digital

TIME-SHARING MONITORS:

MULTIPROGRAMMING MONITOR (10/40)
SWAPPING MONITOR (10/50)

PDP-10

TIME-SHARING MONITORS:

MULTIPROGRAMMING MONITOR (10/40)
SWAPPING MONITOR (10/50)

Programmer's Reference Manual

November 1968

Order No. DEC-10-MTE0-D from Program Library, Maynard, Mass. Price $3.00
Direct comments concerning this manual to Software Quality Control, Maynard, Mass.

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

Original Printing April 1967
Reprinted July 1967
Revised November 1967
Reprinted March 1968
Revised May 1968
Revised October 1968

Instruction times, operating speeds and the like are included in this manual for reference only; they are not to be taken as specifications.

CONTENTS

CONTENTS (Cont)

CONTENTS (Cont)

CONTENTS (Cont)

ILLUSTRATIONS

TABLES

TABLES (Cont)

FOREWORD

This manual covers use of the Time Sharing Monitors, which include the Multiprogramming Monitor (formerly known as 10/40) and the Swapping Monitor (formerly known as 10/50).

The Single-User Monitors (formerly known as 10/20, 10/30) are covered in the manual Single User Monitor Systems.

CHAPTER 1
INTRODUCTION

This manual covers commands, program loading and programming of the PDP-10 Time-Sharing Monitors -- two multiprogramming, time-sharing systems designed to allow many independent user programs to share the facilities of the computing system. Such users can access the computer at the same time from consoles located at the computer site, at nearby offices and laboratories, or even at remote consoles connected by telephone lines.

Operating concurrently under Monitor control, these diverse users may access available I/O device and system software to compile, assemble, and execute their programs, or perform this sequence automatically for many jobs by using the batch control processor (Batch). Real-time jobs can operate either as independent user programs or as fully integrated Monitor subroutines.

The Multiprogramming (10/40) Monitor is a multiprogramming, time-sharing system which includes an I/O controller, run-time selection of I/O devices, job-to-job transition, job save and restore features, and memory dump facilities. All of these features are incorporated with concurrent realtime processing, batch processing, and time sharing. The Swapping (10/50) Monitor incorporates all of the features of the 10/40 system and, in addition, swaps programs between high speed disk and core.

1.1 MONITOR FUNCTIONS

Both Monitors schedule multiple-user time sharing of the system, allocate available facilities to user programs, accept input from and direct output to all system I/O devices, and relocate and protect user programs in core memory.

The Monitors utilize the PDP-10 hardware features of memory protection, memory relocation, executive/user mode, and real-time clock to provide an advanced, third-generation, multiprogramming, time-sharing environment. System facilities start with a minimum configuration of 16K core and two DECtapes, and can accommodate magnetic tapes, disks, communication line controllers, card readers, paper tape readers and punches, line printers, displays, plotters, and user Teletype consoles. Other special devices, including real-time digitizers and analog converters, easily interface with the system.

Several programs are loaded into core. The Monitors allow each program to run for a certain length of time, based on a scheduling algorithm which permits the most efficient use of system facilities. The Monitors process input/output commands from the programs, making them device independent, and perform I/O operations concurrently with computation for high system efficiency.

1.2 USER FACILITIES

Users gain access to the system from a console at the facility or remotely located at any point with telephone facilities. Three levels of communication are available at the consoles. Initially, the console communicates with the Monitor Command Interpreter, which provides the system with access protection (LOGIN); allocates and protects memory (CORE) and peripherals (ASSIGN, REASSIGN) requested by the user; provides communication to the operator (TALK) for mounting of special tapes; provides the user with run control (RUN, GET, START, HALT, CONT) over programs stored in the system; allows the user to initiate background jobs (CSTART, CCONT, DETACH, ATTACH); provides the user with job monitoring and debugging (E, D, DDT, REENTER) facilities, and returns facilities to the system (KJOB, DEASSIGN) when the job is finished.

With this set of Monitor commands, the user at his console has access to the system file, which contains programs such as TECO, EDITOR and PIP, for creating and editing program source files, assembling or compiling (MACRO, FORTRAN) program source files, and loading relocatable binary files. The core image of a loaded relocatable binary file may be stored on a retrievable storage device (SAVE) and thereafter be available through the Monitor Command Interpreter. Many other programs are available in the system file to facilitate file management and translation.

1.3 OPERATING TECHNIQUE

When a user starts a program, his console serves as an input/output device, which provides a control and data path to his private program. The console is switched back to the Monitor Command Interpreter by either the program (HALT, EXIT) or by the user striking both the CTRL and C keys (↑C) at the console. The user can exercise another dimension of control over his program by loading it with the powerful Dynamic Debugging Technique (DDT) available in the system file. Entry to DDT is through the Monitor Command Interpreter or by break points from the program. While program control is in DDT, the console permits examining intermediate results and modifying the program (symbolically).

The user's program communicates with the Monitors by means of the PDP-10 operation codes 040 through 077. With these calls, the Monitors provide the program with complete device-independent input/output services, which relieves the programmer of the arduous task of I/O programming, as well as freeing him from dependence on the availability of particular devices at run time. In addition, the user's program may exercise control over central processor trapping (overflow, underflow, pushdown overflow, clock), modify its memory allocation (CORE), and monitor its own running time. Provision exists for inter-job communication and control, reentrant user programs, and, in selected cases, direct user I/O control.

