figure 5-31. If the tape fails to load properly, the figure contains some operator troubleshooting hints that can be tried before maintenance personnel are called. Refer to Figure 5-30 to identify tape path components.

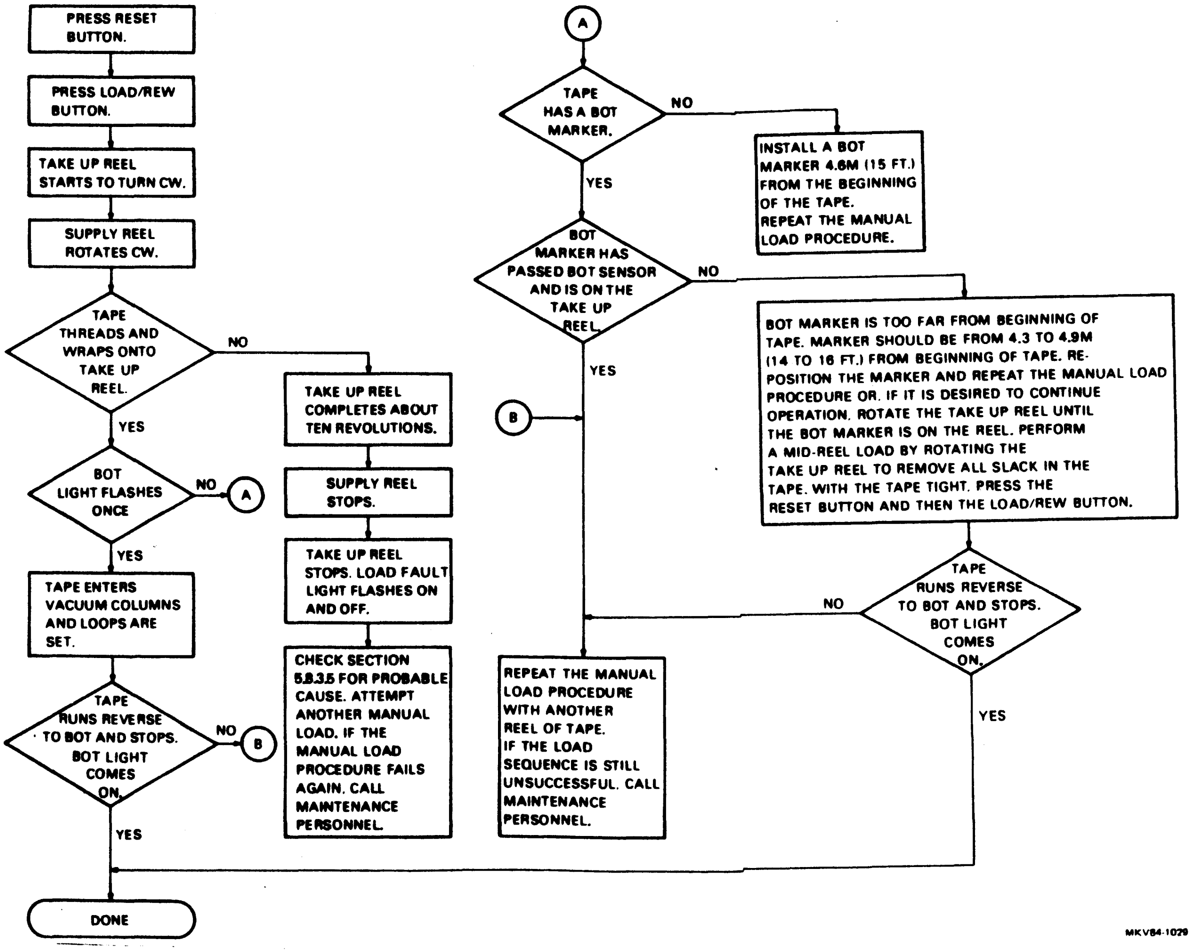

PRESS RESET BUTTON.
PRESS LOAD/REW BUTTON.
TAKE UP REEL STARTS TO TURN CW.
SUPPLY REEL ROTATES CW.
TAPE THREADS AND WRAPS ONTO TAKE UP REEL.
NO
YES
TAKE UP REEL COMPLETES ABOUT TEN REVOLUTIONS.
SUPPLY REEL STOPS.
TAKE UP REEL STOPS. LOAD FAULT LIGHT FLASHES ON AND OFF.
CHECK SECTION 5.8.3.5 FOR PROBABLE CAUSE. ATTEMPT ANOTHER MANUAL LOAD. IF THE MANUAL LOAD PROCEDURE FAILS AGAIN, CALL MAINTENANCE PERSONNEL.
BOT LIGHT FLASHES ONCE
NO
A
YES
TAPE ENTERS VACUUM COLUMNS AND LOOPS ARE SET.
TAPE RUNS REVERSE TO BOT AND STOPS. BOT LIGHT COMES ON.
NO
B
YES
DONE
A
TAPE HAS A BOT MARKER.
NO
INSTALL A BOT MARKER 4.6M (15 FT.) FROM THE BEGINNING OF THE TAPE. REPEAT THE MANUAL LOAD PROCEDURE.
YES
BOT MARKER HAS PASSED BOT SENSOR AND IS ON THE TAKE UP REEL.
NO
BOT MARKER IS TOO FAR FROM BEGINNING OF TAPE. MARKER SHOULD BE FROM 4.3 TO 4.9M (14 TO 16 FT.) FROM BEGINNING OF TAPE. RE-POSITION THE MARKER AND REPEAT THE MANUAL LOAD PROCEDURE OR. IF IT IS DESIRED TO CONTINUE OPERATION. ROTATE THE TAKE UP REEL UNTIL THE BOT MARKER IS ON THE REEL. PERFORM A MID-REEL LOAD BY ROTATING THE TAKE UP REEL TO REMOVE ALL SLACK IN THE TAPE. WITH THE TAPE TIGHT. PRESS THE RESET BUTTON AND THEN THE LOAD/REW BUTTON.
YES
B
TAPE RUNS REVERSE TO BOT AND STOPS. BOT LIGHT COMES ON.
NO
YES
REPEAT THE MANUAL LOAD PROCEDURE WITH ANOTHER REEL OF TAPE. IF THE LOAD SEQUENCE IS STILL UNSUCCESSFUL, CALL MAINTENANCE PERSONNEL.
MKV84-1029

Figure 5-31 TU77 Manual Load Sequence

5.8.3.5 Probable Causes Of Load Failure -

1. Check that the buffer door is closed and sealed properly.

2. Check for ripped, creased, or damaged tape leader. This is especially critical if an auto-load failure has occurred. If necessary, crimp the end of the tape with a tape crimper.

3. Check that the write-enable ring is inserted in the tape reel if a write operation will be performed. The file protect indicator should be off.

4. If doing a manual load with a 267 mm (10-1/2 in) reel of tape, check that the transport front door is open and the auto-load interlock switch is not pulled out.

5. If doing an auto-load, check these items.

   a. A 267 mm (10-1/2 in) reel of tape is on the supply hub.

   b. The supply reel has a wrap around cartridge. If it does not, check that the transport door is closed. If the door is open, check that the interlock switch is pulled out.

   c. The supply reel has a full reel of tape. The tape must be between 0.64 and 1.59 cm (1/4 and 5/8 in) from the outer edge of the reel.

5.8.3.6 Mid-Reel Load -

A mid-reel load is usually required after a power failure or a lost interlock.

1. Rotate the take-up reel to remove all slack in the tape.

NOTE

> Do not attempt mid-reel load until all slack is manually removed from the tape between the reels.

2. Press the RESET button.

3. Press the LOAD/REW button.

The transport loads tape and starts a reverse operation to BOT. To stop the transport, press the RESET button. Any other command may now be given.

5.8.3.7 Unload -

1. Press the RESET button to terminate any current operation and place the transport off-line.

2. Press the UNLOAD button. If the tape is at BOT, it will backwind onto the supply reel. If the tape is at mid-reel, it will rewind to BOT and then unload (backwind onto the supply reel).

5.8.3.8 Rewind -

If the LOAD/REW button is pressed when the tape is tensioned and the transport is off-line, the tape rewinds to BOT. If the tape is at BOT when the button is pressed, no action occurs.

5.8.3.9 On-Line/Off-Line -

Pressing the ON LINE button places the transport on-line and lights the on-line indicator. The transport can accept external commands if it is on-line, selected, and ready. Pressing the ON LINE button again will cause the transport to go off-line, turning off the on-line indicator.

The transport goes off-line if any of the following occur.

- An external REWIND UNLOAD command is received.
- The vacuum column interlock is broken.
- AC power is lost.
- The RESET button is pressed.

5.8.4 Customer Care And Maintenance

5.8.4.1 Customer Responsibilities -

1. obtaining operating supplies, including magnetic tape and cleaning supplies.
2. Supplying accessories, including cabinetry, tables, and chairs.
3. Maintaining the required logs and report files consistently and accurately.
4. Making the necessary documentation available in a location convenient to the system.
5. Keeping the exterior of the system and the surrounding area clean.
6. Ensuring that ac plugs are securely plugged in each time equipment is used.
7. Performing the magnetic tape care and preventive maintenance operations described below.

5.8.4.2 Magnetic Tape Care -

1. Do not expose magnetic tape to excessive heat or dust. Most tape read errors are caused by dust or dirt on the read head. The tape must be kept clean.

2. Always store tape reels inside containers when not in use. Keep the empty containers tightly closed to keep out dust and dirt.

3. Never touch the portion of tape between the BOT and EOT markers. Oil from fingers attracts dust and dirt.

4. Never use a contaminated reel of tape. A contaminated reel spreads dirt to clean tape reels and could have an adverse effect on tape transport reliability.

5. Always handle tape reels by the hub hole. Squeezing the reel flanges could lead to tape edge damage in winding or unwinding tapes.

6. Do not smoke near the tape transport or storage area. Tobacco smoke and ash are especially damaging to tapes.

7. Do not place magnetic tape near any line printer or other device that produces paper dust.

8. Do not place magnetic tape on top of the tape transport, or in any other location where it might be affected by hot air.

9. Do not store magnetic tape near electric motors.

5.8.4.3 Preventive Maintenance -

1. Dismount the tape from the unit.

2. Clean the following components of the transport using a foam-tipped swab soaked in cleaning fluid. Daily cleaning is recommended for TU77 units in constant operation. See Figure 5-32.

   Read/write head
   Erase head
   Tape cleaner
   Air bearings (6)
   Spring-loaded ceramic tape guides (2)
   Thread blocks (6)
   Capstan
   Buffer boxes (2)
   Pucker pocket
   Buffer box glass, pocket glass, door overlay

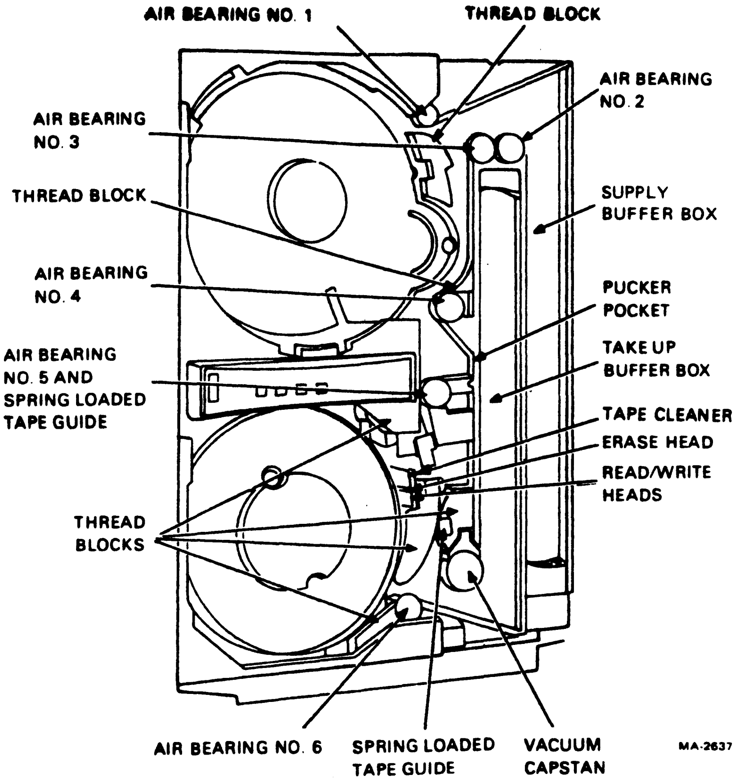
NOTE

Use only foam-tipped swabs. Do not touch any part of the tape path since oil from fingers attracts dust and dirt. Also, excessive physical pressure on the capstan could affect its alignment.

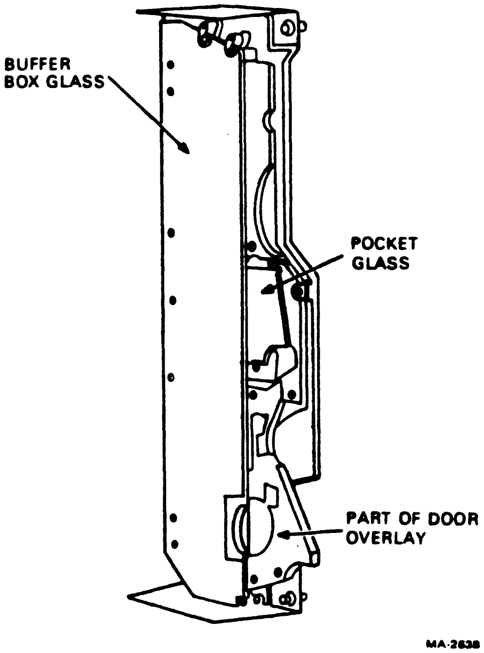
3. When cleaning the thread blocks, be sure to clean the air guide ports. Every thread block has air guide ports except the metal block containing the spring-loaded tape guide.

4. When cleaning the spring-loaded ceramic guides, be sure that the washer is pressed firmly against the tape guide surface and not hung up on its shaft.

5. When cleaning the inner surface of the vacuum door, use a lint-free wipe and cleaning fluid. Pass another lint-free wipe over the head using a polishing action to remove any remaining deposits.



a. Tape Path Items



b. Buffer Box Door

Figure 5-32 TU77 Transport Items to Clean Daily

5.9 TU78 MAGNETIC TAPE TRANSPORT

The TU78 magnetic tape transport writes and reads data in 9-track phase-encoded (PE) or group code recording (GCR) format. Bit density is 1600 bits per inch for the PE format and 6250 bits/in for the GCR format. The transport can read data in the forward or reverse direction. The read/write tape speed for both directions is 125 inches per second. Nominal rewind time for a 731.5 m (2400 ft) reel is 65 seconds.

5.9.1 TU78 Specifications

Table 5-24 TU78 Specifications

Tape (computer grade)	
Width	**12.6492 mm ± 0.0508 mm (0.498 in ± 0.002 in)**
Thickness	**0.0381 mm (1.5 mil)**
Tape tension	**2.780 N (10.0 oz +0.0/−1.0 oz) nominal**
Reel diameter (autoload)	**266.7 mm (10.5 in) maximum (note 1) and Easy-Load cartridge 1 and 2**
Recording modes	**1600 bits/in (3200 FCI) PE** **6250 bits/in (9042 FCI) GCR**
Magnetic head	**Dual stack (with erase head)**
Tape speed	**3.2 m/s (125 in/s)**
Instantaneous speed variation	**±3 percent**
Long term speed variation	**±1 percent (forward)** **±2 percent (reverse)**
Rewind time for 731.5 m (2400 ft) tape	**65 seconds nominal (80 seconds maximum)**
Tape cleaner	**Dual blade type connected to vacuum supply**
Interchannel displacement error	
Read	**3.81 μm (150 μin) maximum (note 2)**
Read-after-write	**8.89 μm (350 μin) maximum (note 3)**
Start/stop time	**1.1 ms ± 0.1 ms**
Start	**1.3 ms ± 0.1 ms**
Stop	**1.1 ms ± 0.1 ms**
Start distance	**1.91 mm +0.25/−0.38 mm (0.075 in +0.010/−0.015 in)**
Stop distance	**2.16 mm +0.25/−0.38 mm (0.85 in +0.010/−0.015 in)**

Table 5-24 TU78 Specifications (Cont)

Beginning of tape (BOT) and end of tape (EOT) detectors (note 4)	Photoelectric
Tape creepage	None
Pneumatic interlock	Tape motion disabled when vacuum is lost in vacuum column
Load time	No greater than 10 seconds without a retry, and 20 seconds with a retry for 10.5 in reels
Unload time	Less than 7 seconds for 10.5 in reels
Write gap to read gap distance	0.381 cm ± 0.013 cm (0.150 in ± 0.005 in)
Weight	286 kg (630 lbs) (master unit)
Cabinet dimensions	
Height	152.4 cm (60.0 in)
Width	67.3 cm (26.5 in)
Depth (from face of front door to rear of cabinet)	81.9 cm (32.3 in)
Operating temperature	15° to 32° C (59° to 90° F) (note 5)
Nonoperating temperature	−46° to 71° C (−50° to 106° F)
Operating altitude	0 m to 2134 m (0 ft to 7000 ft) (note 6)
Nonoperating altitude	9100 m (30,000 ft) maximum
Power	
Volts ac	188 V to 264 V at 60 Hz 198 V to 264 V at 50 Hz
Frequency	50 Hz ± 1 Hz or 60 Hz ± 1 Hz
Kilovolt amp (kVA)	
Standby (loaded)	1.5 kVA maximum (note 7)
Start/stop	2.4 kVA maximum (note 7)

NOTES

1. **177.8 mm (7 in) and 216.0 mm (8.5 in) reels may be used but cannot be autoloaded.**

2. **The maximum displacement between any two bits of a character when reading a master tape using the read section of the read-after-write head.**

3. **The maximum displacement between any two bits of a character on a tape written with all ones using the write section of the read-after-write head.**

4. **Approximate distance from detection area to write head gap is 35.6 mm (1.40 in).**

5. **For data transfer, the operating temperature is dictated by the nature of the tape material.**

6. **Operation above 610 m (2000 ft), in 610 m (2000 ft) increments requires installation of high altitude pulleys and belts in TU78 power pack.**

7. **Slave unit onlv. Add 700 W for master units.**

5.9.2 TU78 Controls And Indicators

The TU78 operator controls and indicators are located on the control panel (Figure 5-33) on the front of the tape transport. The controls are described in Table 5-25, the indicators in Table 5-26.

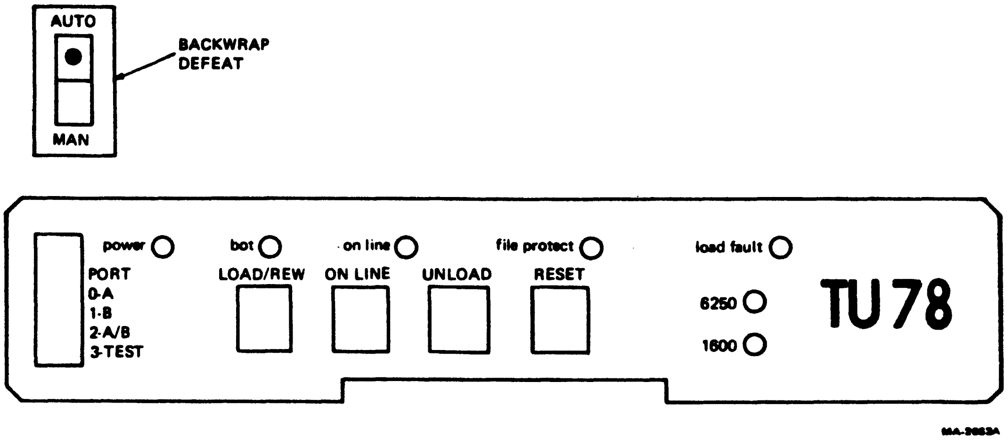


Figure 5-33 TU78 Control Panel

Table 5-25 TU78 Controls

Control	Function
Port Select Switch	Selects the MASSBUS I/O port(s) allowed to send commands to this tape transport. **Switch Position** — **Function** 0 — Transport connected to MASSBUS port A 1 — Transport connected to MASSBUS port B 2 — Transport connected to both MASSBUS ports A and B 3 — Transport disconnected from both MASSBUS ports and placed in maintenance mode (available to TM78 formatter maintenance panel)
LOAD/REW	Initiates one of three sequences. 1. With no tape in path, initiates a load sequence. 2. With tape in path but not tensioned, initiates a midreel load sequence. In a midreel load sequence the tape loads and runs in reverse direction to BOT. 3. With tape in path and tensioned, and the transport off-line, the tape rewinds to BOT. If the tape is already at BOT or if the transport is on-line, no action occurs.
ON LINE	Switches the transport off-line or on-line.
UNLOAD	If the TU78 is off-line, causes the tape to rewind and unload. If the tape is already at BOT, it unloads. If the TU78 is on-line, button has no effect.
RESET	Terminates all functions and clears a load fault.
BACKWRAP DEFEAT	AUTO (automatic) position: Allows a backwrap and retry for 10.5 inch reel, with or without a cartridge MAN (manual) position: Inhibits a backwrap and retry for 10.5 inch reel without a cartridge

Table 5-26 TU78 Indicators

Indicator	Meaning
Power	Presence of dc and secondary ac power.
BOT	Tape is at BOT.
On Line	TU78 is on-line. The transport reverts to the off-line mode if any of the following occur.
	1. ON LINE button is pressed.
	2. External rewind unload command is received.
	3. Vacuum column interlock is broken.
	4. AC power is lost.
	5. RESET button is pressed.
File Protect	Tape reel without a write enable ring has been loaded onto the transport.
Load Fault	Load fault has occurred.
	1. Autoload sequence has failed to load a tape from a 267 mm (10.5 in) reel after two tries.
	2. Load sequence has failed to load tape from a 216 mm or 178 mm (8.5 in or 7 in) reels.
1600	Tape transport set to read or write at 1600 bits/in (PE mode).
6250	Tape transport set to read or write at 6250 bits/in (GCR mode).

5.9.3 TU78 Operating Instructions

5.9.3.1 Power -

When the VAX-11/785 is turned on, power is applied to the TU78. The VAX-11/785 remotely enables the 874-E power control, which in turn applies ac power to the TU78 cabinet.

5.9.3.2 Auto-load/Manual Load Selection -

Figure 5-34 illustrates various conditions and interlock settings that determine the selection of an auto-load or manual load procedure.

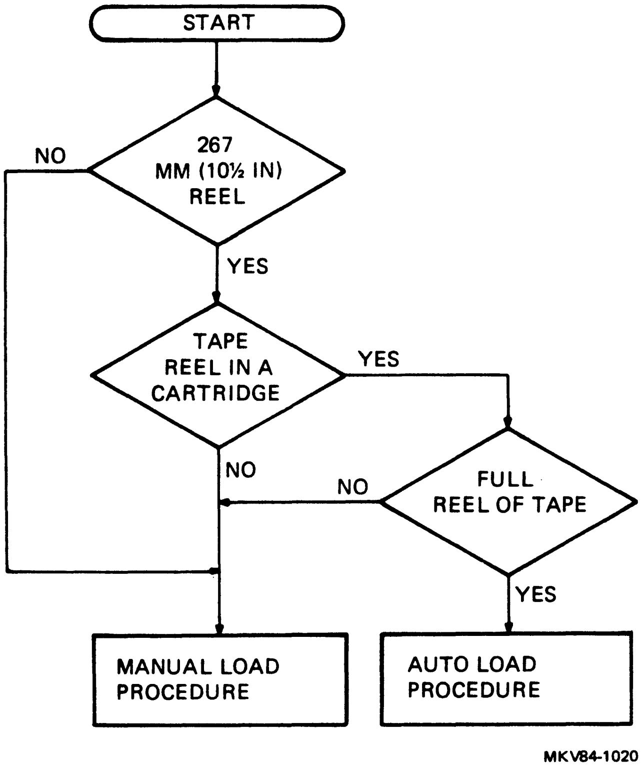


Figure 5-34 TU78 Auto-load/Manual Load Selection

5.9.3.3 Auto-load Procedure -

1. Check that the tape path is clean.

2. Check that the tape is a 267 mm (10.5 in) reel, and that it is a full reel of tape. See Figure 5-35.

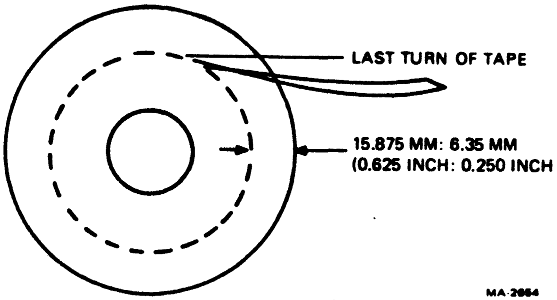


Figure 5-35 TU78 Tape Content Limit

3. Check that the end of the tape has a clean edge and is not bent, torn, or frayed. If necessary, trim the end with the tape crimper (P/N 47-00038).

NOTE

> Do not crimp the tape unnecessarily. Each crimping shortens the tape leader and could eventually lead to failures in the auto-load sequence.

4. Check that static charge is not causing the end of the tape to stick to the reel. The end of the tape must be free of the reel to accomplish auto-load.