CHAPTER 2
MONITOR COMMANDS

2.1 CONSOLE CONTROL

From the user's point of view, his time-sharing console is in one of three modes: the Monitor mode, the user mode, or the detached mode. In the Monitor mode, characters typed in are presented to the Monitor Command Interpreter. In the user mode, the console acts as an ordinary input/output device under control of the user's program (the DDT submode, a special user mode, is used when running under control of the Dynamic Debugging Technique program). The console is in the detached mode if nothing has been typed on it since the Monitor was started or if the DETACH command is typed. The ATTACH command places it back in Monitor mode.

If the console is in the detached mode and a character is typed in, the console either enters the Monitor mode or immediately responds with "X" or "JOB CAPACITY EXCEEDED," both indicating that the system is at maximum job capacity. It remains in the detached mode. Once in the Monitor mode, each line of text typed in is sent to the Monitor Command Interpreter for processing. If the command is not understood by the Monitor Command Interpreter, an error message is typed out and the console mode is unchanged. Figure 2-1 indicates the console mode at the successful completion of each command.

2.2 COMMAND INTERPRETER AND COMMAND FORMAT

Table 2-1 lists the commands and their characteristics. Each command is a line of ASCII characters. Spaces and nonprinting characters preceding the command name are ignored. The Monitor Command Interpreter ignores a line preceded by a semicolon.

2.2.1 Command Names

Command names are strings of from one to six letters. Characters after the sixth are ignored. Only enough characters to uniquely identify the command need be typed.

2.2.2 Arguments

Arguments follow the command name, separated from it by a space or any printing character that is not a letter or a numeral. Argument formats are described under the associated commands.

If the Monitor Command Interpreter recognizes the command name, but a necessary argument is missing, the Monitor responds with

TOO FEW ARGUMENTS

Extra arguments are ignored.

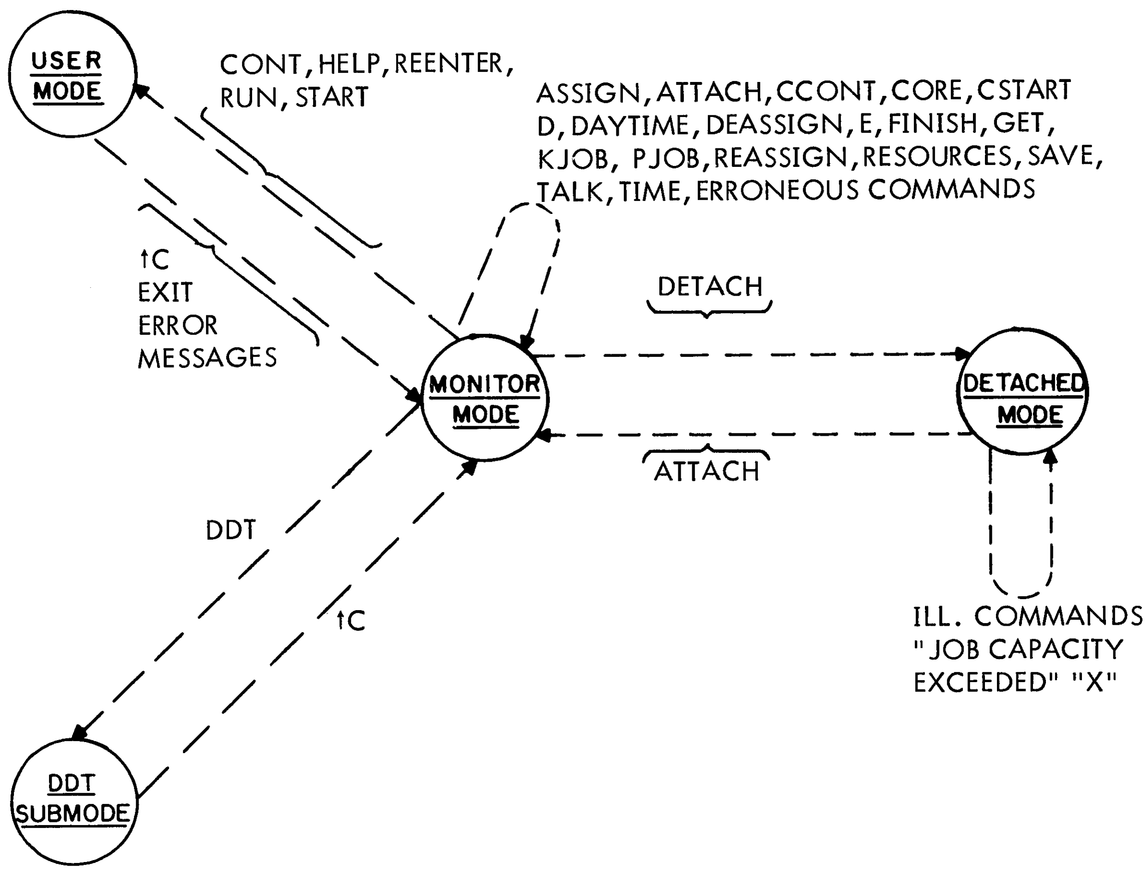


Figure 2-1 Console Teletype Modes

2.2.3 Login Check (10/50 Monitor)

If a user who has not logged in (see "LOGIN Command") types a command requiring the user to be logged in, the 10/50 Monitor responds with

LOGIN PLEASE

The user's command is not executed. Login is not required by the 10/40 Monitor.

Table 2-1
Monitor Commands

Name	Abbre-viation	Arguments 1	2	3	4	5	Console Mode	Characteristics
ASSIGN	AS	dev	dev°				m	L,J
ATTACH	AT	job	[p,p]				m	
ATTACH	AT	dev					m	L
CCONT	CC						m	L,J,C,I
CONT	CON						u	L,J,C,I
↑C	-						m	
CORE	COR	core°					m	J,A,I
CSTART	CS	addr°					m	L,J,C,I
D	D	lh	rh	addr			m	L,J,C,I
DAYTIME	DA						m	
DDT	DD						u (DDT)	L,J,C,I
DEASSIGN	DEA	ldev°					m	L
DETACH	DET	dev°					d	L
E	E	addr°					m	L,J,C,I
FINISH	F	ldev					m	L,J,C,A,I
GET	G	ldev	file	ext°	[p,p]°	core°	m	L,J,A
HELP	H						u	
KJOB	K						m	A
LOGIN	L						u	I
PJOB	P						m	L,J
R	R	file	ext°	core°			u	L
REASSIGN	REA	ldev	job				m	L,J,I
REENTER	REE						u	L,J,C,I
RESOURCES	RES						m	L
RUN	RU	ldev	file	ext°	[p,p]°	core°	u	L
SAVE	SA	ldev	file	ext°	[p,p]°	core°	m	L,J,C,A,I
START	ST	addr°					u	L,J,C,A,I
SYSTAT	SYS						u	
TALK	TA	tty					m	
TIME	TI	job°					m	

°	optional argument	[p,p]	[project number, programmer number] (see "LOGIN Command")
addr	octal address		
core	decimal number of 1K blocks	tty	CTY,OPR,TTY0,...,TTYn
dev	CDR,CTY,DIS,DSK,DTA0,...,DTA7 LPT,MTA0,...,MTA7,OPR,PTP,PTR PTY0,...PTYn,SYS,TTY0,...,TTYn	d	detached
		m	Monitor
		u	user
ldev	dev or a logical device name.	L	LOGIN required (10/50 Monitor)
ext	filename extension, 1 to 3 characters, must be preceded by a point (.)	A	no active devices
		C	core required
file	filename, 6 characters or less	I	must be in core
job	job number assigned by Monitor	J	requires job number (10/40 Monitor)
lh rh	octal value of left and right half words.		

2.2.4 Job Number Check (10/40 Monitor)

If the 10/40 Monitor recognizes a command name which requires a job number and no job number is assigned, the Monitor assigns a job number, n, and responds with,

JOB n

together with a line identifying the Monitor version, and proceeds to execute the command.

2.2.5 Core Storage Check

If the Monitor Command Interpreter recognizes a command name which requires core storage to have been allocated to the job and the job has no core, the Monitor responds with

NO CORE ASSIGNED

The user's command is not executed.

2.2.6 Delayed Command Execution

If the Monitor Command Interpreter recognizes the command name and the job has devices actively transmitting data to or from its core area and the command requires that all devices be inactive, or if the job is swapped out to the disk and the command requires core residence, the Monitor delays execution of the command until the devices are inactive or the job is in core. If another command is typed while a command is waiting, the first command is ignored.

2.2.7 Completion-of-Command Signal

Most commands are processed instantly. The completion of each command is signaled by the output of a carriage return, line feed. If the console is left in Monitor mode, a period follows the carriage return, line feed. If the console is left in user mode, any response other than the carriage return, line feed must come from the user's program.

2.2.8 Program Searching

If the Monitor Command Interpreter does not recognize the command name, the Monitor assumes that it is the name of a program in the system file. If the Monitor cannot find the program in the system file, it responds with the name, followed by

NOT FOUND

If the program is found, the Monitor loads the program into core and starts it with the console in user mode.

2.3 SYSTEM ACCESS CONTROL COMMANDS (10/50 Monitor System Only)

Access to the system is limited to authorized personnel. The system administrator provides each user with a project number, a programmer number, and a password. The project and programmer numbers are octal numbers up to nine digits each. The password is a sequence of from one to five ASCII characters, which must match the password stored in the system accounting file to LOGIN successfully.

2.3.1 HELP Command

The HELP command carries on a dialogue with the user at the console to explain the use of the system. No knowledge of the system is assumed on the part of the user.

2.3.2 LOGIN Command (10/50 Monitor)

LOGIN waits for the user to type in the project and programmer numbers on a line, separated by a comma (,). If the project-programmer number is incorrect, Monitor responds with the error message

"?SORRY---WRONG NUMBER "

LOGIN then responds with three overprinted lines of random characters intended to make the next line of input illegible. The user types his password onto this mask. If it matches the system password, LOGIN responds with

↑C

leaving the console in Monitor mode. If the password does not match, LOGIN responds with the error message

"?INCORRECT CODE---TRY AGAIN"

and waits for the project-programmer number, password combination to be typed again.