5. If a write operation is to be performed, install a write enable ring onto the rear flange of the tape reel.

6. Place the supply reel in position on the upper hub. Rotate the reel until it slips easily into place, and press the reel retaining actuator. The reel should be positioned so that the tape unwinds if it is turned clockwise.

7. Carefully close the buffer door. Make sure it is closed securely.

8. Close the transport front door.

9. Check that the tape transport power light is on.

10. Follow the instructions and check for the events specified in Figure 5-36. If the tape fails to load properly, the figure contains some operator troubleshooting hints that can be tried before maintenance personnel are called. Refer to Figure 5-37 to identify tape path components.

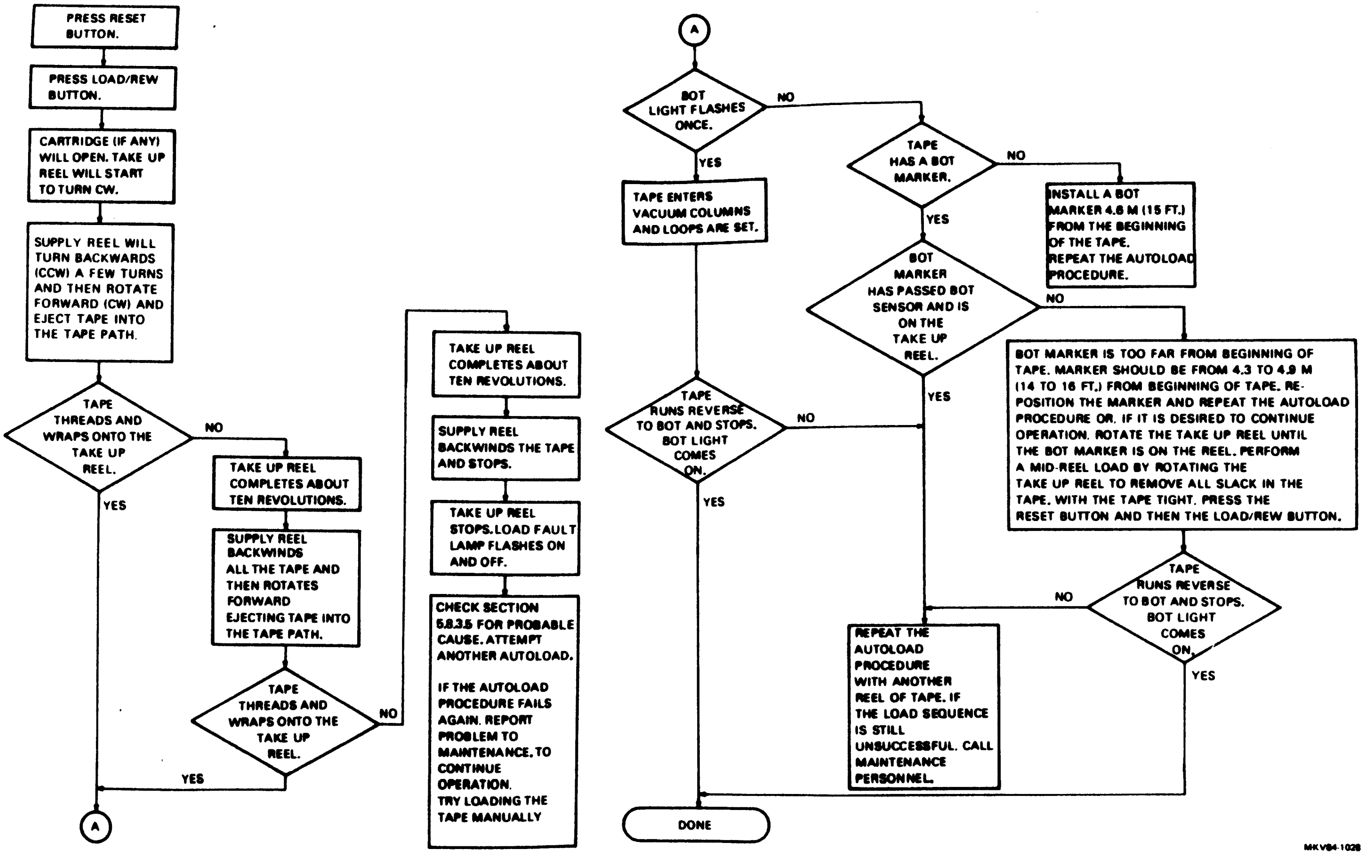

PRESS RESET BUTTON.
PRESS LOAD/REW BUTTON.
CARTRIDGE (IF ANY) WILL OPEN. TAKE UP REEL WILL START TO TURN CW.
SUPPLY REEL WILL TURN BACKWARDS (CCW) A FEW TURNS AND THEN ROTATE FORWARD (CW) AND EJECT TAPE INTO THE TAPE PATH.
TAPE THREADS AND WRAPS ONTO THE TAKE UP REEL.
NO
YES
TAKE UP REEL COMPLETES ABOUT TEN REVOLUTIONS.
SUPPLY REEL BACKWINDS ALL THE TAPE AND THEN ROTATES FORWARD EJECTING TAPE INTO THE TAPE PATH.
TAPE THREADS AND WRAPS ONTO THE TAKE UP REEL.
NO
YES
A
TAKE UP REEL COMPLETES ABOUT TEN REVOLUTIONS.
SUPPLY REEL BACKWINDS THE TAPE AND STOPS.
TAKE UP REEL STOPS. LOAD FAULT LAMP FLASHES ON AND OFF.
CHECK SECTION 5.8.3.5 FOR PROBABLE CAUSE. ATTEMPT ANOTHER AUTOLOAD.
IF THE AUTOLOAD PROCEDURE FAILS AGAIN, REPORT PROBLEM TO MAINTENANCE. TO CONTINUE OPERATION. TRY LOADING THE TAPE MANUALLY
A
BOT LIGHT FLASHES ONCE.
NO
YES
TAPE ENTERS VACUUM COLUMNS AND LOOPS ARE SET.
TAPE RUNS REVERSE TO BOT AND STOPS. BOT LIGHT COMES ON.
NO
YES
DONE
TAPE HAS A BOT MARKER.
NO
YES
INSTALL A BOT MARKER 4.6 M (15 FT.) FROM THE BEGINNING OF THE TAPE. REPEAT THE AUTOLOAD PROCEDURE.
BOT MARKER HAS PASSED BOT SENSOR AND IS ON THE TAKE UP REEL.
NO
YES
BOT MARKER IS TOO FAR FROM BEGINNING OF TAPE. MARKER SHOULD BE FROM 4.3 TO 4.9 M (14 TO 16 FT.) FROM BEGINNING OF TAPE. REPOSITION THE MARKER AND REPEAT THE AUTOLOAD PROCEDURE OR, IF IT IS DESIRED TO CONTINUE OPERATION, ROTATE THE TAKE UP REEL UNTIL THE BOT MARKER IS ON THE REEL. PERFORM A MID-REEL LOAD BY ROTATING THE TAKE UP REEL TO REMOVE ALL SLACK IN THE TAPE. WITH THE TAPE TIGHT, PRESS THE RESET BUTTON AND THEN THE LOAD/REW BUTTON.
TAPE RUNS REVERSE TO BOT AND STOPS. BOT LIGHT COMES ON.
NO
YES
REPEAT THE AUTOLOAD PROCEDURE WITH ANOTHER REEL OF TAPE. IF THE LOAD SEQUENCE IS STILL UNSUCCESSFUL. CALL MAINTENANCE PERSONNEL.
MKV84-1028

Figure 5-36 TU78 Auto-load Sequence

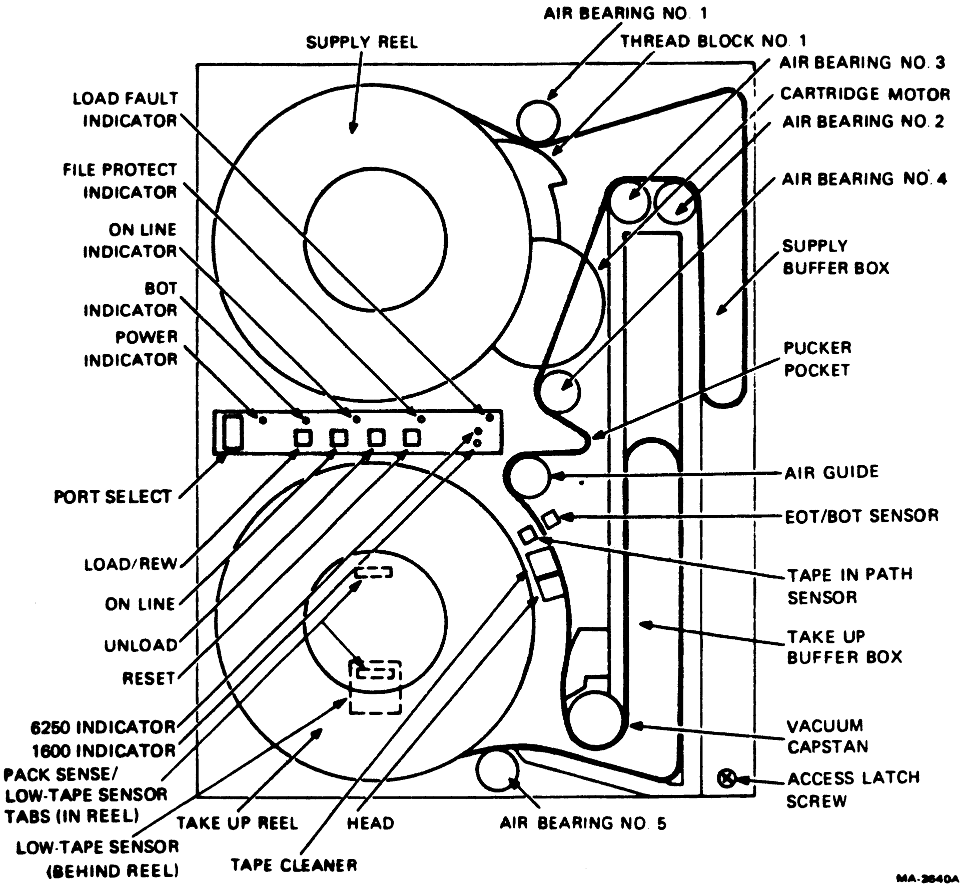


Figure 5-37 TU78 Tape Path and Controls

5.9.3.4 Manual Load -

1. Check that the tape path is clean.

2. If a write operation is to be performed, install a write enable ring onto the rear flange of the tape reel.

3. Place the supply reel in position on the upper hub. Rotate the reel until it slips easily into place, and press the reel retaining actuator. The reel should be positioned so that the tape unwinds if it is turned clockwise.

4. Place the tape leader between thread block 1 and air bearing 1 (Figure 5-37).

NOTE

> Make sure that there is no tape slack or sag between the supply reel and thread block 1.

5. Close the transport front door.

6. Check that tape transport power light is on.

7. Follow the instructions and check for the events specified in Figure 5-38. If the tape fails to load properly, the figure contains some operator troubleshooting hints that can be tried before maintenance personnel are called. Refer to Figure 5-37 to identify tape path components.

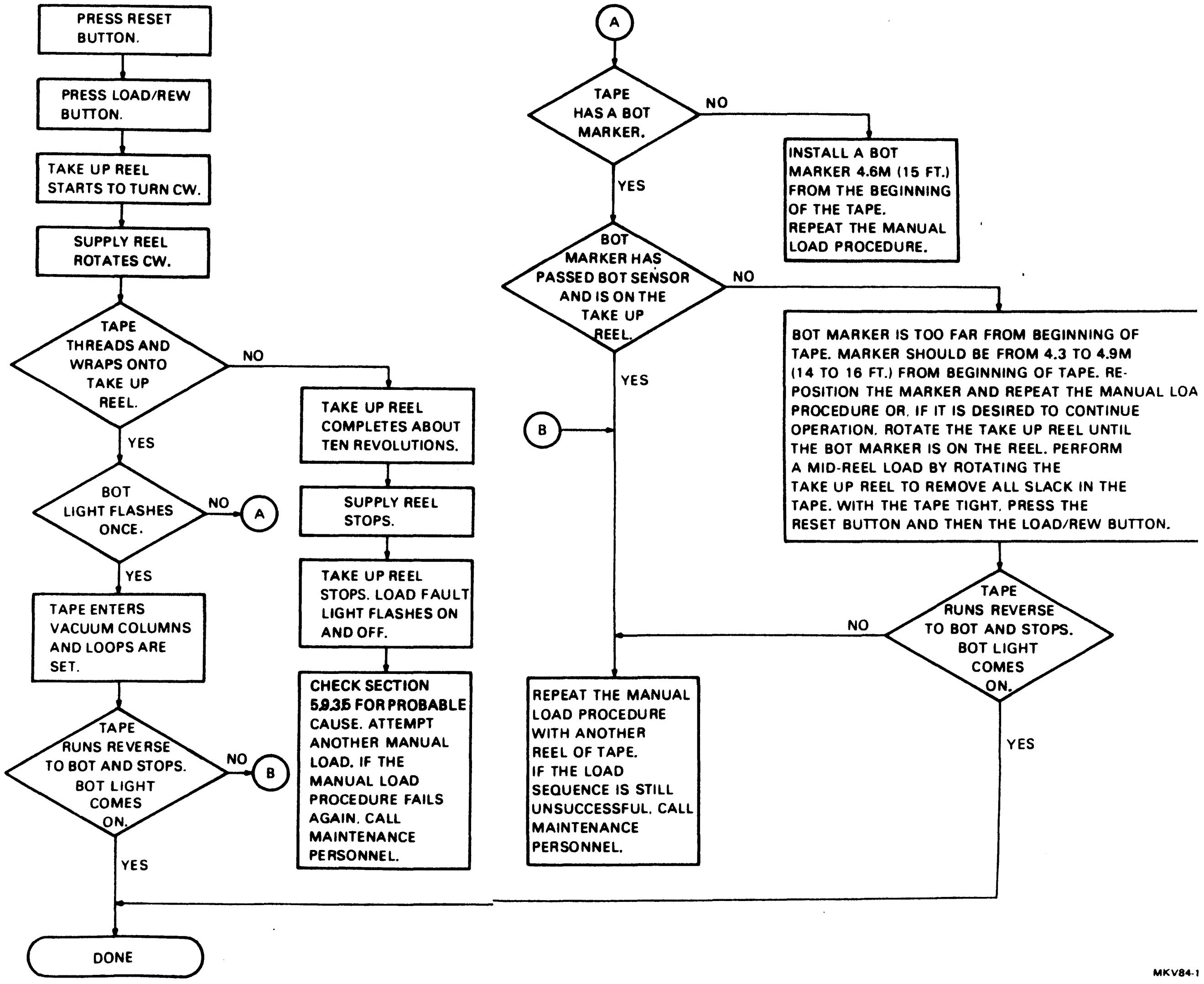

PRESS RESET BUTTON.
PRESS LOAD/REW BUTTON.
TAKE UP REEL STARTS TO TURN CW.
SUPPLY REEL ROTATES CW.
TAPE THREADS AND WRAPS ONTO TAKE UP REEL.
NO
YES
TAKE UP REEL COMPLETES ABOUT TEN REVOLUTIONS.
SUPPLY REEL STOPS.
TAKE UP REEL STOPS. LOAD FAULT LIGHT FLASHES ON AND OFF.
CHECK SECTION 5.9.35 FOR PROBABLE CAUSE. ATTEMPT ANOTHER MANUAL LOAD. IF THE MANUAL LOAD PROCEDURE FAILS AGAIN, CALL MAINTENANCE PERSONNEL.
BOT LIGHT FLASHES ONCE.
NO
A
YES
TAPE ENTERS VACUUM COLUMNS AND LOOPS ARE SET.
TAPE RUNS REVERSE TO BOT AND STOPS. BOT LIGHT COMES ON.
NO
B
YES
DONE
A
TAPE HAS A BOT MARKER.
NO
INSTALL A BOT MARKER 4.6M (15 FT.) FROM THE BEGINNING OF THE TAPE. REPEAT THE MANUAL LOAD PROCEDURE.
YES
BOT MARKER HAS PASSED BOT SENSOR AND IS ON THE TAKE UP REEL.
NO
BOT MARKER IS TOO FAR FROM BEGINNING OF TAPE. MARKER SHOULD BE FROM 4.3 TO 4.9M (14 TO 16 FT.) FROM BEGINNING OF TAPE. REPOSITION THE MARKER AND REPEAT THE MANUAL LOA PROCEDURE OR, IF IT IS DESIRED TO CONTINUE OPERATION, ROTATE THE TAKE UP REEL UNTIL THE BOT MARKER IS ON THE REEL. PERFORM A MID-REEL LOAD BY ROTATING THE TAKE UP REEL TO REMOVE ALL SLACK IN THE TAPE. WITH THE TAPE TIGHT, PRESS THE RESET BUTTON AND THEN THE LOAD/REW BUTTON.
YES
B
TAPE RUNS REVERSE TO BOT AND STOPS. BOT LIGHT COMES ON.
NO
YES
REPEAT THE MANUAL LOAD PROCEDURE WITH ANOTHER REEL OF TAPE. IF THE LOAD SEQUENCE IS STILL UNSUCCESSFUL, CALL MAINTENANCE PERSONNEL.
MKV84-1

Figure 5-38 TU78 Manual Load Sequence

5.9.3.5 Probable Causes Of Load Failure -

1. Check that the buffer door is closed and sealed properly.

2. Check that the front door is closed securely.

3. Check for ripped, creased, or damaged tape leader. This is especially critical if an auto-load failure has occurred. If necessary, crimp the end of the tape with a tape crimper.

4. Check that the write enable ring is inserted in the tape reel if a write operation is to be performed. The file protect indicator should be out.

5. If an auto-load is being performed, check the following.

   a. A 267 mm (10.5 in) tape reel is on the supply hub

   b. The supply reel has a full reel of tape. The tape must be between 0.64 cm and 1.59 cm (0.25 in and 0.63 in) from the outer edge of the reel.

5.9.3.6 Mid-reel Load -

A mid-reel load is usually required after a power failure or a lost interlock. Follow the steps below to perform a mid-reel load.

1. Rotate the take-up reel to remove all slack in the tape.

CAUTION

> Do not attempt a mid-reel load until all slack is manually removed from the tape between the reels.

2. Press the RESET button.

3. Close the front door.

4. Press the LOAD/REW button.

The transport loads the tape and starts a reverse operation to BOT. To stop the transport, press the RESET button. Any other command may now be given.

5.9.3.7 Unload -

1. Press the RESET button to terminate any current operation and place the transport off-line.

2. Press the UNLOAD button. If the tape is at BOT, it backwinds onto the supply reel. If the tape is at mid-reel, it rewinds to BOT and then unloads.

5.9.3.8 Rewind -

Press the LOAD/REW button, with tape tensioned and the transport off-line, to rewind the tape to BOT. When the button is pressed with the tape at BOT, no action occurs.

5.9.3.9 On-Line/Off-Line -

Press the ON LINE button to place the transport on-line. This lights the on-line indicator. While on-line, the transport can accept external commands provided it is selected and ready. If the ON LINE button is pressed again, the transport goes off-line. The on-line indicator goes out. Also, the transport goes off-line when any of the following conditions occur.

- An external rewind unload command is received
- Vacuum column interlock is broken
- AC power is lost
- RESET button is pressed
- Front door opens

5.9.4 TU78 Customer Care And Maintenance

5.9.4.1 Customer Responsibilities -

1. Obtain operating supplies, including magnetic tape and cleaning supplies.
2. Supply accessories, including cabinetry, tables, and chairs.
3. Maintain the required logs and report files consistently and accurately.
4. Make the necessary documentation available in a location convenient to the system.
5. Keep the exterior of the system and the surrounding area clean.
6. Make sure that ac plugs are securely plugged in each time equipment is used.

5.9.4.2 Magnetic Tape Care -

1. Do not expose magnetic tape to excessive heat or dust. Most tape read errors are caused by dust or dirt on the read head; it is imperative that the tape be kept clean.

2. Always store tape reels inside containers when they are not in use. Keep the empty containers tightly closed to keep out dust and dirt.

3. Never touch the portion of tape between the BOT and EOT markers. Oil from fingers attracts dust and dirt.

4. Never use a contaminated reel of tape. This spreads dirt to clean tape reels and could have an adverse effect on tape transport reliability.

5. Always handle tape reels by the hub hole. Squeezing the reel flanges could lead to tape edge damage in winding or unwinding tapes.

6. Do not smoke near the tape transport or storage area. Tobacco smoke and ash are especially damaging to tapes.

7. Do not place magnetic tape near any line printer or other device that produces paper dust.

8. Do not place magnetic tape in any location where it might be affected by hot air.

9. Do not store magnetic tape in the vicinity of electric motors.

5.9.4.3 TU78 Tape Transport Maintenance -

1. Dismount the tape from the unit.

2. Clean the following components of the transport using a foam-tipped swab soaked in cleaning fluid. Daily cleaning is recommended for TU78 units in constant operation. See Figure 5-39.

   Read/write head
   Erase head
   Tape cleaner
   Air bearings (6)
   Spring-loaded ceramic tape guides (2)
   Thread blocks (6)
   Capstan
   Buffer boxes (2)
   Pucker pocket
   Buffer box glass, pocket glass, door overlay

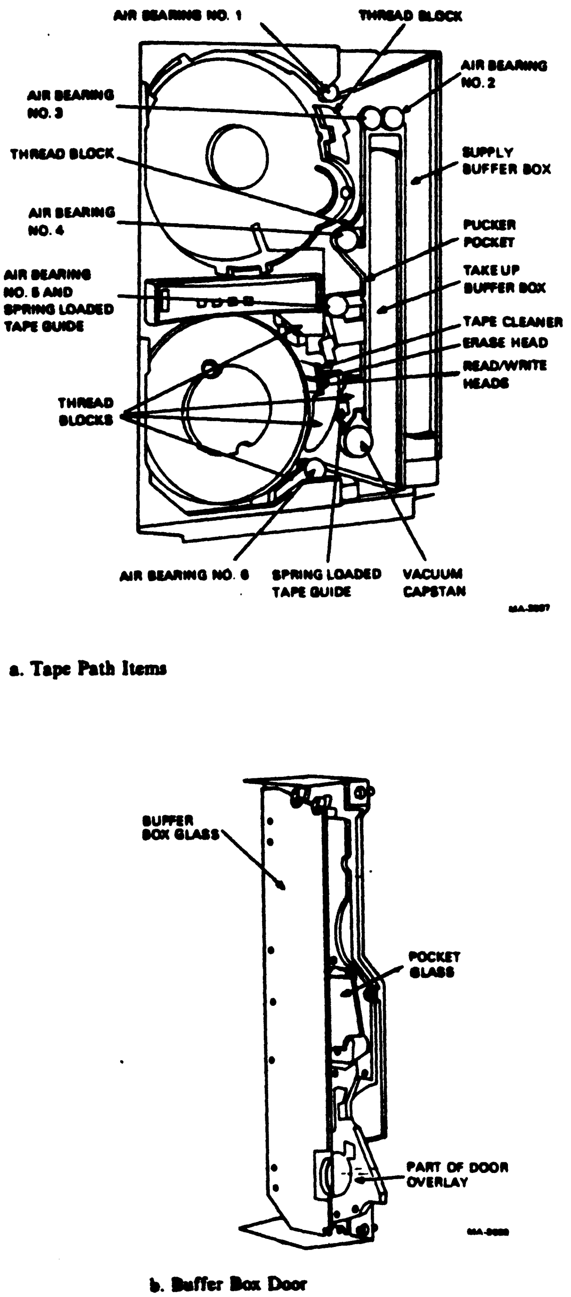
NOTE

Use only foam-tipped swabs. Do not touch any part of the tape path since oil from fingers attracts dust and dirt. Also, excessive physical pressure on the capstan could affect its alignment.

3. When cleaning the thread blocks, be sure to clean the air guide ports. Every thread block has air guide ports except the metal block containing the spring-loaded tape guide.

4. When cleaning the spring-loaded ceramic guides, be sure that the washer is pressed firmly against the tape guide surface and not hung up on its shaft.

5. When cleaning the inner surface of the vacuum door, use a lint-free wipe and cleaning fluid. Pass another lint-free wipe over the head using a polishing action to remove any remaining deposits.



a. Tape Path Items

b. Buffer Box Door

Figure 5-39 TU78 Transport Items to Clean Daily

5.10 TU80 MAGNETIC TAPE SUBSYSTEM

The TU80 magnetic tape subsystem is a low-cost, dual-speed 9-track data storage system. It is packaged with its formatter and power controller in a standard 40-inch high H9643 cabinet. The TU80 writes and reads data in ANSI-compatible phase-encoded (PE) 9-track 1600-bit/inch format. The tape speed is 25 in/s (63.5 cm/s) in start/stop mode and 25 in/s (63.5 cm/s) or 100 in/s (254 cm/s) in streaming mode.

The TU80 Subsystem consists of the standalone tape transport and a UNIBUS adapter module. It plugs into the VAX-11/785 UNIBUS small peripheral controller (SPC) slot and provides communication with the corresponding tape transport.

Figure 5-40 shows the TU80 and its major components.