Example

.LOGIN↵ (or .L↵)	User issues LOGIN command.
JOB n	Monitor responds with job number assigned.
xxx..xx	Monitor types out its version designation
proj,prog↵	User types in his project-programmer number (each number can contain up to nine octal digits).
ABCDE	Monitor types out password mask; user types in his password over the mask.
↑C .	If user entries are correct, Monitor responds with ↑C and a period, indicating readiness to accept a command.

2.3.3 SYSTAT Command (10/50 Monitor)

SYSTAT prints a summary of the current system status on the user's console.

2.4 FACILITY ALLOCATION COMMANDS

One of the functions of the Monitor is to allocate peripheral devices and core memory to users upon request, and to protect allocated facilities from interference by other users. To this end, the Monitor maintains a pool of available facilities from which a user can draw and restore by request.

A user should never abandon a time-sharing console without returning allocated facilities to the pool.

2.4.1 Device Descriptors

The devices controllable by the system are listed in Table 5-1. Associated with each device is a physical name, made up of three letters and zero to three numerals to specify unit (transport) number. All references to devices in the Monitor are made by these physical names or by assigned logical names.

2.4.2 ASSIGN dev ldev°

ASSIGN has one required argument, dev (device), and one optional argument, ldev. Dev must be a physical device name, or DTA or MTA. If dev is DTA or MTA, the Monitor searches the device pool for a free unit. Monitor responses are:

DEVICE dev ASSIGNED	(physical device dev was free and has been assigned to the user)
NO SUCH DEVICE	(all units are in use)
ALREADY ASSIGNED TO JOB n	(dev is allocated to another job, n)

2.4.2.1 Logical Device Names (ldev) - The second argument, ldev, is optional. It represents a logical device name of one to six alphanumeric characters of the user's choice, usable synonymously with dev in all references to the device. Logical device names take precedence over physical device names. Thus, a user may write programs to use arbitraily named devices which he assigns to the most convenient physical devices at run time.

If the user has the name ldev assigned to another device, the Monitor responds with

LOGICAL NAME ALREADY IN USE
DEVICE dev ASSIGNED

2.4.2.2 Examples

User types	ASSIGN DTA, ABC	
Monitor responds	DEVICE DTA6 ASSIGNED	(successful)

User then types	ASSIGN DTA,DEF	(find another unit)
Monitor responds	NO SUCH DEVICE	(all in use)
User then types	ASSIGN PTP, ABC	(reserve paper tape punch)
Monitor responds	LOGICAL NAME ALREADY IN USE DEVICE PTP ASSIGNED	(paper tape punch is reserved, but ABC still refers to DTA6 only)
User then types	ASSIGN DTA1, DEF	
Monitor responds	ALREADY ASSIGNED TO JOB 2	(another user has it)

2.4.2.3 ASSIGN SYS: dev - This command is used to change the systems device (SYS:) from its currently allocated device to some other device (dev). In order to issue this command, the user must be logged in under either [1,1] or [1,2].

2.4.2.4 Device Protection - When a device is assigned to a job, it is removed from the Monitor's pool of available devices. Any attempt by another job to reference the device fails. The device is returned to the pool when the user deassigns it or kills the job.

2.4.2.5 Special Functions - The ASSIGN command applied to DECtapes clears the copy of the directory currently in core, forcing any directory references to read a new copy from the tape. This is especially important when changing reels. (See Chapter 5 for further details.)

2.4.3 DEASSIGN ldev°

This command cancels device reservations made via the ASSIGN command and returns the device(s) to the Monitor pool. The command may be typed alone or with one argument, ldev. When an argument is typed, it must be the logical or physical name of some device previously reserved by the ASSIGN command. If no argument is typed, all devices currently reserved by the user via the ASSIGN command are affected. The DEASSIGN command may be typed, even though the user's program continues to use the devices affected.

Monitor error responses are:

NO SUCH DEVICE
DEVICE WASN'T ASSIGNED

2.4.3.1 Special Functions - The DEASSIGN command applied to DECtapes performs the same special function as ASSIGN, section 2.4.2.5.

2.4.4 REASSIGN dev job

REASSIGN allows one job to pass a device to a second job without going through the Monitor pool. Two arguments are required: the physical device name, dev, and the job number of the second job. Dev is deassigned from the current job and assigned to the second job. All devices except user consoles can be reassigned.

Monitor error responses are:

DEVICE dev WASN'T ASSIGNED
JOB NEVER WAS INITIATED
NO SUCH DEVICE
DEVICE CAN'T BE REASSIGNED

2.4.5 FINISH ldev

FINISH terminates any input or output currently in progress on device ldev and relinquishes it (see RELEASE).

Monitor error response is:

NO SUCH DEVICE

2.4.6 TALK tty

The TALK command allows a user to type directly on another user's console, and the latter to type back. If device tty is in the detached mode or in Monitor mode and at the left margin, the user's console is inserted into a talk "ring" with tty. Otherwise the Monitor responds with BUSY. Any number of consoles can be in the same talk ring. Each character typed on any console in the ring is printed on all other consoles in the ring. Any console is removed from the ring by typing ↑C. The required argument, tty, can be any of the physical device names CTY, TTY0,...,TTYn or the special device name OPR.

2.4.6.1 Operator's Console - When the Monitor is started, one console, usually CTY, is designated as the operator's console and given the name OPR. All requests for local operations such as mounting and unmounting tapes, etc., can be performed with TALK OPR.

2.4.7 CORE core°

The CORE command has one optional argument, core. Without the argument, the Monitor responds with the decimal number of 1024-word blocks of unallocated core in its pool if 10/40 system and with the maximum size of user's core if 10/50 system. The optional argument, core, is the total decimal number of 1024-word blocks of core memory allocated to the job upon successful completion of the command. If it is smaller than the current allocation, the difference is removed from the top of the user's core area, and returned to the Monitor pool. If it is larger than the current allocation, the difference, if available, is removed from the pool and appended to the top of the user's core area. In the 10/40 (nonswapping) system, if the difference is not available, the user's current core area is unchanged, and the Monitor responds with the decimal number of 1024-word blocks in the pool. In the 10/50 (swapping) system, if the difference is not available, the user program is swapped out and brought back a short time later when it can fit. The user need not know if his program is swapped out or not.

2.4.8 RESOURCES

This command causes the typeout of all available devices (except Teletypes) and the number of free blocks on the disk.

2.5 RUN CONTROL COMMANDS

Core image files located on retrievable storage devices such as disk, DECtape, and magnetic tape can be retrieved and controlled from the user's console. The process of creating such files is described in Chapter 3. Files stored on disk and DECtape are addressable by name. Files on magnetic tape require prepositioning the tape by the user.

2.5.1 File Descriptors

2.5.1.1 Filenames - Filenames are from one to six letters or digits. All letters and digits after the sixth are ignored. A filename is terminated by any character that is not a letter or digit.

2.5.1.2 Filename Extension - If the filename is terminated by a period, a filename extension is assumed to follow. A filename extension is from one to three letters or digits. It is generally used to indicate file format. The filename extension is terminated by any character not a letter or a digit. If a filename extension is not specified with the RUN, GET, and SAVE commands, an extension of DMP is assumed.

2.5.1.3 Project-Programmer Numbers - If a user wants to perform a RUN or GET command on a disk file belonging to another user, he must specify the user's project-programmer numbers. The format is

[project-number, programmer-number]

2.5.2 RUN ldev file ext° [p,p]° core°

The RUN command loads a core image from a retrievable storage device (DECtape, disk, and magnetic tape), ldev, and starts it at a location specified within the file (see JOBSA, "Job-Data Area", Chapter 3). The arguments file, ext, and [p,p] are used to select the file. The minimum amount of core required to load the file is allocated. After the file is loaded, core is reallocated if the optional fifth argument, core, is specified or if the file was saved with a core argument. If both were specified, the RUN command core argument takes precedence. The optional argument is ignored if it is less than the size of the file. If ldev is a magnetic tape, the fifth argument must be specified, and be at least as large as the core image file to assure proper loading.

Monitor error responses are:

ldev NOT AVAILABLE	(ldev is allocated to another job)
NO SUCH DEVICE	(ldev is undefined)
nK OF CORE NEEDED	(where n is a decimal number of 1024-word blocks, if there is insufficient free core to load the file or to satisfy the optional core argument on the reallocation) - 10/40 Monitor only.
NOT A DUMP FILE	(the selected file is not a core image file)
TRANSMISSION ERROR	(a parity or device error occurred during data transmission)

2.5.3 R file.ext° core° - The R command is equivalent to the command

RUN SYS file.ext° core°

and is provided as a convenience for the user. In other words, the R command is the usual command for running one of the CUSPs (Commonly Used System Programs) in the system library. Note that R is not an abbreviation for RUN; if the program is on a device other than SYS, the user must use the RUN command (abbreviated RU).

2.5.4 GET, START, HALT (↑C), and CONT Commands

The GET, START, HALT, and CONT commands permit the user to control the running of his program from the console.

2.5.4.1 GET ldev file ext° [p,p]° core° - The GET command is the same as the RUN command, except that the Monitor responds with

JOB SETUP

instead of starting the program. The assignment of core is also similar to that of the RUN command.

2.5.4.2 START addr° - The START command begins execution of the user's program. If the optional argument, addr, is not specified, the starting address is found in the core area (right half of JOBSA, see Chapter 3). The optional argument, addr, is an octal number and, if specified, the program is started at that location. The only Monitor error response is:

NO CORE ASSIGNED

2.5.4.3 HALT and ↑C - Typing a ↑C (hold down the CTRL key and strike "C") on the console puts the console in Monitor mode and transmits a HALT command to the Monitor Command Decoder. The HALT command stops the job and stores the program counter in the job's core area (JOBPC, "Job-Data Area," Chapter 3).

2.5.4.4 CONT - The CONT command starts the program at the location specified by the contents of the saved program counter in the job's core area (JOBPC, see "Job-Data Area," Chapter 3), and puts the console in user mode. If the CONT command is given to a job which was stopped as a result of a Monitor-detected error, the Monitor responds with

CAN'T CONTINUE

The CONT command is applicable only if the job was stopped by the HALT (↑C) command or the HALT instruction.

2.5.5 DDT, REENTER, E and D

The DDT, REENTER, E, and D commands are used primarily for program debugging and exception handling. The DDT and REENTER commands provide alternate program entry points. E and D provide a means of examining and modifying locations in the user's core area from the console.