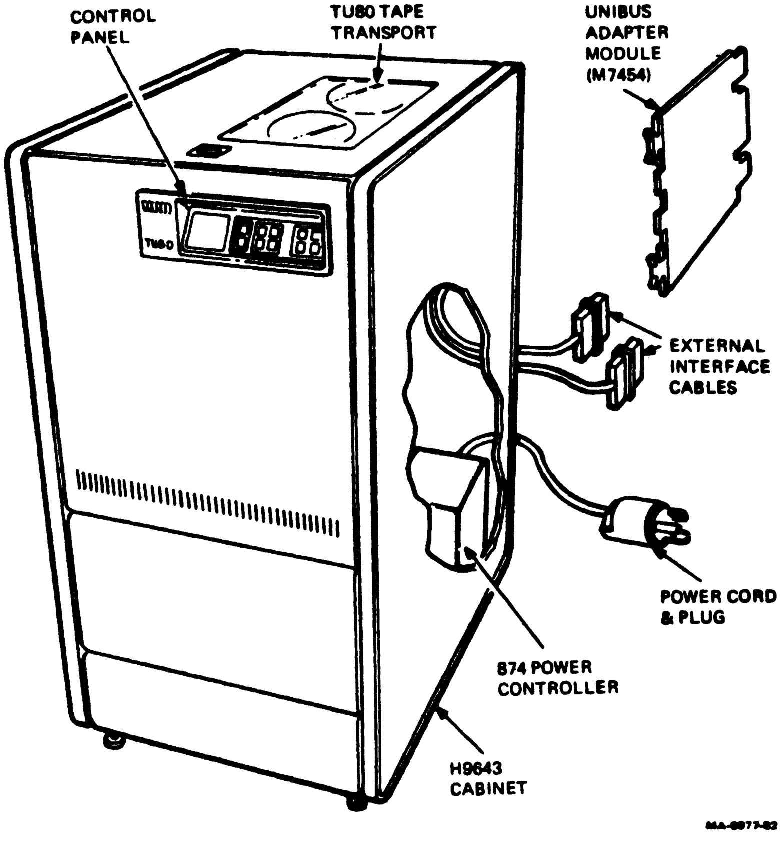


Figure 5-40 TU80 Tape Subsystem

5.10.1 TU80 Specifications

NOTE

Back-coated magnetic tapes are not recommended for use on the TU80 tape drive.

Table 5-27 TU80 Specifications

Functional Characteristics	
Tape speed (nominal)	
Low speed start/stop mode	25 in/s
Low speed streaming mode	25 in/s
High speed streaming mode	100 in/s
Tape capacity (maximum)	40 Mbytes
Data transfer rate (burst)	
Low speed modes	40 Kbyte/s
High speed streaming mode	160 Kbyte/s
Data format/recording	9 track, phase-encoding, 1600 bit/inch
Rewind time (nominal)	2.5 minutes
Load time (maximum)	15 seconds
Physical Characteristics	
Electronics	Solid-state
Tape width	12.65 mm (0.5 in)
Tape thickness	38.1 micron (1.5 mil)
Tape tension	2.23 N (8 oz)
Reel diameter	26.7 cm* (10.5 in)
Reel capacity	732 m* (2400 ft)
Transport dimensions	105.7 cm (H) × 54.6 cm (W) × 76.2 cm (D) 41.6 in (H) × 21.5 in (W) × 30 in (D)
Transport weight (in cabinet)	127 kg (280 lbs)
UNIBUS adapter	Standard quad-size module
Power requirement	4 A at +5 Vdc

* Smaller tape reels are also allowed.

Table 5-27 TU80 Specifications (Cont)

Operational Characteristics	
Power requirements	
Voltage	93–128 Vac, 120 Vac nominal, 60 Hz, single phase 187–256 Vac, 220 or 240 Vac nominal, 50 Hz, single phase
Current	Average input current 2.0 A rms at 100 in/s
Power consumption	300 VA – Standby and loaded 550 VA max – Start/stop
Power controller	Model 874B, 50 Hz, 240 V, 12 A Model 874D, 60 Hz, 120 V, 24 A
Power cord	3 wire number 12 AWG 16 ft long; plug - NEMA 6-15P (50 Hz) NEMA 5-30 (60 Hz)
Operating temperature†	10° C (50° F) to 40° C (104° F)‡
Storage temperature	−10° C (14° F) to 50° C (122° F)
Relative humidity	20% to 80% (10% to 90% in storage)
Altitude	Up to 3048 m (10,000 ft) or 688 millibar (9.98 psi)
Heat dissipation (average)	1024 BTU/hr
Acoustical noise	51 dba for open office environment
Data Reliability	
Recoverable write error	1 in 10^8 bits
Recoverable read-forward error	1 in 10^9 bits
Recoverable read-reverse error	1 in 10^8 bits
Unrecoverable read error	1 in 10^{10}
Unrecoverable write error	Not allowed

† Restricted by the operating temperature of the media.
‡ Operating temperature will be limited by the temperature limitations of the media.

5.10.2 TU80 Controls And Indicators

The left part of the control panel (Figure 5-41) contains operator switches and indicators. The 2-digit display indicates a fault code related to an abnormal operating condition or a test code when in the diagnostic mode. The right part of the panel contains the switches for diagnostic test procedures.

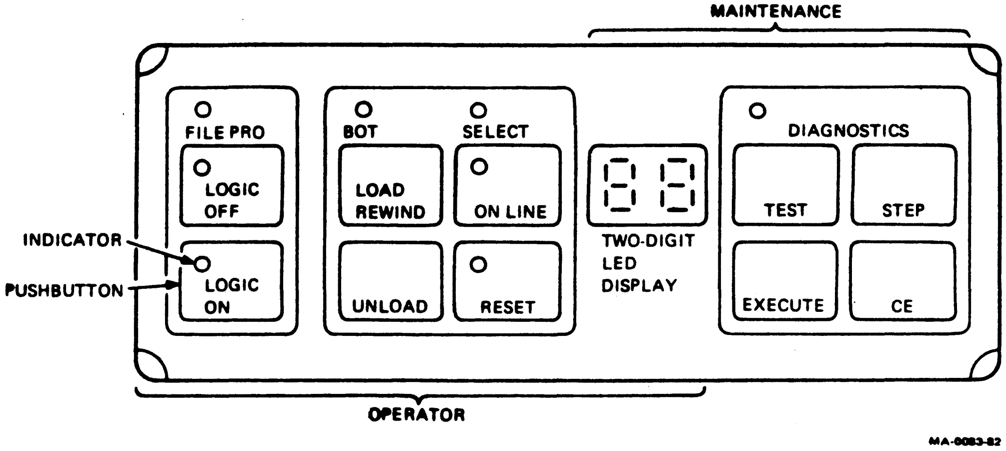


Figure 5-41 TU80 Control Panel

Table 5-28 TU80 Operator Controls

SWITCH/INDICATOR	FUNCTION
LOGIC OFF	Switch - When pressed, power is removed from the transport logic and LOGIC OFF indicator goes on. Indicator - When on, LOGIC OFF indicates a standby power condition exists (power supply is on, transport logic is off). If the indicator is off and no other indicators are on, then power switch is set off (0) or there is a power supply problem. If this indicator is off and LOGIC ON indicator is on, all transport circuitry is in the power-on condition.
LOGIC ON	Switch - If LOGIC ON is pressed when the power switch is on (1), the transport logic and control system is powered up and the TU80 is ready for operation. Indicator - When on, LOGIC ON indicates the transport is powered up and ready for use.
FILE PRO	Indicator - When on, FILE PRO indicates absence of a write-enable ring in the supply reel and write operation is inhibited. Otherwise, write operations are allowed.
BOT	Indicator - When on, BOT indicates the tape is positioned at the beginning-of-tape (BOT) marker.
LOAD/REWIND	Switch - If the transport is powered on and tape is threaded, pressing this switch causes a load operation to be performed. If tape is loaded, pressing the switch causes a rewind operation to BOT.

Table 5-28 TU80 Operator Controls (Cont)

SWITCH/INDICATOR	FUNCTION
UNLOAD	Switch - If tape is loaded at BOT, pressing this switch causes tape to unload from the take-up reel and tape path onto the supply reel. If tape is threaded, but not loaded, pressing the switch causes the transport to slowly unload tape onto the supply reel.
ON-LINE	Switch - If tape is loaded, pressing this switch causes the transport to go on-line to the host system. Press the RESET switch to take the TU80 off-line.
SELECT	Indicator - When on, indicates the following. - The address in TU80 and CPU are the same. - When the two I/O cables are disconnected, the address decoder in TU80 is functional. - Pin 1 in I/O cables and Pin 1 on the M7454 are assembled correctly. - TU80 is not in diagnostic mode.
RESET	Switch - When pressed, this switch places the transport off-line, stops tape motion, and clears error status. The switch can stop a LOAD or REWIND operation. It also turns off the RESET indicator and clears diagnostic test condition. Indicator - When on, RESET indicates that the transport is in fault or diagnostic status. Two-digit display may contain a fault or diagnostic code.

Table 5-29 TU80 Diagnostic Controls

SWITCH/INDICATOR	FUNCTION
DIAGNOSTIC	Indicator - When on, DIAGNOSTIC indicates that the transport is in the diagnostic/test mode. The indicator remains on until the RESET switch is pressed.
TEST	Switch - If the transport has tape threaded but not loaded, pressing this switch places the unit in the diagnostic/test mode.
STEP	Switch - If the transport is in the diagnostic/test mode (and the test number is displayed), pressing this switch allows the diagnostic/test sequence numbers in the two-digit display to increment by one.
EXECUTE	Switch - If the transport is in the diagnostic/test mode (and the test number is displayed), pressing this switch initiates the internal diagnostic test shown on the two-digit display.
CE	Switch - If the transport is in the diagnostic/test mode, this switch helps run Field Service Diagnostics.

Operator's Hints

1. When the LOGIC ON indicator is on, all control panel switches and indicators are functional except the LOGIC OFF indicator.

2. The ON-LINE switch can be pressed while the tape is loading. The TU80 goes on-line immediately when loaded.

3. When the RESET indicator is on, the control panel is not operable except for the LOGIC OFF and RESET switches. Press the RESET switch to turn off the RESET indicator.

4. Tape unloading is performed at low speed. To speed the procedure press LOAD REWIND, then UNLOAD.

5.10.3 TU80 Operating Instructions

5.10.3.1 Inserting Write-Enable Ring -

Before mounting the reel of tape onto the supply hub, determine if the tape is to be recorded. If write operations are to be performed, place a write-enable ring into the bottom recessed portion of the reel (Figure 5-42).

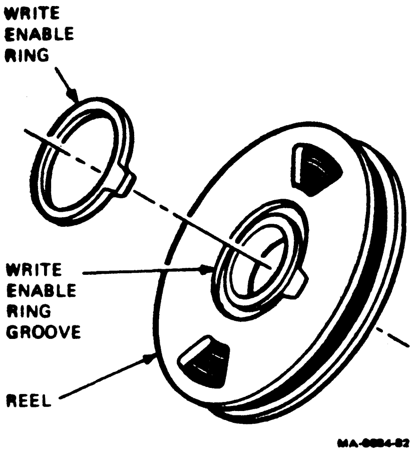


Figure 5-42 Write Enable Ring

5.10.3.2 Tape Threading And Loading -

NOTE

> The circuit breaker on the 874 power controller must be ON.

1. Pull the cover latch toward the front and lift the top access cover. Check that the power switch is ON (1 pressed). The LOGIC OFF indicator should be on.

NOTE

> The power switch can be left in the ON position to enable the TU80 transport to be powered-on under CPU control.

2. Press the LOGIC ON switch. The power-up health check is initiated. The LOGIC OFF indicator goes off and all other indicators momentarily go on. The LOGIC ON and FILE PRO indicators should remain on.

NOTE

> Pressing the LOGIC ON switch enables the transport power amplifier module, control logic, and support circuits. If a fault code displays at this time, press the RESET and LOGIC OFF switches and then the LOGIC ON switch again to repeat the check.

3. Press the inner button on the face of the supply reel hub.

4. Mount the supply reel onto the hub so that the reel is seated on the bottom flange. Secure the reel by pressing the periphery of the hub face to latch the reel.

5. Thread magnetic tape through the tape path as shown in Figure 5-43.

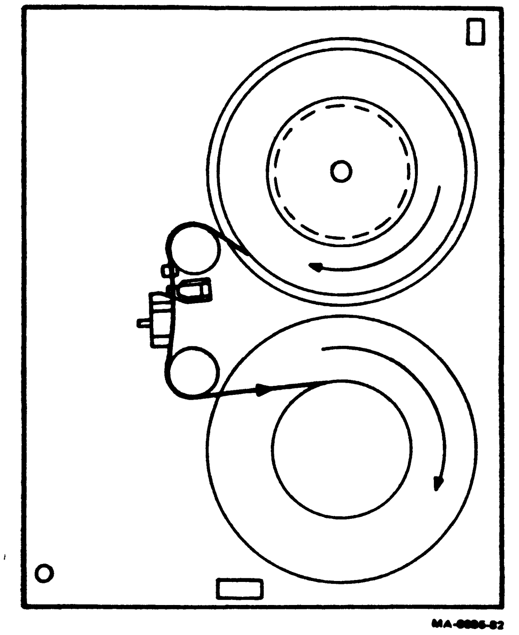


Figure 5-43 TU80 Tape Path

6. Wrap the tape leader (a length of tape before the BOT marker) onto the take-up reel for several turns.

CAUTION

> Make sure that the tape is positioned correctly over all tape path components, otherwise tape damage may occur.

7. Close and latch the top cover.

8. Press the LOAD/REWIND switch. In one second air pressure builds at the air bearings and the tape starts moving forward. This motion stops when the BOT marker on the tape is detected. If the tape was over-threaded and the BOT marker was positioned after the sensor, forward tape motion continues for approximately 40 feet. Then the transport initiates reverse tape motion until the BOT marker is detected. At this moment, tape motion stops and the BOT indicator is on.

9. Press the ON-LINE switch to place the unit on-line to the VAX-11/785. The switch may be pressed while the load operation is in progress. On completion of a load operation, the transport indicates ON-LINE status by turning the ON-LINE indicator on. (If the transport is ready for a write operation, the FILE PRO indicator goes off.) Now the transport is ready for use.

Tape Unloading

Manual Mode

1. Press the RESET switch to place the transport off-line.

2. Press the UNLOAD switch. The tape now moves in reverse (rewind), gently winding onto the supply tape reel until it clears the take-up reel and the tape path.

NOTE

> If the tape is loaded beyond the BOT marker, it rewinds to BOT.

3. Open the top cover and press the center button of the supply reel hub. The hub unlatches and the supply reel can be removed.

4. Close the top cover to prevent dust accumulation on the tape deck components.

On-Line Mode

When on-line to the VAX-11/785, the UNLOAD operation can be performed under CPU control by the TU80. To remove the reel, perform steps 3 and 4 above.

5.10.3.3 Operator Corrective Actions -

When the LOGIC ON switch is pressed, the TU80 transport initiates a built-in power-on health check. If a fault occurs that cannot be corrected by the VAX-11/785 and the transport takes itself off-line, the operator is informed of the fault by the fault code in the two-digit display. Refer to Table 5-30 for the list of fault codes and corresponding corrective actions.

The operator can check the TU80 transport operating condition using the built-in basic Diagnostic Test 01. This test checks the basic operating functions of the transport, and it can verify a malfunction when a fault code indicates transport failure. Table 5-30 lists fault codes and operator corrective actions.

Table 5-30 TU80 Operator Corrective Actions

Fault Code	Corrective Action
01–09	Clean the magnetic head and tape path following instructions in Chapter 4 of this manual.
10	Make sure the top cover is securely closed.
11	Thread the tape.
12	Latch the supply reel hub.
13	Thread the tape correctly as shown on the tape threading diagram.
14	Check for BOT marker on tape. Attach BOT/EOT marker following instructions in Chapter 4.
15	This fault code indicates that the RESET switch was pressed inadvertently. Reinitiate the test.
16	Check for presence of the write-enable ring in the supply tape reel. Install the ring if not present.
17	Check for presence of EOT marker. Attach EOT marker following instructions in Chapter 4.
18	Indicates that the tape was loaded when the test was initiated. Thread the tape, but DO NOT press LOAD switch.
20–29	Mount a tape of known good quality.
All others	Attempt to run Diagnostic Test 01 (refer to Chapter 3) and then report a fault code to Field Service.

5.10.4 TU80 Customer Care And Maintenance

5.10.4.1 Customer Responsibilities -

1. Obtaining operating supplies, including magnetic tape and cleaning supplies.

2. Keeping the exterior of the system and the surrounding area clean.

3. Making sure that ac power plugs are securely plugged in every time the equipment is used.

4. Performing the specific operations for equipment care described in this section at the suggested periods, or more often if usage and environment warrant.

Also, it is recommended that the customer:

1. Maintain the required logs and report files consistently and accurately.

2. Make the necessary documentation available in a location convenient to the system.

5.10.4.2 Care Of Magnetic Tape -

1. Do not expose magnetic tape to excessive heat or dust. Most tape read errors are caused by dust or dirt on the read/write head. It is important to keep the tape clean.

2. Always store the tape reels inside their containers when the tape is not in use. Keep all empty containers tightly closed to guard against dust and dirt.

3. Never touch the portion of tape between the BOT and EOT markers. Oil from fingers attracts dust and dirt.

4. Never use a contaminated reel of tape. This spreads dirt to the clean tape reels and could adversely affect tape transport reliability.

5. Always handle tape reels by the hub hole. Squeezing the reel flanges leads to tape edge damage when winding or unwinding tapes.

6. Do not smoke near the tape transport or storage area. Tobacco smoke and ash are especially damaging to tapes.

7. Do not place magnetic tape near line printers or other devices that produce paper dust.

8. Do not place magnetic tape in any location where it may be affected by hot air.

9. Do not store magnetic tape near electric motors or any other magnetic sources that may erase data.

5.10.4.3 Tape Transport Customer Maintenance -

Before cleaning the tape transport, remove the supply reel and store it properly. When cleaning, be gentle but thorough.

CAUTION

> Do not use acetone, lacquer thinner, rubbing alcohol, or trichlorethylene to clean the tape path.

Magnetic Tape Transport Cleaning Kit

Unscrew the top from the cleaning fluid can and punch a small hole in the metal seal covering the pour spout.

WARNING

> When using DECmagtape cleaning fluid, avoid excessive skin contact and contact with the eyes. Do not swallow it. Use the cleaning fluid only in a well-ventilated area.

CAUTION

> When cleaning the tape equipment, never dip a dirty cleaning swab or wipe into the can. Instead pour a little fluid onto the swab or wipe.
>
> Use the smallest amount of cleaning fluid necessary. DO NOT soak the swab or wipe in the cleaning fluid.

Always keep the can of fluid tightly closed when not in use; the fluid evaporates rapidly when exposed to air.

Use the cleaning materials from the kit to clean tape heads, tape guides, tape cleaner, and any part of the drive where residue could contact the tape. To clean other parts of the drive, such as the exterior tape deck surfaces or doors, use any clean, lint-free material with or without cleaning fluid.

CAUTION

To clean exterior surfaces, use only soap and water.

A mild soap and water solution can be used on any unusually stubborn dirt that resists the cleaning fluid. After using soap, be sure to wash the area thoroughly with cleaning fluid to remove the soap residue.

Reflective Tape Markers

Every reel of magnetic tape must have BOT and EOT reflective markers, which enable the transport to recognize starting and stopping areas. Tapes are always supplied with reflective markers attached. However, if the markers become detached for any reason, or if a tape leader is shortened because of tape damage, then the operator must install a marker at the position shown in Figure 5-44.

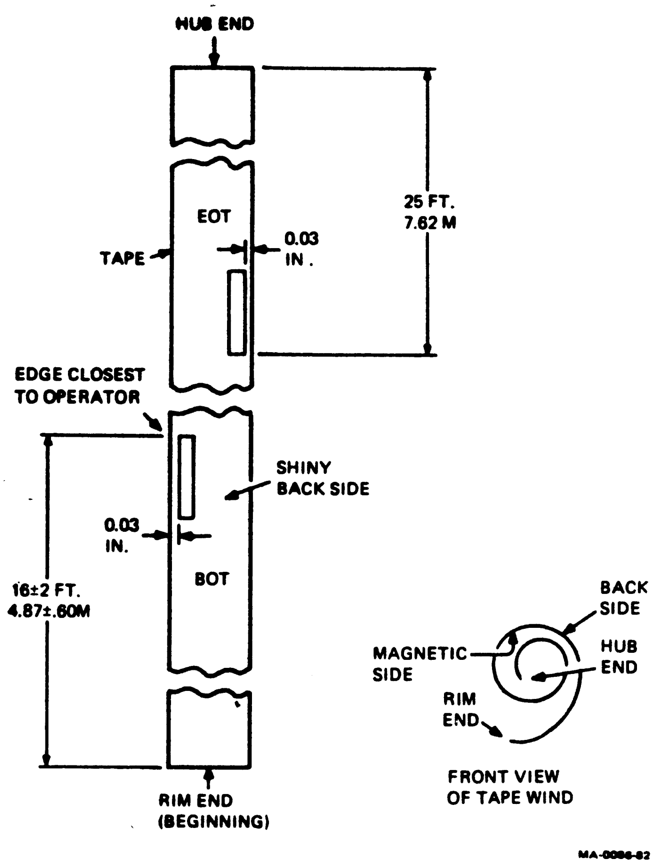


Figure 5-44 Reflective Tape Markers

Transport Cleaning

All transport components that need regular cleaning are located on the tape deck. These components include the magnetic head assembly, EOT/BOT sensor, tape cleaner, and air bearings. Also, the user should regularly examine and clean, as required, the top of the tape deck, reel hubs, and top cover door.

To access the magnetic head components (Figure 5-45), pull up the two-part head assembly dust cover from the tape deck.

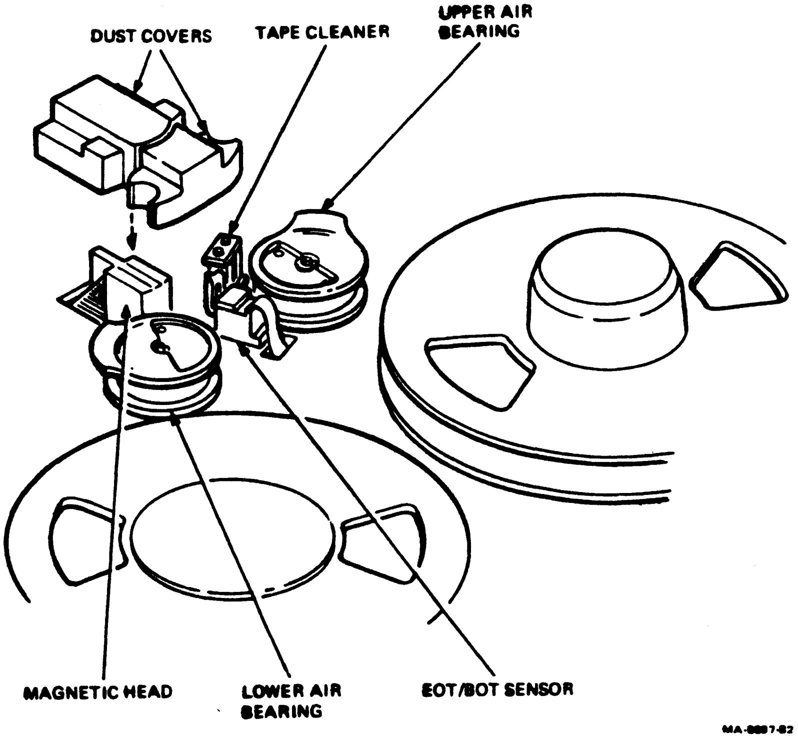


Figure 5-45 TU80 Magnetic Head and Tape Path Components

Magnetic Heads

1. Clean the magnetic head working surface with a soft lint-free wipe moistened with the cleaning fluid.

NOTE

> Wipe the head surface in the same direction that the tape travels during data recording (forward motion).

2. Clean the area adjacent to the head of any dirt that can be transported with the tape.

Air Bearings

1. Clean both air bearings with a swab moistened with the cleaning fluid.
2. Make sure that the foil area, guide springs, and both tape guides are cleared of any oxide or dirt buildup. If available, use a small mirror to inspect the inner edge of the guides.

Top Cover, Tape Deck, and Head Assembly Dust Covers

To prevent the transfer of dirt to the tape components, DO NOT allow dust or dirt to accumulate on the inside of the dust cover. Clean the inside cover surface with a clean, lint-free wipe and any cleaning fluid that is safe for plastic, painted surfaces.

APPENDIX A
VIRTUAL MEMORY CONCEPTS

Virtual memory provides major advantages over traditional computer system designs. Virtual memory benefits are:

1. Allows the user to run programs larger than the size of real memory
2. Eliminates overlays
3. Is transparent to the user
4. Provides direct improvement in performance as real memory is added.

The programmer is freed from restrictive programming practices and memory constraints. However, the programmer may fine-tune program performance and efficiency.

The following paragraphs develop some basic concepts (of virtual memory systems) that are prerequisite to an understanding of the VAX-11 system design.

A.1 VIRTUAL ADDRESS

A source program is composed of data and instructions that are processed by a compiler or assembler to produce executable code. During compiling or assembly, the data and instructions of the program are allocated virtual addresses (addresses used to identify or represent memory locations). For example, in Figure A-1, source Program A is processed and allocated virtual addresses that start at address 0 and continue through address 1000.

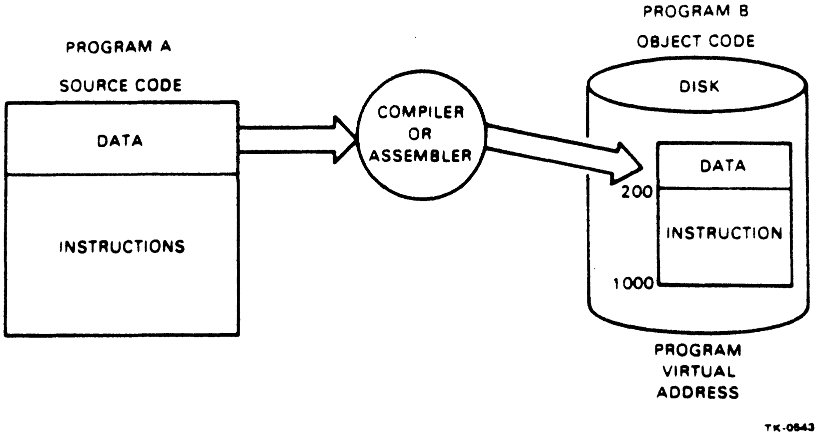


Figure A-1 Program Virtual Address

For the object program to be executed it must be loaded into physical memory. The simplest kind of program loading is absolute program loading. A software program called an absolute loader loads the object program into the same physical addresses as the program's virtual addresses. In this case, the virtual address is the same as the physical address. Additional mechanisms in the hardware and software are required to make the virtual address anything different from the physical address.

A system that uses absolute loading has limited flexibility because a program is always loaded into the same physical addresses as its virtual addresses. Separately compiled or assembled routines all start with virtual address 0. To combine several routines into one program, the addresses of each routine must be changed so they do not conflict.

A.2 PROGRAM LINKING

Linking is the process of combining object programs (e.g., a main routine and several subroutines). The linker program relocates (changes) the virtual addresses of each routine to prepare the combined programs for loading into memory. The programmer can instruct the linker to assign specific virtual addresses to programs, so that two or more programs will not use the same virtual addresses.

This kind of relocation is static address relocation (done at link time), and is the linking process available in systems that have no memory management or relocation hardware. Static address relocation can be used also to resolve addressing conflicts between separate programs when they reside concurrently in unmapped physical memory. This technique is used by multiprogramming operating systems.

A.3 MAPPING

Dynamic address relocation (sometimes called mapping) is done after program loading. Mapping continually relocates addresses in the program during program execution. Special mapping hardware is used to change a program's virtual addresses into physical addresses used by the hardware.

Figure A-2 shows three separate programs, A, B, and C, each of which has different virtual memory requirements starting at address 0.

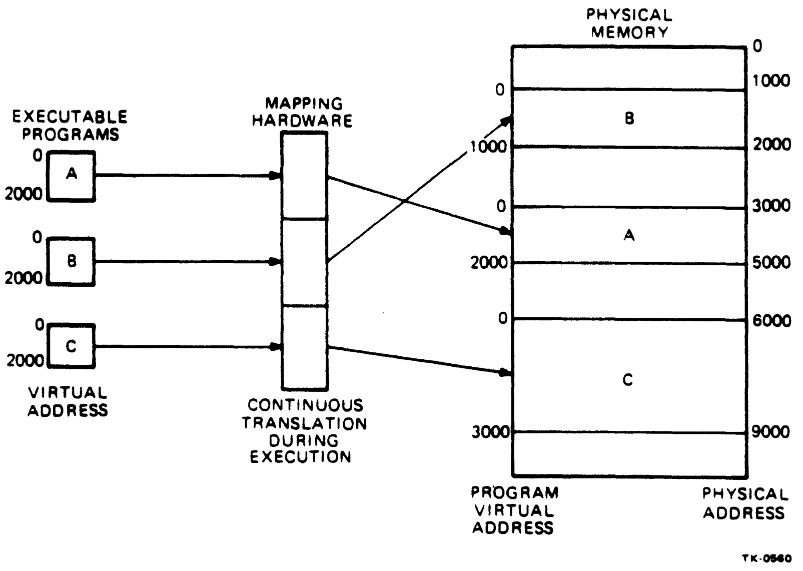


Figure A-2 Mapping

A benefit of mapping hardware is that a program can be run in different areas of physical memory without relinking. The relocation is done at run-time by special mapping hardware. The hardware is very fast and does not significantly degrade the performance of the program.

A.4 MULTIPROGRAMMING

In order to make full use of computer hardware, it is often desirable to have several programs in memory concurrently. This technique is called multiprogramming. Mapping is often used to resolve address conflicts between programs in a multiprogramming environment.

A.5 MEMORY FRAGMENTATION

Memory fragmentation is a problem that occurs when the available memory is broken into segments that are too small to accommodate each program in its entirety.

Figure A-3 shows three executable programs, A, B, and C. Program A is 4000 bytes long, B is 4000 bytes long, and C is 2000 bytes long. The available physical memory in this example is 11000 bytes, so there is clearly sufficient physical memory to hold all three programs. However, if program A has been loaded at address 1000 and B at address 6000 as shown here, there is no single contiguous 2000 byte area for C.

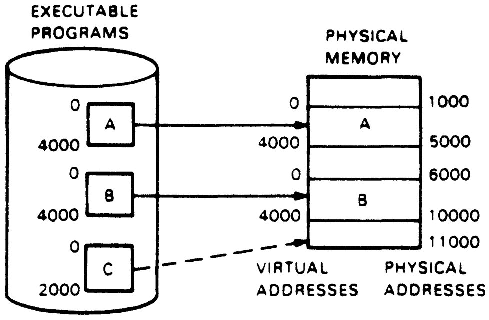


Figure A-3 Physical Memory Fragmentation

One solution employed, when holes develop in memory is to shuffle programs toward one end of memory, leaving a contiguous area of available memory at the other end.

Another solution is to map programs in segments that are smaller than the entire program.

A.6 NONCONTIGUOUS MAPPING

Although programs have been shown as being mapped as a single unit, it is not a requirement that the program be mapped as one unit.

Breaking the program into units and mapping the units separately is called noncontiguous mapping. Because the program is mapped in several pieces, the program need not be loaded in contiguous memory. Mapping executable programs in small increments reduces the problem of memory fragmentation because there is a higher probability of finding areas of memory that are large enough to hold the smaller units of a program.

A.7 PAGED MAPPING

A logical extension of the concepts of noncontiguous mapping is the concept of paged mapping. Paged mapping refers to the mapping of programs and data in units that have a fixed length. These units are called pages.

Mapping a program in pages further reduces memory fragmentation. Since memory is mapped in equal length pages, all free memory is divided into pages and is the correct size for pages to be loaded.

Figure A-4 illustrates a program composed of three sections, main routine A, and subroutines B and C. These routines are subdivided into fixed length pieces (pages); A is shown as being two pages long, B four pages long, and C two pages long.

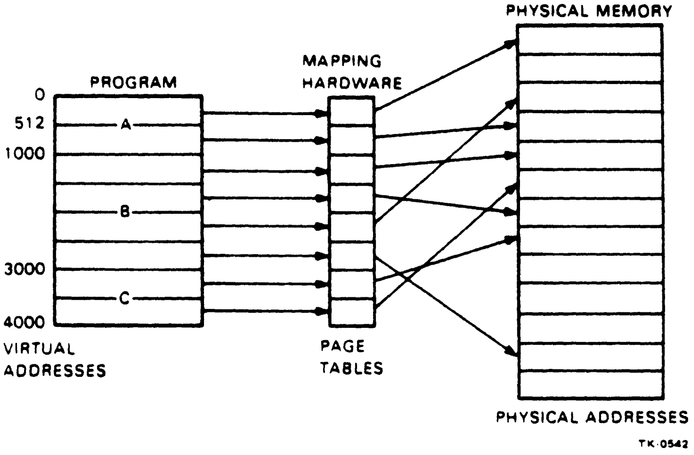


Figure A-4 Paged Mapping

Mapping hardware can now locate each page separately.

One of the benefits of mapping, and paged mapping in particular, is the ability for programs to easily share data or code. The specification of data or code to be shared is done explicitly by the programmer at link time. When the programs are physically loaded into memory, the operating system recognizes those pages to be shared between two or more programs and maps the common data or code to the same area of physical memory.

So far, discussion has been limited to situations in which the virtual memory requirements of the program(s) involved do not exceed the physical memory capabilities of the system. What if a program's virtual memory requirement is greater than the available physical memory of the system?

A.8 OVERLAYING

Overlay structures have been used to overcome the limitation of not enough memory. Overlays are logical segments of a program and are specified at link-time. The overlay structure defines a root section (which is always resident in memory), and two or more overlay sections (which are loaded into the same memory during program execution).

In Figure A-5, the root section is linked for addresses 0–1000, each overlay is linked for addresses 1000–2000. In this scheme, overlays 1A and 1B share the same physical memory, but at different time during the execution of the program. The overlay (either 1A or 1B) is loaded from disk as needed.

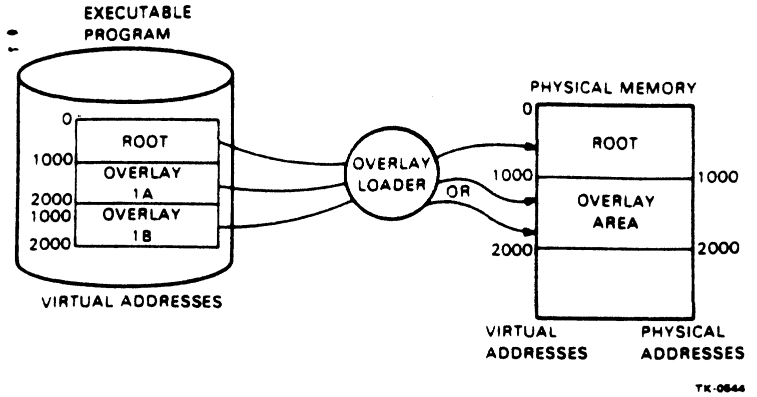


Figure A-5 Overlays

A different type of overlay scheme can be used when the available physical memory is larger than the virtual address space (Figure A-6).

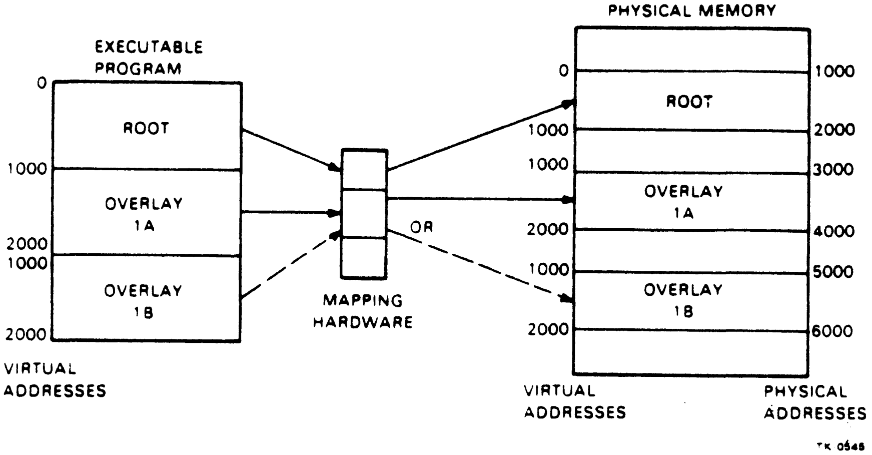


Figure A-6 Mapping Overlays

This example shows the same root and overlay sections; but in this case, mapping hardware does continual virtual to physical address translation at run time. Overlays 1A and 1B may be in physical memory at the same time. The programmer must still define the overlay structure at link time. This process is known as the in-core overlay technique, which is implemented by program logical address space (PLAS).

Thus, defining overlay structures and using overlay mapping are two ways of circumventing the problem of small virtual memory. However, in each case, it is the programmer's responsibility to plan and specify an overlay structure and to ensure that it works properly.

A.9 PROCESS

Programs on the VAX-11 system are run as part of a process. A process is like a task on other systems. When a program is running, it is known to VAX/VMS as a process.

A.10 PAGING

As the concept of overlaying has implied, it is not necessary to have all of a program resident in memory at one time. In fact, only those segments that are actually required for execution at any given time need be resident.

In the VAX-11 system, the set of segments, called pages, that reside in memory for any given process is called the process working set. Usually the working set is a subset of the total number of pages of the entire process, and will change over time as the program runs.

As a process executes, pages are brought into memory as required. Eventually, as execution proceeds, the process references pages that are not in memory. Whenever a process references a page not in memory, a page fault occurs. In order for execution to continue, the required page must be brought into memory. This is done by the operating system, which is transparent to the user. This is calleu paging.

As shown in Figure A-7, pages move back and forth between a file on disk and physical memory under the control of a page fault handler, which is part of the operating system.

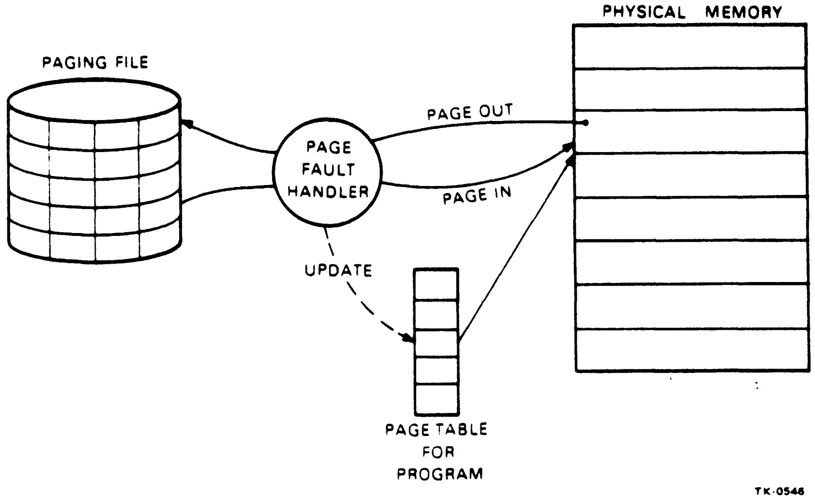


Figure A-7 Paging

On the VAX-11 operating system, all processes have a maximum working set size established for them. The maximum working set size is the total number of pages that a particular process may have resident in physical memory at any given time. The maximum working set for each process is decided by the system manager, based on both total number of users and the size of physical memory.

In order to avoid exceeding the maximum working set size for a process, a page from the current working set of the process may have to be removed.

The VAX-11 uses process-oriented paging. It discards least-used pages from the process in operation in order to make room for new pages. This is called paging against the process. Because a process in VAX/VMS pages only against itself, performance of individual processes can be controlled by the user. The performance of each process is more predictable than with system-wide paging.

A.11 SWAPPING

In a multiprogramming system, where two or more processes may be in physical memory at the same time, the term balance set is used to define the number of processes that are in memory. The balance set may be less than or equal to the total number of active processes in the system.

The number of processes in the balance set is controlled by the system and is determined by the amount of available physical memory. The system tries to balance the number of active processes in memory against the resources that are available to serve them.

It is the function of the scheduler, a system service, to select one process for execution from those processes currently in the balance set.

Figure A-8 shows five processes A, B, C, D, and E, all currently active. However, only the working sets of processes A, B, and C are in physical memory (i.e., in the balance set). The working sets of processes D and E are not in physical memory; consequently, processes D and E are not in the balance set.

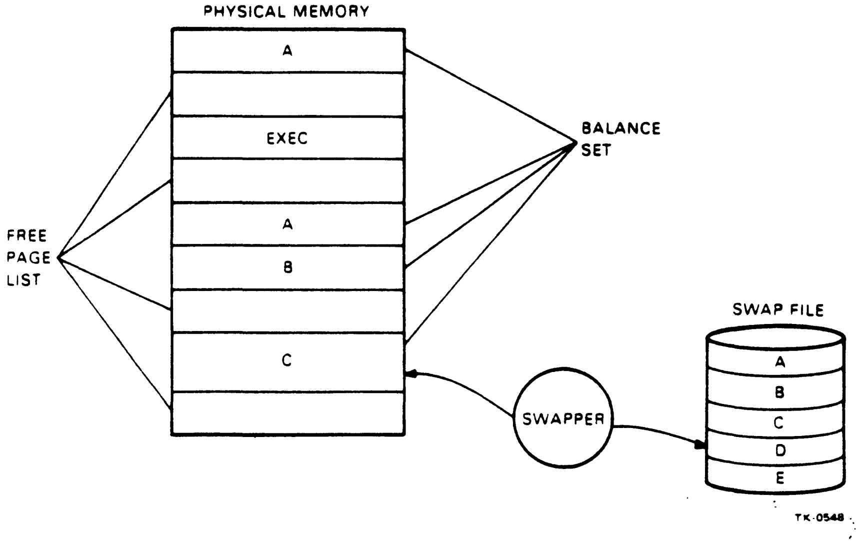


Figure A-8 Balance Set

In order for nonresident processes D or E to be scheduled to execute, they must first be brought into the balance set. To avoid exceeding the number of programs allowed in the balance set processes D and E must be substituted (swapped) for active processes already in the balance set.

Swapping, then, is the moving of entire working sets between physical memory and disk so that a nonresident process can be scheduled to execute.

Note the contrast between swapping and paging. Swapping is the moving of the working set (i.e., all the pages that make up the working set) of a process to allow a new process to be scheduled. Paging is the replacement of just a single page to allow a single process to continue to execute.

APPENDIX B
BOOTSTRAPPING SEQUENCES

The VAX-11 system can be bootstrapped from a number of sets of conditions. Operator participation in the bootstrapping operation is limited. All bootstrap sequences are performed with the control panel key selector switch in the LOCAL position.

Console and CPU power is applied and the AUTO RESTART switch is OFF. This is an automatic bootstrap sequence that is performed when power is applied. When completed, the console subsystem is booted and operating in the console I/O mode. The CPU subsystem is not booted. The sequence for this bootstrap is as follows.

1. Apply power to the console and CPU subsystems. Set the five position rotary key selector switch to LOCAL. (Disk RX1A must be installed in the RX01 disk drive.)

2. The console boot program (resident in the LSI-11 ROM) is initiated.

3. The console boot begins loading the console program from the disk to the LSI-11 RAM. When loading is completed, the console program is executed.

4. The console boot displays the VAX-11 status. During this operation, the following data is printed on the terminal

   ```
   CPU HALTED, SOMM CLR
   ```

5. The console is placed in the console I/O mode of operation. The CPU is initialized to the architecture defined state. When the initializing of the CPU is completed, the following data is printed on the terminal

   ```
   INIT SEQ DONE
   HALTED AT 00000000
   ```

6. The WCS/ECO microcode is loaded from the disk to the writable control store (WCS) area in the CPU subsystem. When the WCS is loaded, the following data is printed on the terminal

   ```
   (RELOADING WCS)
   LOAD DONE, NNN BYTES LOADED
   ```

7. The console program tests the status of the AUTO RESTART switch. If the switch is ON, skip to the sequence described in the next paragraph (Step 8). If the switch is OFF, continue to Step 8 of this paragraph.

8. The console remains in the console I/O mode of operation and the terminal prints

   >>>

   This is an input prompt signifying that the console subsystem is booted and operator console command language commands can be entered.

Console and CPU power is applied and the AUTO RESTART switch is ON. This is an automatic boot sequence that is performed when the power is applied. When completed, the VAX-11 is booted and operated in the program I/O mode. The sequence for this bootstrap is as follows.

1-7. This is the same sequence as described in Steps 1 through 7 of the preceding paragraph.

8. The console program initiates the restart referee program (contained in a CPU ROM) and the following is printed by the terminal

   (AUTO RESTART)

9. The restart referee tests to determine if this is a cold start or a warm start.

10. If this boot is part of a warm start, the restart referee goes to VAX/VMS/SEP restart vector and the following is printed on the terminal

    ```
    VAX/VMS/SEP prompt

    The VAX-11 is ready for user input.
    ```

11. If this boot is part of a cold start, the restart referee returns to the console program to obtain the default reading for the system load device.

12. The VAX-11 primary boot (on the RX01 disk) is loaded to VAX-11 memory and started. The console enters program I/O mode.

13. The VAX-11 primary boot initiated the memory sizer. The memory sizer searches memory for 128K bytes of contiguous main memory.

APPENDIX C

BOOTSTRAP HELP FILE

BOOT.HLP - VAX-11/785 Bootstrap Help File
Rev 0.2, 02-Nov-1983

To abort printing, type ^C.

This file describes the input parameters to the bootstrap program VMB.EXE. Normally the bootstrap will look up the file SYSBOOT.EXE in [SYSEXE] on the specified device, load it to memory, and transfer control to it.

To perform a normal, non-interactive boot from standard devices, use the BOOT (abbreviation: B) command as shown below:

```
   BOOT DMn          ! Boot RK07 unit n
      (ie: B DM3 boots RK07 unit 3 using DMBOO.CMD)
   BOOT DBn          ! Boot RM03 or RP06 unit n
      (ie: B DB2 boots RM03/RP06 unit 2 using DBBOO.CMD)
```

The console reads the specified device from the boot command line. If the third character is an integer in the range from 0-7, then it strips off the unit number and deposits it to R3, and executes the appropriate ddBOO.CMD command file. Two-character device names are handled in a similar manner; the console assumes a unit number of zero. Else, the console searches for nnnBOO.CMD, (where nnn was the three-character sequence following the boot command).

The BOOT command is also used to boot with the Diagnostic Supervisor:

```
   BOOT SMn          ! Boot with DS from RK07 unit n
      (ie: B SM2 boots the DS with RK07 unit 2 using SMB00.CMD)
   BOOT SBn          ! Boot with DS from RM/RP unit n
      (ie: B SB5 boots the DS with RP/RM unit 5 using SBBOO.CMD)
```

To perform a boot which stops in SYSBOOT to alter system parameters, the ddGEN.CMD command files are used. To use these files, first deposit the required unit number to R3, then invoke the filename as an indirect command file as shown below:

```
DB-class disks:    >>> D R3 5    ! Selects unit 5.
           >>> @DBGEN    ! Interactive boot file for RM/RP disks.
DM-class disks: >>> D R3 0    ! Selects unit 0.
                >>> @DMGEN    ! Interactive boot file for RK disks.
```

The bootstrap is loaded into memory at least one page above the first available working memory to allow space for the Restart Parameter Block. The address of the base of the bootstrap is passed through SP, the stack pointer, where it also serves as a temporary stack pointer.

Input Parameters:

```
R0 - Boot Device Type Code (RPB$B_DEVTYPE)
     <07:00>  0 => Disk Pack        (RM03/RP04/RP05/RP06/RP07)
              1 => Cartridge Disk   (RK06/RK07)
              2 => RL01/2
             17 => UDA-50
             32 => HSC on CI
             64 => Console Block Storage Device
     <15:08> Reserved to DIGITAL
     <31:16> Device Class Dependent (RPB$W_ROUBVEC)
             UNIBUS  - Optional vector address,
                       0 implies use the default vector
             MASSBUS - N/A

R1 - Boot Device Bus Address:
             <31:04> MBZ
             <03:00> TR number of the Adaptor

R2 - UNIBUS: <31:18> MBZ
             <17:03> Offset UNIBUS address of Control Register
             <02:00> MBZ
     MASSBUS:<31:04> MBZ
             <03:00> Controller/Formatter Number
     CI:     <31:08> MBZ
             <07:00> HSC Port Number (Station Address)

R3 - UNIT:   <31:04> MBZ
             <03:00> Unit Number
```

R4 - Logical Block Number to read as secondary boot block if bit 3 (RPB$V_BBLOCK) is set in R5.

R5 - Software Boot Control flags

Bit	Meaning
0	RPB$V_CONV: Conversational boot. At various points during the system boot procedure, input will be solicited from the console.
1	RPB$V_DEBUG: VMS will map the XDELTA debugger into th system page tables of the running system.

2 RPB$V_INIBPT: Initial breakpoint. If <1> is also set then a breakpoint will occur immediately after mapping is enabled.

3 RPB$V_BLOCK: Boot block. The secondary boot block (whose logical block number is in R4) will be read and control will be transferred to it.

4 RPB$V_DIAG: Diagnostic boot. Boot the Diagnostic Supervisor ([SYSMAINT]DIAGBOOT.EXE).

5 RPB$V_BOOBPT: Bootstrap breakpoint. Stops the primary and secondary bootstraps at a breakpoint before testing memory.

6 RPB$V_HEADER: Image header. If set, the transfer address for the secondary boot image will be taken from its image header. If NOT set, transfers control t the first byte of the secondary boot file.

7 RPB$V_NOTEST: Memory test inhibit. Sets a bit in PFN bit map for each page of memory present but does not test the memory.