2.5.5.1 DDT - The DDT command copies the saved value of the user's program counter (JOBPC) into a second location in his core area (JOBOPC, see "Job Data Area," Chapter 3), and starts his program at an alternate entry point specified by another location (JOBDDT, see "Starting Addresses," Chapter 3) in his core area. This alternate entry point is set to the beginning address of DDT by the loader, if the program was loaded with DDT. Alternately, the user may set this address to any desired location. To resume computation following the DDT command interruption, execute a ↑C and START (JRST 2, @ JOBOPC).

2.5.5.2 REENTER - The REENTER command is similar to the DDT command. The alternate entry point is specified by a different location (JOBREN, see "Job Data Area," Chapter 3) in the job core area, and must be set by the user or his program.

A typical use of this command is interrupting a long computation to examine intermediate results. The user types ↑C, and then REENTER, which transfers to his routine to print intermediate results. This routine should preserve the state of his main program, and return to the interrupted computation by executing a JRST 2,@ JOBOPC.

2.5.5.3 E addr° - The E command allows the user to examine locations in his core area. If the optional argument, addr, which is an octal number, is specified, the octal contents of the left and right halves of location addr are typed. Leading zeros in the half words are suppressed. The half-word values are separated by a space, and the right half value is followed by a horizontal tab. If the optional argument, addr, is not specified, the contents of the next location are typed. If the location to be examined lies outside the user's allocated core area, the Monitor responds with

OUT OF BOUNDS

2.5.5.4 D lh rh addr° - The D command allows a user to deposit into his core area. The required arguments lh and rh are the octal values of the left and right half words to be deposited. If the optional argument, addr, which is an octal number, is specified, the word is deposited at location addr. If it is not specified, the word is deposited at the location following the last location examined or deposited. If the location is above the user's core area, or in the protected part of the job data area (see Table 3-1) above user AC 17, the Monitor responds with

OUT OF BOUNDS

2.5.6 SAVE ldev file ext° core°

The SAVE command writes a core image file of the user's core area. If DDT is loaded, i.e., if JOBDDT is nonzero (see Chapter 3), the entire core area is written. Otherwise, the area starting from JOBDDT and extending up through the program break (as specified by the contents of JOBFF, see Chapter 3) is written. If the optional argument, ext, is not specified, the filename extension is SAV. The optional argument, core, specifies the minimum number of 1024-word blocks in which the program is to be run. This parameter is stored in the job's core area (JOBCOR, see Table 3-1), and is used by the RUN and GET commands. The state of the users accumulators and input/output devices are not saved.

After the output is completed, the Monitor responds with:

JOB SAVED

Monitor error responses are:

n 1K BLOCKS OF CORE NEEDED	(where n is the contents of JOBFF modulo 1024, if the user's current core allocation is less than the contents of JOBFF)

DEVICE NOT AVAILABLE	(device ldev is allocated to another user)
TRANSMISSION ERROR	(an error was detected while writing)
DIRECTORY FULL	(the maximum number of files already exists for device ldev)

2.6 BACKGROUND JOB CONTROL

A job is a "background" job if it is not under control of a user console. Any console can initiate any number of background jobs. Input/output to the console while a job is running in background mode causes the job to stop until a console is attached.

2.6.1 PJOB

The PJOB command responds with the job number to which the user's console is attached. If the console is not attached to a job, the 10/40 Monitor assigns a job number, and responds with the job number and a line identifying the Monitor version; the 10/50 Monitor responds with the message

LOGIN PLEASE

The job number is a necessary argument for the ATTACH command.

2.6.2 CSTART and CCONT

The RUN, START, and CONT commands always leave the user console in user mode. ↑C switches the console to Monitor mode, but also stops the job. The CSTART and CCONT commands are identical to the START and CONT commands, respectfully, with the exception that the console is left in Monitor mode.

In general, to start a job running with the console in Monitor mode, it is necessary to begin with the console in user mode; type control information to the program; type ↑C, which stops the job with console in Monitor mode; and, finally, type the CCONT command, which allows the job to continue running with the console in Monitor mode. Further commands may now be executed while the job is running.

2.6.3 DETACH

The DETACH command disconnects the user's console from the job, placing the console in the detached mode without affecting the status of the job. For instance, if the job was running, it remains running in the background mode. The user console is now free to control another job, either by creating a new job or ATTACHing to a background job.

2.6.3.1 DETACH dev - This command causes the assignment of device dev to JOB 0, thus making it unavailable to the system. In order to issue this command, the user must be logged in under [1,1].

2.6.4 ATTACH job [p,p]

The ATTACH command allows a user to connect a console to a background job. Two arguments are required. The first argument, job, is the job number of the job to which the user desires to attach. The second argument, [p,p], is the project-programmer number pair of the originator of the desired job. Following the ATTACH command, the console is always left in the Monitor mode. If the job happens to be running, typing CONT places the console in the user mode without affecting the operation of the job. It is not necessary to execute the DETACH command before the ATTACH command, in order to switch the console between two jobs, since the current job is automatically DETACHed.

If an error condition occurs, the console is left attached to the job to which it was connected before the ATTACH was typed.

Monitor error responses are:

TTYn ALREADY ATTACHED	(either the job number typed is erroneous and by coincidence is attached to another console, or another user is attached to the job number specified)
JOB NEVER WAS INITIATED	(the specified job number is not assigned to any job)
NOT JOB ORIGINATOR	(the second argument, [p,p] is not the project-programmer pair of the job originator)

2.6.4.1 ATTACH dev - This command returns a detached device to the Monitor pool and makes it available to the system. In order to issue this command, the user must be logged in under [1,1].