8 RPB$V_SOLICT: VMB will prompt for the name of the secondary bootstrap file.

9 RPB$V_HALT: A HALT instruction will be executed before control transfers to the secondary bootfile.

10 RPB$V_NOPFND: (Not used)

11 RPB$V_MPM: Multi-port memory is to be used for the total executive memory requirement. No local memory is used. If bit <5> is also on, Diagnostic Supervisor will enter "autotest" mode.

12 RPB$V_USEMPM: The multiport memory should be used in addition to local memory as single memory.

13 RPB$V_MEMTEST: More exhaustive test of memory should be used when checking for uncorrectable errors.

14 RPB$V_FINDMEM: Use MA780 memory when MS780 memory is insufficient for booting.

31:28 RPB$V_TOPSYS: Specifies the top level directory number for system disks with multiple systems.

SP - ADDRESS+(^X200) of first working 64Kb memory region usable as both stack pointer and pointer to good memory.

Output parameters:

R10 - Base address of region containing secondary bootstrap

R11 - Pointer to Restart Parameter Block (RPB)

SP - Stack Pointer

PR$_SCBB - System Control Block base register

Memory layout at start of secondary bootstrap:

```
+---------------------------------------+ :BASE
|                                       |
|   Restart Parameter Block (RPB)       |
|                                       |
+---------------------------------------+ :BASE+^X200
|                                       |
|   Primary Bootstrap Code              |
|                                       |
+---------------------------------------+ :PR$_SCBB
|                                       |
|   System Control Block                |
|                                       |
+---------------------------------------+ :PFNMAP
|                                       |
|   PFN Bitmap                          |
|                                       |
+---------------------------------------+ :PFNMAP+^X800
|                                       |
|   Bootstrap Stack                     |
|                                       |
+---------------------------------------+ :(SP)
|                                       |
|   Secondary Bootstrap Code            |
|                                       |
+---------------------------------------+
```

APPENDIX D

CONSOLE HELP FILE

```
CONSOL.HLP - VAX-11/785 Console Help File
Rev: 0.2, 03-Nov-1983
Copyright (c) 1983, Digital Equipment Corporation,
Tewksbury, Massachusetts

To abort printing, type ^C.

For boot information, type '@BOOT.HLP'
For error message help, type '@ERROR.HLP'
For Remote Diagnostics access help, type '@REMOTE.HLP'

Syntax:   All commands are terminated by carriage return.
          All commands may be abbreviated to the capitalized letters.
            (ie: "CLear SOmm" may be abbreviated to "CL SO")
          Angle brackets (<>) signify required arguments.
          Square brackets ([]) signify optional arguments.


'Boot'          - Boot the CPU from default device (DEFBOO.CMD)

'Boot <ddn>'    - If n is between 0-7, then the console deposits n to
                  R3 and executes the indirect file 'ddBOO.CMD'.  If
                  only dd is supplied, n is assumed to be zero.  If
                  three characters are supplied and n is not 0-7, then
                  the file 'dddBOO.CMD' is executed.

'CLear Step'    - Disable single-step mode.

'CLear SOmm'    - Clear Stop-On-Micro-Match enable.
                  Note: ID register 21 is the micro-match register.

'Continue'      - Issue a continue to the ISP.
```

```
'Deposit [/qualifier(s)] <address> <data>'
              - Deposit <data> to <address>.

                Qualifiers are:

                    Data length:

                      '/B'    = Byte (8 bits).
                      '/W'    = Word (2 bytes).
                      '/L'    = Longword (2 words).
                      '/Q'    = Quadword (4 words).

                    Address space:

                      '/CON'  = Console (LSI-11) memory.
                      '/G'    = General registers 0 - F (R0 - PC).
                      '/I'    = Internal (processor) registers.
                      '/ID'   = IDbus registers.
                      '/P'    = Physical memory.
                      '/V'    = Virtual memory.
                      '/VB'   = VBus registers.

                    Repeat count:

                      '/N:<number' = Next count.  Repeat the examine
                                     or deposit <number> times.

                <address> may be a number, or one of the following:

                    'R0', 'R1', ..., 'R11', 'AP', 'FP', 'SP', 'PC'
                    'PSL'
                    '*'      (Last address)
                    '+'      (Next address)
                    '-'      (Preceding address)
                    '@'      (Use last exam/dep data as address)
                    <number> (String of digits in current default
                             radix, or prefixed by a radix operator:
                             (%O for octal, %X for hex)).
```

```
'DIRectory'     - Print a directory of file on floppy in CS1:.

'ENable DX1:'   - Enable console software to access floppy drive 1 on
                  those systems with dual floppies.

'Examine [/qualifier(s)] <address>'
                - Examine contents of <address>.
                  * See 'Deposit' for information on qualifiers and add

'Examine IR'    - Examine instruction reg (IR).  Displays op-code,
                  specifier, execution point counter.

'Halt'          - Halt the ISP.

'HElp'          - Print this file (CONSOL.HLP).

'Initialize'    - Initialize the CPU.

'LInk'          - Enter command linking mode (prompt becomes "<<<").
                  All commands typed by the user while linking are
                  stored in an indirect command file for later
                  execution.  ^C ends linking.  (See also 'Perform'.)

'Load [/qualifier(s)] [filename]'
                - Load the specified file to main memory starting at
                  address 0.  If the '/WCS' qualifier is not used, the
                  filename parameter is required.

                  Qualifiers are:

                    '/Start:<adrs>'  Specifies the starting address.
                    '/WCS'           Load to control store.  (If this
                                     qualifier is used, the default
                                     filename is SSUnnn.WCS.)

'Next [number]'
                - Execute <number> (default = 1) single-step cycles.
                  Type of step depends on last 'SET STEP' command.

'Perform'       - Execute a file of linked commands previously
                  generated via a 'LInk' command.

'Qclear <address>'
                - Quad clear <address>, which is forced to a quad word
                  boundary.  (Clears ECC errors.)
```

```
'REboot'        - Reload the console software.

'Repeat <command>'
                - Repeatedly execute <command> until stopped by a ^C.

'SEt Clock <speed>'
                - Set CPU clock speed to Fast, Slow, or Normal.

'SEt Default <option[,option,...]>
                - Set console defaults:

                  CONsole, General, IDbus, Internal, Physical, VBus,
                  Virtual; Byte, Word, Long, Quad; Octal, Hex

'SEt Reloc:<num>'
                - Put <num> into console relocation register.
                  Relocation register is added to effective address of
                  physical and virtual examines and deposits.

'SEt SOmm'      - Set Stop-On-Micro-Match enable.  (Stop when UPC
                  equals the microaddress deposited into ID 21.)

'SEt Step instruction'
                - Enter single instruction mode [SEt Step default].

'SEt Step State'
                - Enter single time-state mode.

'SEt Terminal Fill:<number>'
                - Set fill count for # of blanks written to the
                  terminal after <CR><LF>.

'SEt Terminal Program'
                - Put console terminal into 'program I/O' mode.

'SHow'          - Show console and CPU state.

'SHow Version' - Show microcode and console version numbers.

'Start <{address} or {@}>'
                - Initialize the CPU, deposit <address> to PC, and
                  issue a continue to the ISP.  If the '@' is used, the
                  contents of the last examined address is used as the
                  starting address.

'Test'          - Run micro-diagnostics.

'Test/Com'      - Load micro-diagnostics, wait for commands.
```

'Unjam' - Unjam the SBI.

'WAit Done' - When executed from an indirect command file, stop command file execution until:

A) A 'done' signal is received from the program running in the VAX (command file execution wil continue), or

B) The VAX-11/785 halts or operator ^C's.

'^P' - Put console terminal into 'console I/O' mode (unless mode switch is set to 'disable').

'@<filename>' - Process an indirect command file (<filename>.CMD).

APPENDIX E

ERROR HELP FILE

```
ERROR.HLP - VAX-11/785 Error Message Help File
Rev: 0.1, 10-Oct-1983
Copyright (c) 1983, Digital Equipment Corporation, Tewksbury,
Massachusetts

To abort printing, type ^C.

This file lists all the console error messages, indicating probable
cause and possible corrective procedures.

?'<text-string>' IS INCOMPLETE
    The specified command requires additional parameters.

?'<text-string>' IS INCORRECT
    The <text-string> is not recognized as a valid command.

?CAN'T DISABLE BOTH FLOPPIES, FUNCTION ABORTED
    An attempt was made to disable both the remote and local floppy.

?CHM ERR
    A change mode instruction was attempted from the interrupt stack.

?CPU CLK STOP, COMMAND ABORTED
    A console command that requires the CPU clock to be running was
    issued while the clock was stopped.  (Clear step mode; re-issue
    command.)

?CPU DBLE-ERR HLT
    A machine check occurred before a previous machine check had been
    handled, causing the CPU to execute a 'double error' halt.
    (Examine ID registers 30-3F (hex); data indicates cause of machine
    check.)

?CPU NOT IN CONSOLE WAIT LOOP, COMMAND ABORTED
    A console command requiring assistance from the CPU was issued
    while CPU was not in the console service loop.  (Halt CPU;
    re-issue command.)

?FATAL ERROR: WCS NOT LOADED
    The WCS-Load operation was aborted, either by ^C or internal
    error.  (Reload WCS [use LOAD/WCS command or restart machine].)

?FILE NAME ERR
    An illegal filename was used.
```

?FILE NOT FOUND

The 'LOAD' or '@' command could not find the specified file on th€ currently loaded floppy; the 'HELP' command could not find CONSOL.HLP; the 'BOOT' command could not find ddBOO.CMD; the 'LOAD/WCS' command could not find SSUnnn.WCS or the specified WCS file.

?FLOPPY ERR, CODE=n

The console floppy driver detected an error. Codes are as follows:

Code = 1	Floppy hardware error (CRC, parity, etc.)
Code = 2	File not found
Code = 3	Floppy driver queue overflow
Code = 4	Console software requested an illegal sector number

?FLOPPY ERROR ON BOOT

A floppy error was detected while attempting a console boot. (Retry.)

?FLOPPY NOT READY

The console floppy drive failed to become ready when booting. (Retry.)

?ILL I/E VEC

The CPU detected an illegal interrupt/exception vector.

?IND-COM ERR

The indirect command file contained a format error, such as:

- More than 80 characters in an indirect command line

- A command line did not end with a carriage return

INT PENDING

This warning indicates that an error was pending at the time that a console-requested halt was performed. Continue CPU to clear interrupt.

?INT-REG ERR

A micro-error occurred while attempting to reference a CPU internal (processor) register. An illegal address will cause thi€ error.

?INT-STK INVLD

The CPU halted because the interrupt stack was marked invalid.

?MEM-MAN FAULT, CODE=nn
A virtual examine or deposit caused an error in the memory management micro-routine. 'nn' is a one byte error code returned by the routine, with the following bit assignments:

Bit 0 = Length violation
Bit 1 = Fault was on PTE reference
Bit 2 = Write or modify intent
Bit 3 = Access violation

Bits 4 through 7 should be ignored.

?MIC-ERR ON FUNCTION
A micro-error occurred in the CPU while servicing a console request. SBI error registers are dumped after this message is printed.

?MICRO-ERR, CODE=n
The micro-error code returned by the CPU is not in the range of recognized error codes. 'n' is the code that was returned by the CPU.

?MICRO-MACHINE TIME OUT
Indicates that the VAX-11/785 micromachine has failed to strobe interrupts within the maximum time period allowed.

?NO BOOT ON FLOPPY
Console attempted to boot from a floppy that does not contain a valid boot block. (Change floppy disk.)

?NO CPU RESPONSE
Console timed-out waiting for a response from the CPU. (Retry; indicates possible CPU-related hardware fault.)

?Q-BLKD
Console's terminal output queue is blocked. Console will reboot.

?TRAP-4, RESTARTING CONSOLE
The console took a time-out trap. Console will restart.

?UNEXPECTED TRAP
MOUNT CONSOLE FLOPPY, THEN TYPE ^C
Console trapped to an unused vector. Console reboots when ^C typed.

?WARNING - WCS & PCS VER MISMATCH

The WCS file loaded from the floppy is not compatible with the PCS ROMs on the JCS module. (Get correct WCS version and reload.)

?WCS LOAD CHECKSUM ERROR

Either the WCS file on the console floppy is corrupted (or is not of standard checksum-encoded format) or the console had a read-error while loading the file. (Try reloading WCS; get a new console floppy.)

APPENDIX F

REMOTE HELP FILE

```
REMOTE.HLP - VAX-11/785 Remote Diagnostics Help File
Rev: 0.1, 10-Oct-1983
Copyright (c) 1983, Digital Equipment Corporation, Tewksbury,
Massachusetts

To abort printing, type ^C.

This file contains information on the Remote Diagnostics console
commands.  (VAX-11/785 Remote Diagnostics option (RD785) is available
under separate license from Digital Equipment Corporation.)

'DISable Carrier Error' - Causes console to inhibit printing of
                          carrier lost message when loss of carrier
                          detected.

'DISable Echo'          - Inhibits echo of characters typed in talk
                          mode.

'DISable Local Copy'    - Disable local terminal from receiving copy
                          of output to remote terminal.

'DISable Local Floppy'  - (Affects protocol operation only) only the
                          remote floppy is searched when opening
                          files.

'DISable Remote Floppy' - On an attempt to open a file, only the
                          directory of the local floppy will be
                          searched.  This command and 'DISable Local
                          Floppy' are mutually exclusive.

'ENable Carrier Error'  - Causes console to print '?CARRIER LOST' when
                          a loss of carrier is detected at remote
                          interface.

'ENable Echo'           - Causes characters typed in talk mode to be
                          echoed to the originating terminal.

'ENable Local COPy'     - Causes the local terminal to get a copy of
                          output being sent to remote terminal.

'ENable Local CONtrol'  - Allows local terminal to control system when
                          console switch is in remote position.
                          Disabled by a ^P from the remote terminal.
```

'ENable Local Floppy' - (Affects protocol operation only) on an attempt to open a file, the directory of local floppy will be searched first. If the file is not found, the remote floppy's directory is searched for file.

'ENable Remote Floppy' - Allows the directory of the remote floppy to be searched on an attempt to open a file.

'ENable Talk' - Establish communication between local and remote terminal. Keys entered at one terminal are echoed on the other. ^P exits talk.

APPENDIX G
GLOSSARY

abort
An exception that occurs in the middle of an instruction that could leave the registers and memory in a state, such that the instruction cannot be restarted.

absolute indexed mode
An indexed addressing mode in which the base operand is addressed in absolute mode.

absolute mode
In absolute mode addressing, the program counter (PC) is used as the register in autoincrement deferred mode. The PC contains the address of the actual operand.

absolute time
Time values expressing a specific date (month, day, and year) and time of day. Absolute time values are always expressed in the system as positive numbers.

access mode

1. Any of the four processor access modes in which software executes. Processor access modes are, in order from most to least privileged and protected: kernel (mode 0), executive (mode 1), supervisor (mode 2), and user (mode 3). When the processor is in kernel mode, the executing software has complete control of, and responsibility for, the system. When the processor is in any other mode, the processor is inhibited from executing privileged instructions. The processor status longword contains the current access mode field. The operating system uses access modes to define protection levels for software executing in the context of a process. For example, the executive runs in kernel and executive mode and is most protected. The command interpreter is less protected and runs in supervisor mode. The debugger runs in user mode and is not more protected than normal user programs.

2. See record access mode.

access type

1. The way in which the processor accesses instruction operands. Access types are: read, write, modify, address, and branch.

2. The way in which a procedure accesses its arguments.

access violation
An attempt to reference an address that is not mapped into virtual memory or an attempt to reference an address that is not accessible by the current access mode.

account name
A string that identifies a particular account used to accumulate data on a job's resource use. This name is the user's accounting charge number, not the user's user identification code (UIC).

address
A number used by the operating system and user software to identify a storage location. See also virtual address and physical address.

address access type
The specified operand of an instruction is not directly accessed by the instruction. The address of the specified operand is the actual instruction operand. The context of the address calculation is given by the data type of operand.

addressing mode
The way in which an operand is specified; for example, the way in which the effective address of an instruction operand is calculated using the general registers. The basic general register addressing modes are called: register, register deferred, autoincrement, autoincrement deferred, autodecrement, displacement, and displacement deferred. In addition, there are six indexed addressing modes using two general registers and literal mode addressing. The PC addressing modes are called: immediate (for register deferred mode using the PC), absolute (for autoincrement deferred mode using the PC), and branch.

address space
The set of all possible addresses available to a process. Virtual address space refers to the set of all possible virtual addresses. Physical address space refers to the set of all possible physical addresses sent out on the SBI.

allocate device
To reserve a particular device unit for exclusive use. A user process can allocate a device only when that device is not allocated by any other process.

alphanumeric character
An upper or lower case letter (A–Z, a–z), a dollar sign ($), an underscore (__), or a decimal digit (0.9).

American standard code for information interchange (ASCII)
A set of 8-bit binary numbers representing the alphabet, punctuation, numerals, and other special symbols used in text representation and communications protocol.

ancillary control process (ACP)
A process that acts as an interface between user software and an I/O driver. An ACP provides functions supplemental to those performed in the driver, such as file and directory management. Three examples of ACPs are: the Files-11 ACP (F11ACP), the magnetic tape ACP (MTACP), and the networks ACP (NETACP).

argument pointer
General register 12 (R12). By convention, argument pointer contains the address of the base of the argument list for procedures initiated using the CALL instructions.

assign a channel
To establish the necessary software linkage between a user process and a device unit before a user process can transfer any data to or from that device. A user process requests the system to assign a channel and the system returns a channel number.

asynchronous
A mode of activity that operates without respect to a clock. For example, asynchronous hardware uses ready and done signals to schedule operations rather than time intervals. If two activities are asynchronous, the second can begin at a random interval with respect to the first.

asynchronous record operation
A mode of record processing in which a user program can continue to execute after issuing a record retrieval or storage request without having to wait for the request to be fulfilled.

asynchronous system trap (AST)
A software-simulated interrupt to a user-defined service routine. ASTs enable a user process to be notified asynchronously with respect to its execution of the occurrence of a specific event. If a user process has defined an AST routine for an event, the system interrupts the process and executes the AST routine when that event occurs. When the AST routine exits, the system resumes the process at the point where it was interrupted.

asynchronous system trap level (ASTLVL)
A value kept in an internal processor register that is the highest access mode for which an AST is pending. The AST does not occur until the current access mode drops in priority (raises in numeric value) to a value greater than or equal to ASTLVL. Thus, an AST for an access mode will not be serviced while the processor is executing in a higher priority access mode.

authorization file
See user authorization file.

autodecrement indexed mode
An indexed addressing mode in which the base operand specifier uses autodecrement mode addressing.

autodecrement mode
In autodecrement mode addressing, the contents of the selected register are decremented and the result is used as the address of the actual operand for the instruction. The contents of the register are decremented according to the data type context of the register: 1 for byte, 2 for word, 4 for longword and floating, 8 for quadword and double floating.

autoincrement deferred indexed mode
An indexed addressing mode in which the base operand specifier uses autoincrement deferred mode addressing.

autoincrement deferred mode
In autoincrement deferred mode addressing, the specified register contains the address of a longword that contains the address of the actual operand. The contents of the register are incremented by 4 (the number of bytes in a longword). If the PC is used as the register, this mode is called absolute mode.

autoincrement indexed mode
An indexed addressing mode in which the base operand specifier uses autoincrement mode addressing.

autoincrement mode
In autoincrement mode addressing, the contents of the specified register are used as the address of the operand, then the contents of the register are incremented by the size of the operand.

balance set
The set of all process working sets currently resident in physical memory. The processes whose working sets are in the balance set have memory requirements that balance with available memory. The balance set is maintained by the system swapper process.

base operand address
The address of the base of a table or array referenced by index mode addressing.

base operand specifier
The register used to calculate the base operand address of a table or array referenced by index mode addressing.

base priority
The process priority that the system assigns a process when it is created. The scheduler never schedules a process below its base priority. The base priority can be modified only by the system manager or the process itself.

base register
A general register used to contain the address of the first entry in a list, table, array, or other data structure.

binding
See linking.

bit string
See variable-length bit field.

block

1. The smallest addressable unit of data that the specified device can transfer in an I/O operation (512 contiguous bytes for most disk devices).

2. An arbitrary number of contiguous bytes used to store logically related status, control, or other processing information.

block I/O
A data accessing technique in which the program manipulates the blocks (physical records) that make up a file, instead of its logical records.

bootstrap block
A block in the index file on a system disk that contains a program that can load the operating system into memory and start its execution.

branch access type
An instruction attribute which indicates that the processor does not reference an operand address, but that the operand is a branch displacement. The size of the branch displacement is given by the data type of the operand.

branch mode
In branch addressing mode, the instruction operand specifier is a signed byte or word displacement. The displacement is added to the contents of the updated PC (which is the address of the first byte beyond the displacement), and the result is the branch address.

bucket
A storage structure of 1 to 32 blocks that is used to store and transfer blocks of data in files with a relative file organization.

bucket locking
A facility that prevents access to any record in a bucket by more than one user until that user releases the bucket.

buffered I/O
See system buffered I/O.

bug check
The operating system's internal diagnostic check. When so indicated by the check the system logs the failure and crashes the system.

byte
A byte is eight contiguous bits starting on an addressable byte boundary. Bits are numbered from the right, 0 through 7, with bit 0 the low-order bit. When interpreted arithmetically, a byte is a two's complement integer with significance increasing from bits 0 through 6. Bit 7 is the sign bit. The value of the signed integer is in the range -128 to 127 decimal. When interpreted as an unsigned integer, significance increases from bits 0 through 7 and the value of the unsigned integer is in the range 0 to 255 decimal. A byte can be used to store one ASCII character.

cache memory
A small, high-speed memory placed between slower main memory and the processor. A cache increases effective memory transfer rates and processor speed. It contains copies of data recently used by the processor, and fetches several bytes of data from memory in anticipation that the processor will access the next sequential series of bytes.

call frame
See stack frame.

call instructions
The processor instructions CALLG (call procedure with general argument list) and CALLS (call procedure with stack argument list).

call stack
The stack, and conventional stack structure, used during a procedure call. Each access mode of each process context has one call stack, and interrupt service context has one call stack.

channel
A logical path connecting a user process to a physical device unit. A user process requests the operating system to assign a channel to a device so the process can transfer data to or from that device.

character
A symbol represented by an ASCII code. See also alphanumeric character.

character string
A contiguous set of bytes. A character string is identified by two attributes: an address and a length. Its address is the address of the byte containing the first character of the string. Subsequent characters are stored in bytes of increasing addresses. The length is the number of characters in the string.

character string descriptor
A quadword data structure used for passing character data (strings). The first word of the quadword contains the length of the character string. The second word can contain type information. The remaining longword contains the address of the string.

cluster
1. A set of contiguous blocks that is the basic unit of space allocation on a Files-11 disk volume.
2. A set of pages brought into memory in one paging operation.
3. An event flag cluster.

command
An instruction, generally an English word, typed by the user at a terminal or included in a command file that requests the software monitoring a terminal or reading a command file to perform some well-defined activity. For example, typing the COPY command requests the system to copy the contents of one file into another file.

command file
A file containing command strings. See also command procedure.

command interpreter
Procedure-based system code that executes in supervisor mode in the context of a process to receive, syntax check, and parse commands typed by the user at a terminal or submitted in a command file.

command parameter
The positional operand of a command delimited by spaces, such as a file specification, option, or constant.

command procedure
A file containing commands and data that the command interpreter can accept in lieu of the user typing the commands individually on a terminal.

command string
A line (or set of continued lines), normally terminated by typing the carriage return key, containing a command and, optionally, information modifying the command. A complete command string consists of a command, its qualifiers, if any, and its parameters (file specifications, for example), if any, and their qualifiers, if any.

common
A FORTRAN term for a program section that contains only data.

common event flag cluster
A set of 32 event flags that enables cooperating processes to post event notification to each other. Common event flag clusters are created as they are needed. A process can associate with up to two common event flag clusters.

compatibility mode
A mode of execution that enables the central processor to execute nonprivileged PDP-11 instructions. The operating system supports compatibility mode execution by providing an RSX-11M programming environment for an RSX-11M task image. The operating system compatibility mode procedures reside in the control region of the process executing a compatibility mode image. The procedures intercept calls to the RSX-11M executive and convert them to the appropriate operating system functions.

compiler
A program that translates a program written in a high-level language (such as FORTRAN or BASIC) into an object program.

condition
An exception condition detected and declared by software. For example, see failure exception mode.

condition codes
Four bits in the processor status word that indicate the results of previously executed instructions.

condition handler
A procedure that is called upon by the system when an exception condition occurs. When an exception condition occurs, the operating system searches for a condition handler and, if found, initiates the condition handler immediately. The condition handler may perform some action to change the situation that caused the exception condition and continue execution for the process that incurred the exception condition. Condition handlers execute in the context of the process at the access mode of the code that incurred the exception condition.

condition value
A 32-bit quantity that uniquely identifies an exception condition.

context
The environment of an activity. See also process context, hardware context, and software context.

context indexing
The ability to index through a data structure automatically because the size of the data type is known and used to determine the offset factor.

context switching
Interrupting the activity in progress and switching to another activity. Context switching occurs as one process after another is scheduled for execution. The operating system saves the hardware context of the interrupted process in its hardware process control block (PCB) using the save process context instruction. It also loads hardware PCB into the hardware context of another process using the load process context instruction, scheduling that process for execution.

continuation character
A hyphen at the end of a command line signifying that the command string continues on to the next command line.

console
The manual control unit integrated into the central processor. The console includes an LSI-11 microprocessor, a DX01 floppy disk, and a serial line interface connected to an LA36 terminal. It enables the operator to start and stop the system, monitor system operation, and run diagnostics.

console terminal
The hard copy terminal (LA36) connected to the central processor console.

control region
The highest addressed half of virtual memory per-process space (the P1 region). Control region virtual addresses refer to the process-related information used by the system to control the process, such as: the kernel, executive, and supervisor stacks, the permanent I/O channels, exception vectors, and dynamically used system procedures (such as the command interpreter and RSX-11M programming environment compatibility mode procedures). The user stack is also normally found in the control region, although it can be located elsewhere.

control region base register (P1BR)
The processor register, or its equivalent in a hardware process control block, that contains the base virtual address of a process control region page table.

control region length register (P1LR)
The processor register, or its equivalent in a hardware process control block, that contains the number of nonexistent page table entries for virtual pages in a process control region.

copy-on-reference
A method used in memory management for sharing data until a process accesses it, in which case it is copied before the access. Copy-on-reference allows sharing of the initial values of a global section whose pages have read/write access but contain pre-initialized data available to many processes.

counted string
A data structure consisting of a byte-sized length followed by the string. Although a counted string is not used as a procedure argument, it is a convenient representation in memory.

current access mode
The processor access mode of the currently executing software. The current mode field of the processor status longword indicates the access mode of the currently executing software.

cylinder
The tracks at the same radius on all recording surfaces of a disk.

data base
1. All the occurrences of data described by a data base management system.
2. A collection of related data structures.

data structure
Any table, list, array, queue, or tree whose format and access conventions are well-defined for reference by one or more images.

data type
In general, the way in which bits are grouped and interpreted. In reference to the processor instructions, the data type of an operand identifies the size of the operand and the significance of the bits in the operand. Operand data types include: byte, word, longword, and quadword integer, floating and double floating, character string, packed decimal string, and variable-length bit field.

deferred echo
Refers to the fact that terminal echoing does not occur until a process is ready to accept input entered by type ahead.

delta time
A time value expressing an offset from the current date and time. Delta times are always expressed in the system as negative numbers whose absolute value is used as an offset from the current time.

demand zero page
A page, typically of an image stack or buffer area, that is initialized to contain all zeros when dynamically created in memory as a result of a page fault. This feature eliminates the waste of disk space that would otherwise be required to store blocks (pages) that contain only zeros.

descriptor
A data structure used in calling sequences for passing argument types, addresses, and other optional information. See character string descriptor.

detached process
A process that has no owner. The parent process of a tree of subprocesses. Detached processes are created by the job controller when a user logs on the system or when a batch job is initiated. The job controller does not own the user processes it creates; these processes therefore are detached.

device
The general name for any physical terminus or link connected to the processor that is capable of receiving, storing, or transmitting data. Card readers, line printers, and terminals are examples of record-oriented devices. Magnetic tape devices and disk devices are examples of mass storage devices. Terminal line interfaces and interprocessor links are examples of communications devices.

device interrupt
An interrupt received on interrupt priority level 16 through 23. Device interrupts can be requested only by devices, controllers, and memories.

device name
The field in a file specification that identifies the device unit on which a file is stored. Device names also include the mnemonics that identify an I/O peripheral device in a data transfer request. A device name consists of a mnemonic followed by a controller identification letter (if applicable), followed by a unit number (if applicable). A colon (:) separates it from following fields.

device queue
See spool queue.

device register
A location in device controller log used to request device functions (such as I/O transfers) and/or report status.

device unit
One drive, and its controlling logic, of a mass storage device system. A mass storage system can have several drives connected to it.

diagnostic
A program that tests logic and reports any faults it detects.

direct I/O
An I/O operation in which the system locks the pages containing the associated buffer in memory for the duration of the I/O operation. The I/O transfer takes place directly from the process buffer. Contrasts with system buffered I/O.

direct mapping cache
A cache organization in which only one address comparison is needed to locate any data in the cache because any block of main memory data can be placed in only one possible position in the cache. Contrasts with fully associative cache.

directory
A file used to locate files on a volume that contains a list of file names (including type and version number) and their unique internal identifications.

directory name
The field in a file specification that identifies the directory file in which a file is listed. The directory name begins with a left bracket ([or <) and ends with a right bracket (] or >).

displacement deferred indexed mode
An indexed addressing mode in which the base operand specifier uses displacement deferred mode addressing.

displacement deferred mode
In displacement deferred mode addressing, the specifier extension is a byte, word, or longword displacement. The displacement is sign extended to 32 bits and added to a base address obtained from the specified register. The result is the address of a longword that contains the address of the actual operand. If the PC is used as the register, the updated contents of the PC are used as the base address. The base address is the address of the first byte beyond the specifier extension.

displacement indexed mode
An indexed addressing mode in which the base operand specifier uses displacement mode addressing.

displacement mode
In displacement mode addressing, the specifier extension is a byte, word, or longword displacement. The displacement is sign extended to 32 bits and added to a base address obtained from the specified register. The result is the address of the actual operand. If the PC is used as the register, the updated contents of the PC are used as the base address. The base address is the address of the first byte beyond the specifier extension.

double floating datum
Eight contiguous bytes (64 bits), starting on an addressable byte boundary, which are interpreted as containing a floating point number. The bits are labeled from right to left, 0 to 63. A four-word floating-point number is identified by the address of the byte containing bit 0. Bit 15 contains the sign of the number. Bits 14 through 7 contain the excess 128 binary exponent. Bits 63 through 16 and 6 through 0 contain a normalized 56-bit fraction with the redundant most significant fraction bit not represented. Within the fraction, bits of decreasing significance go from 6 through 0, 31 through 16, 47 through 32, then 63 through 48. Exponent values of 1 through 255 in the 8-bit exponent field represent true binary exponents of −128 to 127. An exponent value of 0, together with a sign bit of 0, represent a floating value of 0. An exponent value of 0 with a sign bit of 1 is a reserved representation: floating-point instructions processing this value return a reserved operand fault. The value of a floating datum is in the approximate range (+ or −) 0.29×10^{-38} to 1.7×10^{38}. The precision is approximately one part in 2^{55} or sixteen decimal digits.

drive
The electromechanical unit of a mass storage device system on which a recording medium (disk cartridge, disk pack, or magnetic tape reel) is mounted.

driver
The set of code that handles physical I/O to a device.

dynamic access
A technique in which a program switches from one record access mode to another while processing a file.

echo
A terminal handling characteristic in which the characters typed by the terminal user on the keyboard are also displayed on the screen or printer.

effective address
The address obtained after indirect or indexing modifications are calculated.

entry mask
A word whose bits represent the registers to be saved or restored on a subroutine or procedure call using the call and return instructions.

entry point
A location that can be specified as the object of a call. It contains an entry mask and exception enables known as the entry point mask.

equivalence name
The string associated with a logical name in a logical name table. An equivalence name can be, for example, a device name, another logical name, or a logical name concatenated with a portion of a file specification.

error logger
A system process that empties the error log buffers and writes the error messages into the error file. Errors logged by the system include memory system errors, device errors and timeouts, and interrupts with invalid vector addresses.

escape sequence
An escape is a transition from the normal mode of operation to a mode outside the normal mode. An escape character is the code that indicates the transition from normal to escape mode. An escape sequence refers to the set of character combinations starting with an escape character that the terminal transmits without interpretation to the software set up to handle escape sequences.

event
A change in process status or an indication of the occurrence of some activity that concerns an individual process or cooperating processes. An incident reported to the scheduler that affects the ability of a process to execute. Events can be synchronous with the process's execution (a wait request), or they can be asynchronous (I/O completion). Some other events include: swapping, wake request, page fault.

event flag
A bit in an event flag cluster that can be set or cleared to indicate the occurrence of the event associated with that flag. Event flags are used to synchronize activities in a process or among many processes.

event flag cluster
A set of 32 event flags that are used for event posting. Four clusters are defined for each process: two process-local clusters, and two common event flag clusters. Of the process-local flags, eight are re served for system use.

exception
An event detected by the hardware (other than an interrupt or jump, branch, case, or call instruction) that changes the normal flow of instruction execution. An exception is always caused by the execution of an instruction or set of instructions (whereas an interrupt is caused by an activity in the system independent of the current instruction). There are three types of hardware exceptions: traps, faults, and aborts. Examples are: attempts to execute a privileged or reserved instruction, trace traps, compatibility mode faults, breakpoint instruction execution, and arithmetic traps such as overflow, underflow, and divide by zero.

exception condition
A hardware- or software-detected event other than an interrupt or jump, branch, case, or call instruction that changes the normal flow of instruction execution.

exception dispatcher
An operating system procedure that searches for a condition handler when an exception condition occurs. If no exception handler is found for the exception or condition, the image that incurred the exception is terminated.

exception enables
See trap enables.

exception vector
See vector.

executable image
An image that is capable of being run in a process. When run, an executable image is read from a file for execution in a process.

executive
The generic name for the collection of procedures included in the operating system software that provides the basic control and monitoring functions of the operating system.

executive mode
The second most privileged processor access mode (mode 1). The record management services (RMS) and many of the operating system's programmed service procedures execute in executive mode.

exit
An image exit is a rundown activity that occurs when image execution terminates either normally or abnormally. Image rundown activities include deassigning I/O channels and disassociation of common event flag clusters. Any user-or-system-specified exit handlers are called.

exit handler
A procedure executed when an image exits. An exit handler enables a procedure that is not on the call stack to gain control and clean up procedure-own data bases before the actual image exit occurs.

extended attribute block (XAB)
A record management service user data structure that contains additional file attributes beyond those expressed in the file access block (FAB), such as boundary types (aligned on cylinder, logical block number, virtual block number) and file protection information.

extension
The amount of space to allocate at the end of a file each time a sequential write exceeds the allocated length of the file.

extent
The contiguous area on a disk containing a file or a portion of a file. Consists of one or more clusters.

failure exception mode
A mode of execution selected by a process indicating that it wants an exception condition declared if an error occurs as the result of a system service call. The normal mode is for the system service to return an error status code for which the process must test.

fault
A hardware exception condition that occurs in the middle of an instruction and leaves the registers and memory in a consistent state, such that elimination of the fault and restarting the instruction will give correct results.

field
1. See variable-length bit field.
2. A set of contiguous bytes in a logical record.

file access block (FAB)
A record management service user data structure that represents a request for data operations related to the file as a whole, such as OPEN, CLOSE, or CREATE.

file header
A block in the index file describing a file on a Files-11 disk structure. The file header identifies the locations of the file's extents. There is a file header for every file on the disk.

file name
The field preceding a file type in a file specification that contains a 1- to 9-character logical name for a file.

file name extension
See file type.

file organization
The particular file structure used to record the data comprising a file on a mass storage medium. Record management services (RMS) file organizations are: sequential, relative, direct, and indexed.

Files-11
The name of the on-disk structure used by the RSX-11, IAS and VAX operating systems. Volumes created under this structure are transportable between these operating systems.

file specification
A unique name for a file on a mass storage medium. It identifies the node, the device, the directory name, the file name, the file type, and the version number under which a file is stored.

file structure
The way in which the blocks forming a file are distributed on a disk or magnetic tape to provide a physical accessing technique suitable for the way in which the data in the file is processed.

file system
A method of recording, cataloging, and accessing files on a volume.

file type
The field in a file specification that is preceded by a period or a dot (.) and consists of a zero- to three-character type identification. By convention, the type identifies a generic class of files that have the same use or characteristics, such as ASCII text files, binary object files, etc.

fixed control area
An area associated with a variable length record available for controlling or assisting record access operations. Typical uses include line numbers and printer format control information.

fixed length record format
A file format in which all records have the same length.

floating (point) datum
Four contiguous bytes (32 bits) starting on an addressable byte boundary. The bits are labeled from right to left from 0 to 31. A two-word floating-point number is identified by the address of the byte containing bit 0. Bit 15 contains the sign of the number. Bits 14 through 7 contain the excess 128 binary exponent. Bits 31 through 16 and 6 through 0 contain a normalized 24-bit fraction with the redundant most significant fraction bit not represented. Within the fraction, bits of decreasing significance go from bit 6 through 0, then 31 through 16. Exponent values of 1 through 255 in the 8-bit exponent field

represent true binary exponents of -128 to 127. An exponent value of 0, together with a sign bit of 0, represent a floating value of 0. An exponent value of 0, with a sign bit of 1, is a reserved representation: floating-point instructions processing this value return a reserved operand fault. The value of a floating datum is in the approximate range (+ or -) 0.29 × 10^{-38} to 1.7 × 10^{38}. The precision is approximately one part in 2^{23} or seven decimal digits.

foreign volume
Any volume other than a Files-11 formatted volume that may or may not be file structured.

fork process
A dynamically created system process such as a process that executes device driver code or the timer process. Fork processes have minimal context. Fork processes are scheduled by the hardware rather than by the software. The timer process is dispatched directly by software interrupt. I/O driver processes are dispatched by a fork dispatcher. Fork processes execute at software interrupt levels and are dispatched for execution immediately. Fork processes remain resident until they terminate.

frame pointer
General register 13 (R13). By convention, the frame pointer (FP) contains the base address of the most recent call frame on the stack.

fully associative cache
A cache organization in which any block of data from main memory can be placed anywhere in the cache. Address comparison must take place against each block in the cache to find any particular block. Contrasts with direct mapping cache.

general register
Any of the sixteen 32-bit registers used as the primary operands of the native mode instructions. The general registers include 12 general purpose registers that can be used as accumulators, as counters, and as pointers to locations in main memory, and the frame pointer (FP), argument pointer (AP), stack pointer (SP), and program counter (PC) registers.

generic device name
A device name that identifies the type of device but not a particular unit; a device name in which the specific controller and/or unit number is omitted.

giga
Metric term used to represent the number 1 followed by nine zeros. Equivalent to billions.

global page table
The page table containing the master page table entries for global sections.

global section
A data structure (e.g., FORTRAN global common) or sharable image section potentially available to all processes in the system. Access is protected by privilege and/or group number of the user identification code (UIC).

global symbol
A symbol defined in a module that is potentially available for reference by another module. The linker resolves (matches references with definitions) global symbols. Contrasts with local symbol.

global symbol table (GST)
In a library, an index of defined global symbols used to access the modules defining the global symbols. The linker will also put global symbol tables into an image. For example, the linker appends a global symbol table to executable images that are intended to run under the symbolic debugger, and it appends a global symbol table to all sharable images.

group

1. A set of users who have special access privileges to each other's directories and files within those directories (unless protected otherwise), as in the context "system, owner, group, world," where group refers to all members of a particular owner's group.

2. A set of jobs (processes and their subprocesses) that have access privileges to a group's common event flags and logical name tables, and may have mutual process controlling privileges, such as scheduling, hibernation, etc.

group number
The first number in a user identification code (UIC).

hardware context
The values contained in the following registers while a process is executing: the program counter (PC); the processor status longword (PSL); the 14 general registers (R0 through R13); the four processor registers (P0BR, P0LR, P1BR, and P1LR) that describe the process virtual address space; the stack pointer (SP) for the current access mode in which the processor is executing; plus the contents to be loaded in the stack pointer for every access mode other than the current access mode. While a process is executing, its hardware context is continually being updated by the processor. While a process is not executing, its hardware context is stored in its hardware process control block (PCB).

hardware process control block (PCB)
A data structure known to the processor that contains the hardware context when a process is not executing. A process's hardware PCB resides in its process header.

hibernation
A state in which a process is inactive, but known to the system with all of its current status. A hibernating process becomes active again when a wake request is issued. It can schedule a wake request before hibernating, or another process can issue its wake request. A hibernating process also becomes active for the time sufficient to service any asynchronous system trap (AST) it may receive while it is hibernating. Contrasts with suspension.

home block
A block in the index file that contains the volume identification, such as volume label and protection.

image
An image consists of procedures and data that have been bound together by the linker. There are three types of images: executable, sharable, and system.

image activator
A set of system procedures that prepares an image for execution. The image activator establishes the memory management data structures required both to map the image's virtual pages to physical pages and to perform paging.

image exit
See exit.

image I/O segment
That portion of the control region that contains the record management services (RMS) internal file access blocks (IFAB) and I/O buffers for the image currently being executed by a process.

image name
The file name of the file in which an image is stored.

image privileges
The privileges assigned to an image when it is linked. See process privileges.

image section (isect)
A group of program sections (psects) with the same attributes (such as read-only access, read/write access, absolute, relocatable, etc.) that is the unit of virtual memory allocation for an image.

immediate mode
In immediate mode addressing, the program counter (PC) is used as the register in autoincrement mode addressing.

indexed addressing mode
In indexed mode addressing, two registers are used to determine the actual instruction operand; an index register and a base operand specifier. The contents of the index register are used as an index (offset) into a table or array. The base operand specifier supplies the base address of the array (the base operand address or BOA). The address of the actual operand is calculated by multiplying the contents of the index register by the size (in bytes) of the actual operand and adding the result to the base operand address. The addressing modes resulting from index mode addressing are formed by adding the suffix "indexed" to the addressing mode of the base operand specifier: register deferred indexed, autoincrement indexed, autoincrement deferred indexed (or absolute indexed), autodecrement indexed, displacement indexed, and displacement deferred indexed.

indexed file organization
A file organization in which a file contains records and a primary key index (and optionally one or more alternate key indices) used to process the records sequentially by index or randomly by index.

index file
The file on a Files-11 volume that contains the access information for all files on the volume and enables the operating system to identify and access the volume.

index file bit map
A table in the index file of a Files-11 volume that indicates which file headers are in use.

index register
A register used to contain an address offset.

indirect command file
See command procedure.

input stream
The source of commands and data, i.e., the user's terminal, the batch stream, or an indirect command file.

instruction buffer
An 8-byte buffer in the processor used to contain bytes of the instruction currently being decoded and to pre-fetch instructions in the instruction stream. The control logic continuously fetches data from memory to keep the 8-byte buffer full.

interleaving
Assigning consecutive physical memory addresses alternately between two memory controllers.

interprocess communication facility
A common event flag, mailbox, or global section used to pass information between two or more processes.

interrecord gap
A blank space deliberately placed between data records on the recording surface of a magnetic tape.

interrupt
An event other than an exception or branch, jump, case, or call instruction that changes the normal flow of instruction execution. Interrupts are generally external to the process executing when the interrupt occurs. See also device interrupt, software interrupt, and urgent interrupt.

interrupt priority level (IPL)
The interrupt level at which the processor executes when an interrupt is generated. There are 31 possible interrupt priority levels. IPL 1 is lowest, 31 highest. The levels arbitrate contention for processor service. For example, a device cannot interrupt the processor if the processor is currently executing at an interrupt priority level greater than the interrupt priority level of the device's interrupt service routine.

interrupt service routine
The routine executed when a device interrupt occurs.

interrupt stack
The system-wide stack used when executing in interrupt service context. At any time, the processor is either in a process context executing in user, supervisor, executive, or kernel mode; or in system-wide interrupt service context operating with kernel privileges, as indicated by the interrupt stack and current mode bits in the PSL. The interrupt stack is not context switched.

interrupt stack pointer
The stack pointer for the interrupt stack. Unlike the stack pointers for process context stacks, which are stored in the hardware PCB, the interrupt stack pointer is stored in an internal register.

interrupt vector
See vector.

I/O driver
See driver.

I/O function
An I/O operation that is interpreted by the operating system and typically results in one or more physical I/O operations.

I/O function code
A 6-bit value specified in a queue I/O request system service that describes the particular I/O operation to be performed (e.g., read, write, rewind).

I/O function modifier
A 10-bit value specified in a queue I/O request system service that modifies an I/O function code (e.g., read terminal input no echo).

I/O lockdown
The state of a page such that it cannot be paged or swapped out of memory until any I/O in progress to that page is completed.

I/O rundown
An operating system function in which the system cleans up any I/O in progress when an image exits.

I/O space
The region of physical address space that contains the configuration registers, and device control/status and data registers.

I/O status block
A data structure associated with the queue I/O request system service. This service optionally returns a status code, number of bytes transferred, and device- and function-dependent information in an I/O status block. It is not returned from the service call, but filled in when the I/O request completes.

job

1. A job is the accounting unit equivalent to a process and the collection of all the subprocesses, if any, that it and its subprocesses create. Jobs are classified as batch and interactive. For example, the job controller creates an interactive job to handle a user's requests when the user logs onto the system and it creates a batch job when the symbiont manager passes a command input file to it.

2. A print job.

job controller
The system process that establishes a job's process context, starts a process running the LOGIN image for the job, maintains the accounting record for the job, manages symbionts, and terminates a process and its subprocesses.

job queue
A list of files that a process has supplied for processing by a specific device, for example, a line printer.

kernel mode
The most privileged processor access mode (mode 0). The operating system's most privileged services, such as I/O drivers and the pager, run in kernel mode.

librarian
A program that allows the user to create, update, modify, list, and maintain object library, image library, and assembler macro library files.

library file
A direct access file containing one or more modules of the same module type.

limit
The size or number of given items requiring system resources (such as mailboxes, locked pages, I/O requests, open files, etc.) that a job is allowed to have at any one time during execution, as specified by the system manager in the user authorization file. See also quota.

line number
A number used to identify a line of text in a file processed by a text editor.

linker
A program that reads one or more object files created by language processors and produces an executable image file, a sharable image file, or a system image file.

linking
The resolution of external references between object modules used to create an image, the acquisition of referenced library routines, service entry points, and data for the image, and the assignment of virtual addresses to components of an image.

literal mode
In literal mode addressing, the instruction operand is a constant whose value is expressed in a 6-bit field of the instruction. If the operand data type is byte, word, longword, or quadword, the operand is zero-extended and can express values in the range 0 through 63 (decimal). If the operand data type is floating or double floating, the 6-bit field is composed of two 3-bit fields, one for the exponent and the other for the fraction. The operand is extended to floating or double floating format.

locality
See program locality.

local symbol
A symbol meaningful only to the module that defines it. Symbols not identified to a language processor as global symbols are considered to be local symbols. A language processor resolves (matches references with definitions) local symbols. They are not known to the linker and cannot be made available to another object module. They can, however, be passed through the linker to the symbolic debuffer. Contrasts with global symbol.

locate mode
A record access technique in which a program reads records in a record management service (RMS) block buffer working storage area to reduce overhead. See also move mode.

locking a page in memory
Making a page in an image ineligible for either paging or swapping. A page stays locked in memory until it is specifically unlocked.

locking a page in the working set
Making a page in an image ineligible for paging out of the working set for the image. The page can be swapped when the process is swapped. A page stays locked in a working set until it is specifically unlocked.

logical block number
A number used to identify a byte on a mass storage device. The number is a volume-relative address rather than a physical (device-oriented) address or a virtual (file-relative) address. The blocks that comprise the volume are labeled sequentially with logical block number 0.

logical I/O function
A set of I/O operations (e.g., read and write logical block) that allow restricted direct access to device level I/O operations using logical block addresses.

logical name
A user-specified name for any portion or all of a file specification. For example, the logical name INPUT can be assigned to a terminal device from which a program reads data entered by a user. Logical name assignments are maintained in logical name tables for each process, each group, and the system. A logical name can be created and assigned a value permanently or dynamically.

logical name table
A table that contains a set of logical names and their equivalent names for a particular process, a particular group, or the system.

longword
Four contiguous bytes (32 bits) starting on an addressable byte boundary. Bits are numbered from right to left with 0 through 31. The address of the longword is the address of the byte containing bit 0. When interpreted arithmetically, a longword is a two's complement integer with significance increas-

ing from bit 0 to bit 30. When interpreted as a signed integer, bit 31 is the sign bit. The value of the signed integer is in the range -2,147,483,648 to 2,147,483,647. When interpreted as an unsigned integer, significance increases from bit 0 to bit 31. The value of the unsigned integer is in the range 0 through 4,294,967,295.

macro
A statement that requests a language processor to generate a predefined set of instructions.

mailbox
A software data structure that is treated as a record-oriented device for general interprocess communication. Communication using a mailbox is similar to other forms of device-independent I/O. Senders perform a write to a mailbox, the receiver performs a read from that mailbox. Some system-wide mailboxes are defined: the error logger and OPCOM read from system-wide mailboxes.

main memory
See physical memory.

mapping window
A subset of the retrieval information for a file that is used to translate virtual block numbers to logical block numbers.

mass storage device
A device capable of reading and writing data on mass storage such as a disk pack or a magnetic tape reel.

member number
The second number in a user identification code that uniquely identifies that code.

memory management
The system functions that include the hardware's page mapping and protection and the operating system's image activator and pager.

memory mapping enable (MME)
A bit in a processor register that governs address translation.

modify access type
The specified operand of an instruction or procedure is read, and is potentially modified and written, during that instruction's or procedure's execution.

module

1. A portion of a program or program library, as in a *source module, object module,* or *image module.*

2. A board, usually made of plastic covered with an electrical conductor, on which logic devices (such as transistors, resistors, and memory chips) are mounted, and circuits connecting these devices are etched, as in a *logic module.*

monitor console routine (MCR)
The command interpreter in an RSX-11 system.

mount a volume

1. To logically associate a volume with the physical unit on which it is loaded (an activity accomplished by system software at the request of an operator).