2.7 JOB TERMINATION

When a user leaves the system, all facilities allocated to his jobs must be returned to the Monitor facility pool, thereby making them available to other users.

2.7.1 KJOB

The KJOB command performs the following functions on the job to which the console is attached:

a. Stops all allocated input/output devices and returns them to the Monitor pool;

b. Returns all allocated core to the Monitor pool;

c. Returns the job number to the Monitor pool;

d. Performs a TIME command; and

e. Leaves the console in Monitor mode.

2.8 SYSTEM TIMING

All system times are kept in increments of one 60th (or 50th) of a second. The DAYTIME and TIME commands print time in the format

hhmm:ss.ss

where hhmm is a 4-digit representation of hours and minutes and ss.ss is seconds to the nearest hundreth.

2.8.1 DAYTIME

The DAYTIME command prints the date followed by the time of day.

2.8.2 TIME job°

The TIME command prints the incremental running time, i.e., the running time since the last TIME command, followed by the total running time used by the job. Interrupt level and job scheduling times are charged to the job running when the interrupt or rescheduling occurred. If the optional argument, job, is not specified, the job to which the console is attached is used. If the optional argument is zero, an approximation of the time spent core shuffling is printed, followed by the running time of the null job and the total system up time.

CHAPTER 3
LOADING USER PROGRAMS

3.1 MEMORY PROTECTION AND RELOCATION

A user's program runs while the computer is in a special mode known as the user mode. In this mode, the contents of the memory relocation register in the central processor are automatically added to each memory address before the address is sent to the memory system. The address, before this addition takes place, is called the relative address; after the addition, the address is called the absolute address. The contents of the memory protection register are compared with the eight high-order bits of each relative address. If the relative address exceeds the contents of the memory protection register, the memory violation flag is set in the central processor and control traps to the Monitor.

Thus, the contents of the memory protection and relocation registers define a contiguous area of core with the following properties:

a. All memory references from within the region are relative to the beginning of the region.

b. It is impossible to address a location outside the region from within the region.

When the Monitor schedules a user's program to run, it sets the memory protection and relocation registers to the bounds of the user's allocated core area and switches the central processor to the user mode.

In this manual, all addresses in the user's area are relative addresses.

To take advantage of the fast accumulators, memory addresses 0 through 17 are not relocated. Thus, relative locations 0 through 17 cannot be referenced by the user's program. The Monitor saves the user's accumulators in this area when the user's program is not running and while the Monitor is servicing a program call from the user.

3.2 USER'S CORE STORAGE

A user's core storage consists of a single contiguous block of memory whose size is an integral number of 1024 words (see Figure 3-1). There are two methods available to the user for loading his core area. The simplest way is to load a core image stored on a retrievable device (see RUN and GET, Chapter 2); the second is to use the relocatable binary loader to link-load binary files. The user may then write the core image on a retrievable device for future usage (see SAVE, Chapter 2).

3.2.1 Job Data Area

The first 140_8 locations of the user's core area comprise the job data area reserved for storing specific information concerning the job, such as the starting address of the user's program (JOBSA), highest legal address (JOBREL), etc. Locations in this area have been given mnemonic assignments whose first three characters are JOB, e.g., JOBSA, JOBFF, JOBDDT, etc (see Table 3-1). As a consequence all mnemonics in this manual with a JOB prefix refer to locations in the job data area.

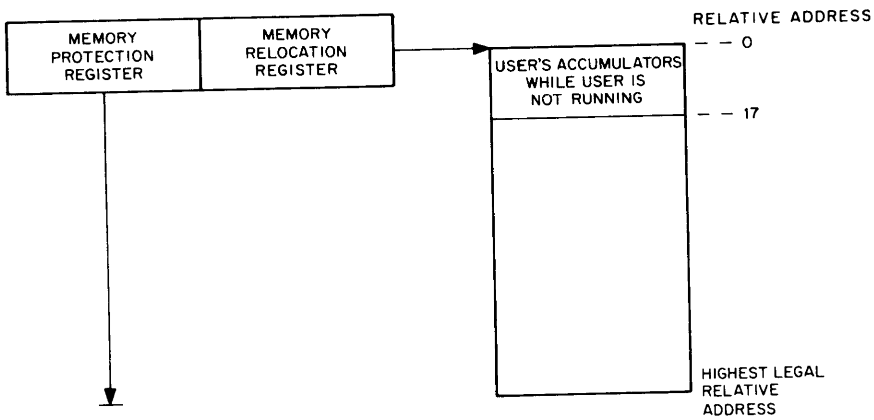


Figure 3-1 User's Core Area

Table 3-1
Job Data Area Locations

Name	Relative Location(s) Octal	Description
JOBUUO	40	User's location 40_8. Used for processing user UUO's (001 through 037).
JOB41	41	User's location 41_8. Contains the beginning address of the user's programmed operator service routine.
JOBREL	44	Left half: 0 Right half: The highest relative core location available to the user (i.e., the contents of the memory protection register when this user is running).
JOBDDT	74	Contains the starting address of DDT. If contents are 0, DDT has not been loaded.
JOBPFI	114	Highest location in the job data area protected from I/O, that is, the Monitor will not perform I/O into or out of locations 0 through JOBPFI.
JOBSYM	116	Contains a pointer to the symbol table created by Linking Loader. Left half: Negative count of the length of the symbol table. Right half: Lowest register used.
JOBSA	120	Left half: First free location in user area (set by Loader). Right half: Starting address of user's program.