2. To load or place a magnetic tape or disk pack on a drive and place the drive on-line (an activity accomplished by a system operator).

move mode
A record I/E access technique in which a program accesses records in its own working storage area. See also locate mode.

mutex
A flag that is used to control exclusive access to a region of code that can share a data structure or other resource. The mutex (mutual exclusion) flag ensures that only one process at a time has access to the region of code.

name block (NAM)
An RMS user data structure that contains supplementary information used in parsing file specifications.

native image
An image whose instructions are executed in native mode.

native mode
The processor's primary execution mode in which the programmed instructions are interpreted as byte-aligned, variable-length instructions that operate on byte, word, longword, and quadword integer, floating and double floating, character string, packed decimal, and variable-length bit field data. The instruction execution mode other than compatibility mode.

network
A collection of interconnected individual computer systems.

nibble
The low-order or high-order four bits of a byte.

node
An individual computer system in a network.

null process
A small system process that is the lowest priority process in the system and takes one entire priority class. One function of the null process is to accumulate idle processor time.

numeric string
A contiguous sequence of bytes representing up to 31 decimal digits (one per byte) and possibly a sign. The numeric string is specified by its lowest addressed location, its length, and its sign representation.

object module
The binary output of a language processor such as the assembler or a compiler, which is used as input to the linker.

object time system (OTS)
See run time procedure library.

offset
A fixed displacement from the beginning of a data structure. System offsets for items within a data structure normally have an associated symbolic name used instead of the numeric displacement. Where symbols are defined, programmers always reference the symbolic names for items in a data structure instead of using the numeric displacement.

op code
The pattern of bits within an instruction that specifies the operation to be performed.

operand specifier
The pattern of bits in an instruction that indicates the addressing mode, a register and/or displacement, which, taken together, identify an instruction operand.

operand specifier type
The access type and data type of an instruction's operand(s). For example, the test instructions are of read access type, since they only read the value of the operand. The operand can be of byte, word, or longword data type, depending on whether the op code is for the TSTB (test byte), TSTW (test word), or TSTL (test longword) instruction.

operator communication manager (OPCOM)
A system process that is always active. OPCOM receives input from a process that wants to inform an operator of a particular status or condition, passes a message to the operator, and tracks the message.

operator's console
Any terminal identified as a terminal attended by a system operator.

owner
In the context "system, owner, group, world," an owner is the particular member (or a group) to which a file, global section, mailbox, or event flag cluster belongs.

owner process
The process (with the exception of the job controller) or subprocess that created a subprocess.

packed decimal
A method of representing a decimal number by storing a pair of decimal digits in one byte, taking advantage of the fact that only four bits are required to represent the numbers zero through nine.

packed decimal string
A contiguous sequence of up to 16 bytes interpreted as a string of nibbles. Each nibble represents a digit except the low-order nibble of the highest addressed byte, which represents the sign. The packed decimal string is specified by its lowest addressed location and the number of digits.

page

1. A set of 512 contiguous byte locations used as the unit of memory mapping and protection.

2. The data between the beginning of file and a page marker, between two markers, or between a marker and the end of a file.

page fault
An exception generated by a reference to a page which is not mapped into a working set.

page fault cluster size
The number of pages read in on a page fault.

page frame number (PFN)
The address of the first byte of a page in physical memory. The high-order 21 bits of the physical address of the base of a page.

page marker
A character or characters (generally a form feed) that separates pages in a file that is processed by a text editor.

pager
A set of kernel mode procedures that executes as the result of a page fault. The pager makes the page for which the fault occurred available in physical memory so that the image can continue execution. The pager and the image activator provide the operating system's memory management functions.

page table entry (PTE)
The data structure that identifies the location and status of a page of virtual address space. When a virtual page is in memory, the PTE contains the page frame number needed to map the virtual page to a physical page. When it is not in memory, the PTE contains the information needed to locate the page on secondary storage (disk).

paging
The action of bringing pages of an executing process into physical memory when referenced. When the process executes, all of its pages are said to reside in virtual memory. Only the actively used pages, however, need to reside in physical memory. The remaining pages can reside on disk until they are needed in physical memory. In this system, a process is paged only when it references more pages than it is allowed to have in its working set. When the process refers to a page not in its working set, a page fault occurs. This causes the operating system's pager to read in the referenced page if it is on disk (and, optionally, other related pages depending on a cluster factor), replacing the least recently faulted pages as needed. A process pages only against itself.

parameter
See command parameter.

per-process address space
See process address space.

physical address
The address used by hardware to identify a location in physical memory on directly addressable secondary storage devices such as a disk. A physical memory address consists of a page frame number and the number of a byte within the page. A physical disk block address consists of a cylinder or track and sector number.

physical address space
The set of all possible 30-bit physical addresses that can be used to refer to locations in memory (memory space) or device registers (I/O space).

physical block
A block on a mass storage device referred to by its physical (device-oriented) address rather than a logical (volume-relative) or virtual (file-relative) address.

physical I/O functions
A set of I/O functions that allow access to all device level I/O operations except maintenance mode.

physical memory
The memory modules connected to the SBI that are used to store:

1. instructions that the processor can directly fetch and execute, and
2. any other data that a processor is instructed to manipulate.

Also called main memory.

position dependent eode
Code that can execute properly only in the locations in virtual address space that are assigned to it by the linker.

position independent code
Code that can execute properly without modification wherever it is located in virtual address space, even if its location is changed after it has been linked. Generally, this code uses addressing modes that form an effective address relative to the program counter (PC).

primary vector
A location that contains the starting address of a condition handler to be executed when an exception condition occurs. If a primary vector is declared, that condition handler is the first handler to be executed.

private section
An image section of a process that is not sharable among processes. See also global section.

privilege
See process privilege, user privilege, and image privilege.

privileged instructions
In general, any instructions intended for use by the operating system or privileged system programs. In particular, instructions that the processor will not execute unless the current access mode is kernel mode (e.g., HALT, SVPCTX, LDPCTX, MTPR, and MFPR).

procedure
1. A routine entered via a call instruction.
2. See command procedure.

process
The basic entity scheduled by the system software that provides the context in which an image executes. A process consists of an address space and both hardware and software context.

process address space
See process space.

process context
The hardware and software contexts of a process.

process control block (PCB)
A data structure used to contain process context. The hardware PCB contains the hardware context. The software PCB contains the software context, which includes a pointer to the hardware PCB.

process header
A data structure that contains the hardware process control block (PCB), accounting and quota information, process section table, working set list, and the page tables defining the virtual layout of the process.

process header slots
That portion of the system address space in which the system stores the process headers for the processes in the balance set. The number of process header slots in the system determines the number of processes that can be in the balance set at any one time.

process identification (PID)
The operating system's unique 32-bit binary value assigned to a process.

process I/O segment
That portion of a process control region that contains the process permanent record management service (RMS) internal file access block for each open file, and the I/O buffers, including the command interpreter's command buffer and command descriptors.

process name
A 1- to 15-character ASCII string that can be used to identify processes executing under the same group number.

processor register
A part of the processor used by the operating system software to control the execution states of the computer system. They include the system base and length registers, the program and control region base and length registers, the system control block base register, the software interrupt request register, and many more.

processor status longword (PSL)
A system-programmed processor register consisting of a word of privileged processor status and the PSW. The privileged processor status information includes: the current interrupt priority level (IPL), the previous access mode, the current access mode, the interrupt stack bit, the trace trap pending bit, and the compatibility mode bit.

processor status word (PSW)
The low-order word of the processor status longword. Processor status information includes: the condition codes (carry, overflow, zero, negative), the arithmetic trap enable bits (integer overflow, decimal overflow, floating underflow), and the trace enable bit.

process page tables
The page tables used to describe process virtual memory.

process priority
The priority assigned to a process for scheduling purposes. The operating system recognizes 32 levels of process priority, where 0 is low and 31 high. Levels 16 through 31 are used for time-critical processes. The system does not modify the priority of a time-critical process (although the system manager or process itself may). Levels 0 through 15 are used for normal processes. The system may temporarily increase the priority of a normal process based on the activity of the process.

process privileges
The privileges granted to a process by the system, which are a combination of user privileges and image privileges. They include, for example, the privilege to: affect other processes associated with the same group as the user's group, affect any process in the system regardless of user identification code (UIC), set process swap mode, create permanent event flag clusters, create another process, create a mailbox, and perform direct I/O to a file-structured device.

process section
See private section.

process space
The lowest-addressed half of virtual address space, where per process instructions and data reside. Process space is divided into a program region and a control region.

program counter (PC)
General register 15 (R15). At the beginning of an instruction's execution, the PC normally contains the address of a location in memory from which the processor will fetch the next instruction it will execute.

program locality
A characteristic of a program that indicates how close or far apart the references to locations in virtual memory are over time. A program with a high degree of locality does not refer to many widely scattered virtual addresses in a short period of time.

programmer number
See member number.

program region
The lowest-addressed half of process address virtual memory space (P0 space). The program region contains the image currently being executed by the process and other user code called by the image.

program region base register (P0BR)
The processor register, or its equivalent in a hardware control block, that contains the base virtual address of the page table entry for virtual page number 0 in a process program region.

program region length register (P0LR)
The processor register, or its equivalent in a hardware process control block, that contains the number of entries in the page table for a process program region.

program section (psect)
A portion of a program with a given protection and set of storage management attributes. Program sections that have the same attributes are gathered together by the linker to form an image section.

project number
See group number or account number.

pure code
See reentrant code.

quadword
Eight contiguous bytes (64 bits) starting on an addressable byte boundary. Bits are numbered from right to left, 0 to 63. A quadword is identified by the address of the byte containing the low-order bit (bit 0). When interpreted arithmetically, a quadword is a two's complement integer with significance increasing from bit 0 to bit 62. Bit 63 is used as the sign bit. The value of the integer is in the range -2^{63} to $2^{63}-1$.

qualifier
A portion of a command string that modifies a command verb or command parameter by selecting one of several options. A qualifier, if present, follows the command verb or parameter to which it applies and is in the format: "/qualifier:option." For example, in the command string "PRINT filename/COPIES:3," the COPIES qualifier indicates that the user wants three copies of a given file printed.

queue

1. noun. A circular, doubly linked list. See system queues. verb. To make an entry in a list or table, perhaps using the INSQUE instruction.

2. See job queue.

queue priority
The priority assigned to a job placed in a spooler queue or a batch queue.

quota
The total amount of a system resource, such as CPU time, that a job is allowed to use in an accounting period, as specified by the system manager in the user authorization file. See also limit.

random access by record's file address
The retrieval of a record by its unique address, which is provided to the program by record management services (RMS). The method of access can be used to randomly access a sequentially organized file containing variable length records.

random access by relative record number
The retrieval or storage of a record by specifying its position relative to the beginning of the file.

read access type
An instruction or procedure operand attribute indicating that the specified operand is only read during instruction or procedure execution.

record access block (RAB)
A record management services (RMS) user data structure that represents a request for a record access stream. A RAB relates to operations on the records within a file, such as UPDATE, DELETE, or GET.

record access mode
The method used in record management services (RMS) for retrieving and storing a record in a file. One of three methods: sequential, random, and record's file address.

record management services (RMS)
A set of operating system procedures that are called by programs to process files and records within files. RMS allows programs to issue READ and WRITE requests at the record level (record I/O), as well as read and write blocks (block I/O). RMS is an integral part of the system software. RMS procedures run in executive mode.

record-oriented device
A device such as a terminal, line printer, or card reader, on which the largest unit of data a program can access in one I/O operation is the device's physical record.

record's file address
The unique address of a record in a file that allows records to be accessed randomly regardless of file organization.

record slot
A fixed length area in a relatively organized file that is used to contain one record.

reentrant code
Code that is never modified during execution. It is possible to let many users share the same copy of a procedure or program written as reentrant code.

register
A storage location in hardware logic other than main memory. See also general register, processor register, and device register.

register deferred indexed mode
An indexed addressing mode in which the base operand specifier uses register deferred mode addressing.

register deferred mode
In register deferred mode addressing, the contents of the specified register are used as the address of the actual instruction operand.

register mode
In register mode addressing, the contents of the specified register are used as the actual instruction operand.

relative file organization
A file organization in which the file contains fixed length record cells. Each cell is assigned a consecutive number that represents its position relative to the beginning of a file. Records within each cell can be as big as or smaller than the cell. Relative file organization permits sequential record access, random record access by record number, and random record access by the record's file address.

resource
A physical part of the computer system such as a device or memory, or an interlocked data structure such as a mutex. Quotas and limits control the use of physical resources.

resource wait mode
An execution state in which a process indicates that it will wait until a system resource becomes available when it issues a service request requiring a resource. If a process wants notification when a resource is not available, it can disable resource wait mode during program execution.

return status code
See status code.

run time procedure library
The collection of procedures available to native mode images at run time. These library procedures (such as trigonometric functions, etc.) are common to all native mode images, regardless of the language processor used to compile or assemble the program.

scatter/gather
The ability to transfer in one I/O operation data from discontiguous pages in memory to contiguous blocks on disk, or data from contiguous blocks on disk to discontiguous pages in memory.

secondary storage
Random access mass storage.

secondary vector
A location that identifies the starting address of a condition handler to be executed when a condition occurs and the primary vector contains zero or the handler to which the primary vector points chooses not to handle the condition.

section
A portion of process virtual memory that has common memory management attributes (protection, access, cluster factor, etc.). It is created from an image section, a disk file, or as the result of a create virtual address space system service. See global section, private section, image section, and program section.

sequential access mode
The retrieval or storage of records in which a program successively reads or writes records one after the other in the order in which they appear, starting and ending at any arbitrary point in the file.

sequential file organization
A file organization in which records appear in the order in which they were originally written. The records can be of fixed length or variable length. Although one does not speak of record slots with sequentially organized files, for purposes of comparison with relatively organized files one can say that the record itself is the same as its record slot, and its record number is the same as its relative slot number. Sequential file organization permits sequential record access and random access by a record's file address. Sequential file organization with fixed length records also permits random access by relative record number.

sharable image
An image that has all of its internal references resolved, but which must be linked with an object module(s) to produce an executable image. A sharable image cannot be executed. A sharable image file can be used to contain a library of routines. A sharable image can be used to create a global section by the system manager.

shell process
A predefined process that the job initiator copies to create the minimum context necessary to establish a process.

signal
1. An electrical impulse conveying information.
2. The software mechanism used to indicate that an exception condition was detected.

slave terminal
A terminal from which it is not possible to issue commands to the command interpreter. A terminal assigned to application software.

small process
A system process that has no control region in its virtual address space and has an abbreviated context. Examples are the working set swapper and the null process. A small process is scheduled in the same manner as user processes, but must remain resident during its execution.

software context
The context maintained by the operating system that describes a process. See software process control block (PCB).

software interrupt
An interrupt generated on interrupt priority level 1 through 15, which can be requested only by software.

software process control block (PCB)
The data structure used to contain a process's software context. The operating system defines a software PCB for every process when the process is created. The software PCB includes the following kinds of information about the process: current state; storage address if it is swapped out of memory; unique identification of the process, and address of the process header (which contains the hardware PCB). The software PCB resides in system region virtual address space. It is not swapped with a process.

software priority
See process priority and queue priority.

spooling
Output spooling: The method by which output to a low-speed peripheral device (such as a line printer) is placed into queues maintained on a high-speed device (such as disk) to await transmission to the low-speed device.

Input spooling: The method by which input from a low-speed peripheral (such as the card reader) is placed into queues maintained on a high-speed device (such as disk) to await transmission to a job processing that input.

spool queue
The list of files supplied by processes that are to be processed by a symbiont. For example, a line printer queue is a list of files to be printed on the line printer.

stack
An area of memory set aside for temporary data storage, or for procedure and interrupt service linkages. A stack uses the last-in, first-out concept. As items are added to ("pushed on") the stack, the stack pointer decrements. As items are retrieved from ("popped off") the stack, the stack pointer increments.

stack frame
A standard data structure built on the stack during a procedure call, starting from the location addressed by the frame pointer (FP) to lower addresses, and popped off during a return from procedure. Also called call frame.

stack pointer (SP)
General register 14 (R14). SP contains the address of the top (lowest address) of the processor-defined stack. Reference to SP will access one of the five possible stack pointers, kernel, executive, supervisor, user, or interrupt, depending on the value in the current mode and interrupt stack bits in the processor status longword (PSL).

state queue
A list of processes in a particular processing state. The scheduler uses state queues to keep track of processes' eligibility to execute. They include: processes waiting for a common event flag, suspended processes, and executable processes.

status code
A longword value that indicates the success or failure of a specific function. For example, system services always return a status code in R0 upon completion.

store through
See write through.

strong definition
Definition of a global symbol that is explicitly available for reference by modules linked with the module in which the definition occurs. The linker always lists a global symbol with a strong definition in the symbol portion of the map. The librarian always includes a global symbol with a strong definition in the global symbol table of a library.

strong reference
A reference to a global symbol in an object module that requests the linker to report an error if it does not find a definition for the symbol during linking. If a library contains the definition, the linker incorporates the library module defining the global symbol into the image containing the strong reference.

subprocess
A subsidiary process created by another process. The process that creates a subprocess is its owner. A subprocess receives resource quotas and limits from its owner. When an owner process is removed from the system, all its subprocesses (and their subprocesses) are also removed.

supervisor mode
The third most privileged processor access mode (mode 2). The operating system's command interpreter runs in supervisor mode.

suspension
A state in which a process is inactive, but known to the system. A suspended process becomes active again only when another process requests the operating system to resume it. Contrasts with hibernation.

swap mode
A process execution state that determines the eligibility of a process to be swapped out of the balance set. If process swap mode is disabled, the process working set is locked in the balance set.

swapping
The method for sharing memory resources among several processes by writing an entire working set to secondary storage (swap out) and reading another working set into memory (swap in). For example, a process's working set can be written to secondary storage while the process is waiting for I/O completion on a slow device. It is brought back into the balance set when I/O completes. Contrasts with paging.

switch
See (command) qualifier.

symbiont
A full process that transfers record-oriented data to or from a mass storage device. For example, an input symbiont transfers data from card readers to disks. An output symbiont transfers data from disks to line printers.

symbiont manager
The function (in the system process called the job controller) that maintains spool queues, and dynamically creates symbiont processes to perform the necessary I/O operations.

symbol
See local symbol, global symbol, and universal global symbol.

synchronous backplane interconnect (SBI)
The part of the hardware that interconnects the CPU, memory controllers, Massbus adapters, the Unibus adapter.

synchronous record operation
A mode of record processing in which a user program issues a record read or write request and then waits until that request is fulfilled before continuing to execute.

system
In the context "system, owner, group, world," the system refers to the group numbers that are used by the operating system and its controlling users, the system operators and system manager.

system address space
See system space and system region.

system base register (SBR)
A processor register containing the physical address of the base of the system page table (SPT).

system buffered I/O
An I/O operation, such as terminal or mailbox I/O, in which an intermediate buffer from the system pool is used instead of a processor-specified buffer. Contrasts with direct I/O.

system control block (SCB)
The data structure in system space that contains all the interrupt and exception vectors known to the system.

system control block base register (SCBB)
A processor register containing the base address of the system control block.

system device
The random access mass storage device unit on which the volume containing the operating system software resides.

system dynamic memory
Memory reserved for the operating system to allocate as needed for temporary storage. For example, when an image issues an I/O request, system dynamic memory is used to contain the I/O request packet. Each process has a limit on the amount of system dynamic memory that can be allocated for its use at one time.

system identification register
A processor register that contains the processor type and serial number.

system image
The image that is read into memory from secondary storage when the system is started up.

system length register (SLR)
A processor register containing the length of the system page table (SPT) in longwords, that is, the number of page table entries (PTEs) in the system region page table.

system page table (SPT)
The data structure that maps the system region virtual addresses, including the addresses used to refer to the process page tables. The SPT contains one page table entry (PTE) for each page of system region virtual memory. The physical base address of the SPT is contained in a register called the system base register (SBR).

system process
A process that provides system-level functions. Any process that is part of the operating system. See also small process, fork process.

system programmer
A person who designs and/or writes operating systems, or who designs and writes procedures or programs that provide general purpose services for an application system.

system queue
A queue used and maintained by operating system procedures. See also state queues.

system region
The third quarter of virtual address space. The lowest-addressed half of system space. Virtual addresses in the system region are sharable between processes. Some of the data structures mapped by the system region virtual addresses are: system entry vectors, the system control block (SCB), the system page table (SPT), and process page tables.

system services
Procedures provided by the operating system that can be called by user processes.

system space
The highest-addressed half of virtual address space. See also system region.

system virtual address
A virtual address identifying a location mapped by an address in system space.

system virtual space
See system space.

task
An RSX-11/IAS term for a process and image bound together.

terminal
The general name for those peripheral devices that have keyboards and video screens or printers. Under program control, a terminal enables people to type commands and data on the keyboard and receive messages on the video screen or printer. Examples of terminals are the LA36 DECwriter hardcopy terminal and VT52 video display terminal.

time-critical process
A process assigned to a software priority level between 16 and 31, inclusive. The scheduling priority assigned to a time-critical process is never modified by the scheduler, although it can be modified by the system manager or process itself.

timer
A system fork process that maintains the time of day and the date. It also scans for device timeouts and performs time-dependent scheduling upon request.

track
A collection of blocks at a single radius on one recording surface of a disk.

transfer address
The address of the location containing a program entry point (the first instruction to execute).

translation buffer
An internal processor cache containing translations for recently used virtual addresses.

trap
An exception condition that occurs at the end of the instruction that caused the exception. The program counter (PC) saved on the stack is the address of the next instruction that would normally have been executed. All software can enable and disable some of the trap condition with a single instruction.

trap enables
Three bits in the processor status word that control the processor's action on certain arithmetic exceptions.

two's complement
A binary representation for integers in which a negative number is one greater than the bit complement of the positive number.

two-way associative cache
A cache organization that has two groups of directly mapped blocks. Each group contains several blocks for each index position in the cache. A block of data from main memory can go into any group at its proper index position. A two-way associative cache is a compromise between the extremes of fully associative and direct mapping cache organizations that takes advantage of the features of both.

type ahead
A terminal handling technique in which the user can enter commands and data while the software is processing a previously entered command. The commands typed ahead are not echoed on the terminal until the command processor is ready to process them. They are held in a type ahead buffer.

unit record device
A device such as a card reader or line printer.

universal global symbol
A global symbol in a sharable image that can be used by modules linked with that sharable image. Universal global symbols are typically a subset of all the global symbols in a sharable image. When creating a sharable image, the linker ensures that universal global symbols remain available for reference after symbols have been resolved.

unwind the call stack
To remove call frames from the stack by tracing back through nested procedure calls using the current contents of the FP register and the frame pointer (FP) register contents stored on the stack for each call frame.

urgent interrupt
An interrupt received on interrupt priority levels 24 through 31. These can be generated only by the processor for the interval clock, serious errors, and power-fail.

user authorization file
A file containing an entry for every user that the system manager authorizes to gain access to the system. Each entry identifies the user name, password, default account, user identification code (UIC), quotas, limits, and privileges assigned to individuals who use the system.

user environment test package (UETP)
A collection of routines that verify that the hardware and software systems are complete, properly installed, and ready to be used.

user file directory (UFD)
See directory.

user identification code (UIC)
The pair of numbers assigned to users and to files, global sections, common event flag clusters, and mailboxes that specify the type of access (read and/or write access, and in the case of files execute and/or delete access) available to the owners, group, world, and system. It consists of a group number and a member number separated by a comma.

user mode
The least privileged processor access mode (mode 3). User processes and the run time library procedures run in user mode.

user name
The name that a person types on a terminal to log on to the system.

user number
See member number.

user privileges
The privileges granted a user by the system manager. See process privileges.

utility
A program that provides a set of related general purpose functions, such as a program development utility (an editor, a linker, etc.), a file management utility (file copy or file format translation program), or operations management utility (disk backup/restore, diagnostic program, etc.).

value return registers
The general registers R0 and R1 used by convention to return function values. These registers are not preserved by any called procedures. They are available as temporary registers to any called procedure. All other registers (R2, R3, . . ., R11, AP, FP, SP, PC) are preserved across procedure calls.

variable-length bit field
A set of zero to 32 contiguous bits located arbitrarily with respect to byte boundaries. A variable bit field is specified by four attributes:

1. the address A of a byte,

2. the bit position P of the starting location of the bit field with respect to bit 0 of the byte at address A,

3. the size, in bits, of the bit field, and

4. whether the field is signed or unsigned.

variable-length record format
A file format in which records are not necessarily the same length.

variable with fixed-length control (VFC) record format
A file format in which records of variable length contain an additional fixed-length control area. The control area may be used to contain file line numbers and/or print format controls.

vector
1. An interrupt or exception vector is a storage location known to the system that contains the starting address of a procedure to be executed when a given interrupt or exception occurs. The system defines separate vectors for each interrupting device controller and for classes of exceptions. Each system vector is a longword.

2. For the purposes of exception handling, users can declare up to two software exception vectors (primary and secondary) for each of the four access modes. Each vector contains the address of a condition handler.

3. A one-dimensional array.

version number

1. The field following the file type in a file specification. It is separated from file type by a period (.) or semicolon (;) and consists of a number that generally identifies it as one version among all the files having the identical file specification but for version number.

2. The number used to identify the revision level of program.

virtual address
A 32-bit integer identifying a byte "location" in virtual address space. The memory management hardware translates a virtual address to a physical address. The term virtual address may also refer to the address used to identify a virtual block on a mass storage device.

virtual address space
The set of all possible virtual addresses that an image executing in the context of a process can use to identify the location of an instruction or data. The virtual address space seen by the programmer is a linear array of 4,294,967,296 (2^{32}) byte address.

virtual block number
A number used to identify a block on a mass storage device. The number is a file-relative address rather than a logical (volume-oriented) or physical (device-oriented) address. The first block in a file is always virtual block number 1.

virtual I/O functions
A set of I/O functions that must be interpreted by an ancillary control process.

virtual memory
The set of storage locations in physical memory and on disk that are referred to by virtual addresses. From the programmer's viewpoint, the secondary storage locations appear to be locations in physical memory. The size of virtual memory in any system depends on the amount of physical memory available and the amount of disk storage used for nonresident virtual memory.

virtual page number
The virtual address of a page of virtual memory.

volume
A mass storage medium such as a disk pack or reel of magnetic tape.

volume set
The file-structured collection of data residing on one or more mass storage media.

wait
To become inactive. A process enters a process wait state when the process suspends itself, hibernates, or declares that it needs to wait for an event, resource, mutex, etc.

wake
To activate a hibernating process. A hibernating process can be awakened by another process or by the timer process, if the hibernating process or another process scheduled a wake-up call.

weak definition
Definition of a global symbol that is not explicitly available for reference by modules linked with the module in which the definition occurs. The librarian does not include a global symbol with a weak definition in the global symbol table of a library. Weak definitions are often used when creating libraries to identify those global symbols that are needed only if the module containing them is otherwise linked with a program.

weak reference
A reference to a global symbol that requests the linker not to report an error or to search the default library's global symbol table to resolve the reference if the definition is not in the modules explicitly supplied to the linker. Weak references are often used when creating object modules to identify those global symbols that may not be needed at run time.

wild card
A symbol, such as an asterisk, that is used in place of a file name, file type, directory name, or version number in a file specification to indicate "all" for the given field.

window
See mapping window.

word
Two contiguous bytes (16 bits) starting on an addressable byte boundary. Bits are numbered from the right, 0 through 15. A word is identified by the address of the byte containing bit 0. When interpreted arithmetically, a word is a two's complement integer with significance increasing from bit 0 to bit 14. If interpreted as a signed integer, bit 15 is the sign bit. The value of the integer is in the range -32768 to 32767. When interpreted as an unsigned integer, significance increases from bit 0 through bit 15 and the value of the unsigned integer is in the range 0 through 65535.

working set
The set of pages in process space to which an executing process can refer without incurring a page fault. The working set must be resident in memory for the process to execute. The remaining pages of that process, if any, are either in memory and not in the process working set or they are on secondary storage.

working set swapper
A system process that brings process working sets into the balance set and removes them from the balance set.

world
In the context "system, owner, group, world," world refers to all users, including the system operators, the system manager, and users both in an owner's group and in any other group.

write access type
The specified operand of an instruction or procedure is only written during that instruction's or procedure's execution.

write allocate
A cache management technique in which cache is allocated on a write miss as well as on the usual read miss.

write back
A cache management technique in which data from a write operation to cache is copied into main memory only when the data in cache must be overwritten. This results in temporary inconsistencies between cache and main memory. Contrasts with write through.

write through
A cache management technique in which data from a write operation is copied in both cache and main memory. Cache and main memory data are always consistent. Contrasts with write back.

COMMONLY USED MNEMONICS

ACP	Ancillary Control Process
ANS	American National Standard
ASCII	American Standard Code for Information Interchange
AST	Asynchronous System Trap
ASTLVL	Asynchronous System Trap LeVeL
CCB	Channel Control Block
CM	Compatibility Mode bit in the hardware PSL
CRB	Channel Request Block
CRC	Cyclic Redundancy Check
DAP	Data Access Protocol
DDB	Device Data Block
DDCMP	DIGITAL Data Communications Message Protocol
DDT	Driver Data Table
DV	Decimal oVerflow trap enable bit in the PSW
ECB	Exit Control Block
ECC	Error Correction Code
ESP	Executive mode Stack Pointer
ESR	Exception Service Routine
F11ACP	Files-11 Ancillary Control Process
FAB	File Access Block
FCA	Fixed Control Area
FCB	File Control Block
FCS	File Control Services
FDT	Function Decision Table
FP	Frame Pointer
FPD	First Part (of an instruction) Done
FU	Floating Underflow trap enable bit in the PSW
GSD	Global Section Descriptor
GST	Global Symbol Table
IDB	Interrupt Dispatch Block
IPL	Interrupt Priority Level
IRP	I/O Request Packet
ISECT	Image SECTion
ISD	Image Section Descriptor
ISP	Interrupt Stack Pointer
IS	Interrupt Stack bit in PSL
ISR	Interrupt Service Routine
IV	Integer oVerflow trap enable bit in the PSW
JCS	Joint Control Store
KSP	Kernel mode Stack Pointer
MBA	Massbus Adapter
MBZ	Must Be Zero
MCR	Monitor Console Routine
MFD	Master File Directory
MFPR	Move From Process Register instruction
MME	Memory Mapping Enable
MTPR	Move To Process Register instruction
MUTEX	MUTual EXclusion semaphore
NSP	Network Services Protocol
OPCOM	OPeration COMmunication manager

P0BR	Program region Base Register
P0LR	Program region Length Register
P0PT	Program region Page Table
P1BR	control region Base Register
P1LR	control region Limit Register
P1PT	control region Page Table
PC	Program Counter
PCB	Process Control Block
PCBB	Process Control Block Base register
PFN	Page Frame Number
PID	Process Identification Number
PME	Performance Monitor Enable bit in PCB
PSECT	Program SECTion
PSL	Processor Status Longword
PSW	Processor Status Word
PTE	Page Table Entry
QIO	Queue Input/Output request system service
RAB	Record Access Block
RFA	Record's File Address
RMS	Record Management Services
RWED	Read, Write, Execute, Delete
SBI	Synchronous Backplane Interconnect
SBR	System Base Register
SCB	System Control Block
SCBB	System Control Block Base register
SLR	System Length Register
SP	Stack Pointer
SPT	System Page Table
SSP	Supervisor mode Stack Pointer
SVA	System Virtual Address
TP	Trace trap Pending bit in PSL
UBA	Unibus Adapter
UCB	Unit Control Block
UETP	User Environment Test Package
UFD	User File Directory
UIC	User Identification Code
USP	User mode Stack Pointer
VCB	Volume Control Block
VPN	Virtual Page Number
WCB	Window Control Block
WCS	Writable Control Store
WDCS	Writable Diagnostic Control Store

APPENDIX H
NATIVE MODE INSTRUCTION SET

The instruction set that the processor executes is selected under operating system control to either native mode or compatibility mode. The native mode instruction set is based on over 200 different op codes. The op codes can be grouped into classes based on their function and use. Instructions used to manipulate the general data types include:

- integer and floating-point instructions
- packed decimal instructions
- character string instructions
- bit field instructions

Instructions that are used to manipulate special kinds of data include:

- queue manipulation instructions
- address manipulation instructions
- user-programmed general register control instructions

Instructions that provide basic program flow control and enable you to call procedures are:

- branch, jump, and case instructions
- subroutine call instructions
- procedure call instructions

Table H-1 lists the basic instruction operations in order by these classifications. Instructions that enable operating system procedures to provide user mode processes with services requiring privilege are listed in the table, but discussed in the system programming environment section. Instructions that are singular in the functions they provide are listed last.

Table H-1 Native Mode Instruction Set Summary

Integer and Floating-Point Logical Instructions

MOV$	Move (B,W,L,F,D,Q)*
MNEG$	Move Negated (B,W,L,F,D)
MCOM$	Move Complemented (B,W,L)
MOVZ$	Move Zero-Extended (BW,BL,WL)
CLR$	Clear (B,W,L=Q,F=D)
CVT$	Convert (B,W,L,F,D) (B,W,L,F,D)
CVTR$	Convert Rounded (F,D) to Longword
CMP$	Compare (B,W,L,F,D)
TST$	Test (B,W,L,F,D)
BIS$2	Bit Set (B,W,L) 2-Operand
BIS$3	Bit Set (B,W,L) 3-Operand
BIC$2	Bit Clear (B,W,L) 2-Operand
BIC$3	Bit Clear (B,W,L) 3-Operand
BIT$	Bit Test (B,W,L)
XOR$2	Exclusive OR (B,W,L) 2-Operand
XOR$3	Exclusive OR (B,W,L) 3-Operand
ROTL	Rotate Longword

Integer and Floating-Point Arithmetic Instructions

INC$	Increment (B,W,L)
DEC$	Decrement (B,W,L)
ASH$	Arithmetic Shift (L,Q)
ADD$2	Add (B,W,L,F,D) 2-Operand
ADD$3	Add (B,W,L,F,D) 3-Operand
ADWC	Add with Carry
ADAWI	Add Aligned Word Interlocked
SUB$2	Subtract (B,W,L,F,D) 2-Operand
SUB$3	Subtract (B,W,L,F,D) 3-Operand
SBWC	Subtract with Carry
MUL$2	Multiply (B,W,L,F,D) 2-Operand
MUL$3	Multiply (B,W,L,F,D) 3-Operand
EMUL	Extended Multiply
DIV$2	Divide (B,W,L,F,D) 2-Operand
DIV$3	Divide (B,W,L,F,D) 3-Operand
EDIV	Extended Divide
EMOD$	Extended Modulus (F,D)
POLY$	Polynomial Evaluation (F,D)

Packed Decimal Instructions

MOVP	Move Packed
CMPP3	Compare Packed 3-Operand
CMPP4	Compare Packed 4-Operand
ASHP	Arithmetic Shift Packed and Round
ADDP4	Add Packed 4-Operand
ADDP6	Add Packed 6-Operand
SUBP4	Subtract Packed 4-Operand
SUBP6	Subtract Packed 6-Operand
MULP	Multiply Packed
DIVP	Divide Packed
CVTLP	Convert Long to Packed
CVTPL	Convert Packed to Long
CVTPT	Convert Packed to Trailing
CVTTP	Convert Trailing to Packed
CVTPS	Convert Packed to Separate
CVTSP	Convert Separate to Packed
EDITPC	Edit Packed to Character String

Table H-1 Native Mode Instruction Set Summary (Cont)

Character String Instructions

MOVC3	Move Character 3-Operand
MOVC5	Move Character 5-Operand
MOVTC	Move Translated Characters
MOVTUC	Move Translated Unit Character
CMPC3	Compare Characters 3-Operand
CMPC5	Compare Characters 5-Operand
LOCC	Locate Character
SKPC	Skip Character
SCANC	Scan Characters
SPANC	Span Characters
MATCHC	Match Characters

Variable-Length Bit Field Instructions

EXTV	Extract Field
EXTZV	Extract Zero-Extended Field
INSV	Insert Field
CMPV	Compare Field
CMPZV	Compare Zero-Extended Field
FFS	Find First Set
FFC	Find First Clear

Index Instruction

INDEX	Compute Index

Queue Instructions

INSQUE	Insert Entry in Queue
REMQUE	Remove Entry from Queue

Address Manipulation Instructions

MOVA$	Move Address (B,W,L=F,Q=D)
PUSHA$	Push Address (B,W,L=F,Q=D) on Stack

General Register Manipulation Instructions

PUSHL	Push Longword on Stack
PUSHR	Push Registers on Stack
POPR	Pop Registers from Stack
MOVPSL	Move from Processor Status Longword
BISPSW	Bit Set Processor Status Word
BICPSW	Bit Clear Processor Status Word

Unconditional Branch and Jump Instructions

BR	Branch with (Byte, Word) Displacement
JMP	Jump

Table H-1 Native Mode Instruction Set Summary (Cont)

Branch on Condition Code

BLSS	Less Than
BLSSU	Less than Unsigned
BLEQ	Less than or Equal
BLEQU	Less than or Equal Unsigned
BEQL	Equal
(BEQLU)	(Equal Unsigned)
BNEQ	Not Equal
(BNEQU)	(Not Equal Unsigned)
BGTR	Greater than
BGTRU	Greater than Unsigned
BGEQ	Greater than or Equal
BGEQU	Greater than or Equal Unsigned
(BCC)	(Carry Cleared)
BVS	Overflow Set
BVC	Overflow Clear

Branch on Bit

BLB$	Branch on Low Bit (Set, Clear)
BB$	Branch on Bit (Set, Clear)
BBS$	Branch on Bit Set and (Set, Clear) bit
BBC$	Branch on Bit Clear and (Set, Clear) bit
BBSSI	Branch on Bit Set and Set bit interlocked
BBCCI	Branch on Bit Clear and Clear bit interlocked

Loop and Case Branch

ACB$	Add, Compare and Branch (B,W,L,F,D)
AOBLEQ	Add One and Branch Less Than or Equal
AOBLSS	Add One and Branch Less Than
SOBGEQ	Subtract One and Branch Greater Than or Equal
SOBGTR	Subtract One and Branch Greater Than
CASE$	Case on (B,W,L)

Subroutine Call and Return Instructions

BSB$	Branch to Subroutine with (B,W) Displacement
JSB	Jump to Subroutine
RSB	Return from Subroutine

Procedure Call and Return Instructions

CALLG	Call Procedure with General Argument List
CALLS	Call Procedure with Stack Argument List
RET	Return from Procedure

Table H-1 Native Mode Instruction Set Summary (Cont)

Protected Procedure Call and Return Instructions

CHMS	Change Mode to (Kernel, Executive, Supervisor, User)
REI	Return from Exception or Interrupt
PROBER	Probe Read
PROBEW	Probe Write

Privileged Processor Register Control Instructions

SVPCTX	Save Process Context
LDPCTX	Load Process Context
MTPR	Move to Process Register
MFPR	Move from Processor Register

Special Function Instructions

CRC	Cyclic Redundancy Check
BPT	Breakpoint Fault
XFC	Extended Function Call
NOP	No Operation
HALT	Halt

APPENDIX I
COMPATIBILITY MODE INSTRUCTION SET (PDP-11)

The compatibility mode instruction set is the PDP-11 instruction set. Programs previously constructed with this instruction set can be run directly on the VAX-11. New programs can also be developed on the VAX-11 · using this instruction. Programs using the native mode instruction set can be run concurrent with programs using the compatibility mode. Table I-1 lists the basic instruction operations.

Table I-1 Compatibility Mode Instruction Set Summary

Single Operand Instructions

CLR(B)	Clear Destination
DEC(B)	Decrement Destination
INC(B)	Increment Destination
NEG(B)	Negate Destination
TST(B)	Test Destination
COM(B)	Complement Destination
ASR(B)	Arithmetic Shift Right Destination
ASL(B)	Arithmetic Shift Left Destination
ASH	Shift Arithmetically
ASHC	Arithmetic Shift Combined
ADC(B)	Add Carry Destination
SBC(B)	Subtract Carry Destination
SXT	Shift Extend Destination
ROL(B)	Rotate Left Destination
ROR(B)	Rotate Right Destination
SWAB	Swap Bytes Destination

Double Operand Instructions

MOV(B)	Move Source to Destination
ADD	Add Source to Destination
MUL	Multiply
DIV	Divide
XOR	Exclusive OR
BIS(B)	Bit Set
BIT(B)	Bit Test
BIC(B)	Bit Clear

Table I-1 Compatibility Mode Instruction Set Summary (Cont)

Program Control Instructions

BR	Branch (Unconditional)
BEQ	Branch on Equal (Zero)
BNE	Branch Not Equal (Zero)
BMI	Branch on Minus
BPL	Branch on Plus
BCS	Branch on Carry Set
BCC	Branch on Carry Clear
BVS	Branch on Overflow Set
BVC	Branch on Overflow Clear
BLT	Branch on Less than (Zero)
BGE	Branch on Greater than or Equal (Zero)
BLE	Branch on Less than or Equal (Zero)
BGT	Branch on Greater than (Zero)
BHI	Branch on Higher
BLOS	Branch on Lower or Same
BHIS	Branch on Higher or Same
BLO	Branch on Lower
JSR	Jump to Subroutine
MARK	Mark
RTS	Return from Subroutine
SPL	Set Priority Level
JMP	Jump
SOB	Subtract One and Branch
EMT	Emulator Tapes
TRAP	Trap
BPT	Breakpoint Trap
IOT	I/O Trap
RTI	Return From Interrupt
RTT	Return From Trap

Miscellaneous

HALT	Halt
WAIT	Wait for Interrupt
RESET	Reset External Bus
MTPI	Move to Previous Instruction Space
MTPD	Move to Previous Data Space
MFPI	Move From Previous Instruction Space
MFPD	Move From Previous Data Space

Condition Code Operators

CLC	Clear C
CLV	Clear V
CLZ	Clear Z
CLN	Clear N
SEC	Set C
SEV	Set V
SEZ	Set Z
SEN	Set N
SCC	Set all CC
CCC	Clear all CCs

APPENDIX J
7-BIT ASCII CODE

OCTAL CODE	CHAR	OCTAL CODE	CHAR	OCTAL CODE	CHAR	OCTAL CODE	CHAR
000	NUL	040	SP	100	@	140	\
001	SOH	041	!	101	A	141	a
002	STX	042	”	102	B	142	b
003	ETX	043	#	103	C	143	c
004	EOT	044	$	104	D	144	d
005	ENQ	045	%	105	E	145	e
006	ACK	046	&	106	F	146	f
007	BEL	047	‘	107	G	147	g
010	BS	050	(	110	H	150	h
011	HT	051	)	111	I	151	i
012	LF	052	*	112	J	152	j
013	VT	053	+	113	K	153	k
014	FF	054	’	114	L	154	l
015	CR	055	-	115	M	155	m
016	SO	056	.	116	N	156	n
017	SI	057	/	117	O	157	o
020	DLE	060	0	120	P	160	p
021	DC1	061	1	121	Q	161	q
022	DC2	062	2	122	R	162	r
023	DC3	063	3	123	S	163	s
024	DC4	064	4	124	T	164	t
025	NAK	065	5	125	U	165	u
026	SYN	066	6	126	V	166	v
027	ETB	067	7	127	W	167	w
030	CAN	070	8	130	X	170	x
031	EM	071	9	131	Y	171	y
032	SUB	072	:	132	Z	172	z
033	ESC	073	;	133	[	173	{
034	FS	074	<	134	\	174	|
035	GS	075	=	135	] or ↑	175	}
036	RS	076	>	136	^	176	~
037	US	077	?	137	– or ←	177	DEL

APPENDIX K

DIAGNOSTIC PROGRAMS

A number of diagnostic programs are available for use in troubleshooting the VAX-11/785 and its peripheral equipment. These diagnostics are resident on floppy disks and on the operating system disk. The floppy disks are listed in Table K-1.

Table K-1 VAX-11/785 Diagnostic Program Floppy Disks

NUMBER	TITLE
RX1A	VAX 11785 Console
RX2A	VAX 11785 Micro Diag #1
RX3A	VAX 11785 Micro Diag #2
RX5	VAX FP785 Micro Diag
RX8A	VAX 11785 Micro Diag #3
RX10	VAX 11785 ETKAX Diag (CPU Cluster Exerciser)
RX22A	VAX 11785 MA780 Micro

The diagnostic programs available for use on VAX-11/785 systems and resident on the system disk are listed in Table K-2.

Table K-2 VAX-11/785 Diagnostic Programs

CODE	TITLE
ESCAA	VAX MBA (RH780) Diagnostic
ESCBA	VAX UBA (DW780) Diagnostic
ESCCA	MA780 Multi-Port Exerciser
ESCCB	MA780 Multi-Port Memory Exerciser
ESCGA	CI780 Repair Level Part I
ESCGB	CI780 Repair Level Part II
ESCGC	CI780 Repair Level Part III
ESCGD	CI780 Repair Level Part IV
ESDFA	DR780 High Speed SBI Interface Part A
ESDFB	DR780 High Speed SBI Interface Part B
ESDFC	DR780 High Speed SBI Interface Part C
ESSAA	VAX-11/780 Diagnostic Supervisor
ETKAB	VAX-11/785 (KA785) Microdiagnostic Monitor
ETKAC	VAX-11/785 (KA785) Hardcore Monitor
ETKAD	VAX-11/785 (KA785) Hardcore Test Stream
ETKAE	VAX-11/785 (KA785) Micro Test Monitor
ETKAF	VAX-11/785 (KA785) Microdiagnostic Command Parser
ETKAG	VAX-11/785 (KA785) Microdiagnostic Directory Search Routine
ETKAH	VAX-11/785 (KA785) Microdiagnostic Volume #1
ETKAJ	VAX-11/785 (KA785) Microdiagnostic Volume #2
ETKAK	VAX-11/785 (KA785) Microdiagnostic Fail Chain Monitor
ETKAL	VAX-11/785 (KA785) Fail Chain Volume #1
ETKAM	VAX-11/785 (KA785) Fail Chain Volume #2
ETKAN	VAX MA780 Microdiagnostic Go Chain
ETKAP	VAX MA780 Microdiagnostic Fail Chain
ETKAR	VAX-11/785 KA785 Microdiagnostic Volume #3
ETKAS	VAX-11/785 KA785 Fail Chain Volume #3
EVKAA	VAX-11 Hardcore Instruction Test
EVKAB	VAX Architectural Instruction
EVKAC	VAX Floating Point Instruction
EVKAD	VAX Compatibility Mode Instruction
EVKAE	VAX Privileged Architecture Instruction

Table K-3 explains the coding convention for VAX diagnostics.

Table K-3 VAX Diagnostic Coding Convention

EPXYD-R.P

E = System

E = VAX

P = Processor

C = VAX-11/750
N = VAX-11/730 and 11/725
S = VAX-11/780
T = VAX-11/785
V = More than one VAX Processor

X = Device Category

A = Application (CR, LP, A-D, Misc.)
C = Channel Adapters (MBA, UBA)
D = Communication Devices (DMC, DZ11)
G = CI Adapter
K = CPU
L = Database Files
M = Magnetic Tape (TE, TU, DT)
P = Test Equipment
Q = QIO Device Driver
R = Rotating Memory (RP, RK, RM, RS, RX)
S = System Monitor (Supervisor, Tape Monitor)
T = Terminal (LA36, VT52)
U = Utilities (Script file, Copy, Update)
W = Workstations
X = System Exerciser (EXR/BUSINT)
Z = Media Packages
0-9 = Engineering File

```
Y = Device Type

   Category Application
           A = LP
           B = CR
           L = LPA11-K

   Category Channel
           A = MBA
           B = UBA
           C = MA780
           D = IEC11, IEU11-A
           E = DT07
           F = DEC DATAWAY
           G = CI780
           H = CI750/CI780 MICROCODE VERIFICATION (INTERNAL)
           I = TEST BED ADAPTERS
           J = NBI ADAPTER

   Category Communications
           A = DZ11,DZ32,DMZ32
           B = DMC
           C = Multiple Devices
           D = DR11-C
           E = DV11
           F = DR780,DF11-W
           G = DQS11
           H = KMC11-B
           I = KMS11-B
           J = DM11-BA
           K = DMS11-BA
           L = DMF32 Combo
           M = DMC11,DMR11,DMP11
           N = DN11
           P = PCL11
           R = DR11B,DR11-W,DRE11
           S = DRS11/DSS
           T = KCT32-A,FEPCM
           U = DUP11
           X = Multiple Devices
           W = UNA

   Category Database Files
           A = CRD DATABASE FILES
           B = SYSTEM EXERCISER DATABASE FILES
```

Category Magnetic Tape
A = TE16/77, TS11, TU78
B = TU81,TU80

Category QIO Device Drivers
For this category, the fourth and fifth letters of the code are the device mnemonic.
For example, EVQDB is the QIO device driver for the DB, or RP disk series.

Category Test Equipment
A = C1780 Node Tester
B = NI Tester

Category Rotating Memory
A = Multiple devices
B = RP04/5/6
C = DCL/RP04/5/6
D = RM03/RM05/RM80
E = RK611,RK06,RK07
F = RL02
G = RM80
H = RP07
I = RX02
J = ML11
K = IDC
L = UDA
M = RC25 (AZTEC)

Category System Monitor
A = Supervisor
B = Autosizer
C = DYDDM

Category Terminals
A = All Terminals
B = Terminal exerciser
C = VS11
D = 11C03-LH Fastek
G = VS70

Category Utility
A = Script File
B = Copy Command file

```
   Category Workstation
           A = VS100
           B = Onyx
           C = Agate,VS125

   Category Exerciser
           B = Exerciser Bus Interaction
```

D = Fifth character makes the code unique. A through Z and 1 through 9.

R = Revision (MAJOR)

P = Revision (MINOR)

Digital Equipment Corporation • Bedford, MA 01730