Table 3-1 (Cont)
Job Data Area Locations

Name	Relative Location(s) Octal	Description
JOBFF	121	Left half: 0 Right half: Address of the first free location following the user's program. Set to $C(JOBSA)_{LH}$ by RESET UUO.
JOBREN	124	Set by user and used by REENTER command as an alternate entry point.
JOBAPR	125	Contains user location to be trapped to when APR trap occurs (see APRENB UUO, Section 4.3.3.1).
JOBCNI	126	Set by CONI APR when an APR trap occurs to user program so that it can see APR flags (see APRENB UUO).
JOBTPC	127	APR trap PC stored here on APR trap to user program so that execution can be continued (see APRENB UUO).
JOBOPC	130	The previous (old) contents of the user's program counter are stored here by Monitor upon execution of a DDT, REENTER START, or CSTART command.
JOBCHN	131	Left half: 0 Right half: Address of first location after first FORTRAN IV Block Data.
JOBCOR	133	Left half: Unused Right half: Highest core address for SAVE, GET, and RUN (i.e., user's 3rd argument).

NOTE: Only those JOBDAT locations of significant importance to the user are given in this table. JOBDAT locations not listed include those which are used by the Monitor and those which are unused at the present time.

Some locations in the job data area, such as JOBSA and JOBDDT, are set by the user's program for use by the Monitor. Others, such as JOBREL, are set by the Monitor for use by the user's program. In particular, the right half of JOBREL contains the highest legal address set by the Monitor whenever the user's core allocation changes.

User programs must reference locations in the job data area with the assigned mnemonics, which must be declared as EXTERNAL references to the assembler. The values are assigned when the loader performs an automatic library search for undefined global references. The specific library subfile, in which these symbols are defined, is called JOBDAT.

3.2.2 Loading Relocatable Binary Files

The relocatable binary loader (LOADER) resides in the system file, and is started by the command

R LOADER core° Example: R LOADER 5

The PDP-10 Systems User's Guide contains a description of the loader command string. Figure 3-2 shows the user's core area with the loader resident.

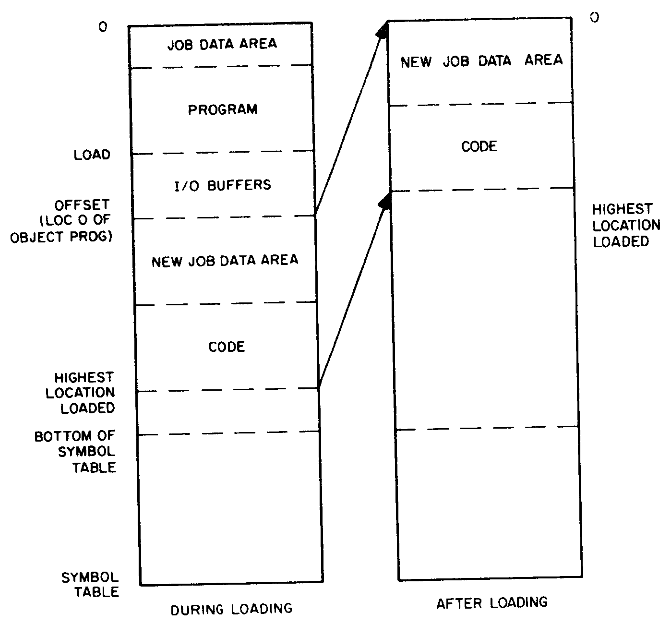


Figure 3-2 Loading User Core Area

3.2.2.1 Program Origin - The new program code is loaded upward from an offset above the resident loader. The program origin (i.e., the first location loaded) is 140_8, unless the user changes it by the assembler LOC pseudo-instruction. The symbol table is built down from the top of the allocated core area. If the code and symbol table overlap, the core area is expanded by 1024 words and the symbol table is moved up to the top of the expanded area. Upon completion of loading, the loader stores some values in the new job data area, and moves the area from the offset to the highest location loaded (top of new code) down to zero. The symbol table remains at the top of the allocated core.

3.2.2.2 Program Break - After loading, the address of the first location above the new code area (i.e., the program break) lies in the left half of location JOBSA and in the right half of JOBFF. The left half of JOBFF contains 0.

3.2.2.3 Starting Addresses - The right half of JOBSA contains the program starting address. The value is the last nonzero address field of the assembler END pseudo-instruction to be loaded, or 0. This is the address used by the RUN and START commands.

If DDT was loaded by means of the D switch in the loader command string, the right half of JOBDDT is set by DDT to the starting address of DDT; the left half is 0; otherwise, the contents of JOBDDT are zero, the DDT command uses this address as the starting address. Location JOBREN may be set by the user's program for use with the REENTER command (see Chapter 2).

3.2.2.4 Symbol Table - JOBSYM contains a pointer to the bottom of the symbol table. The left half is the negative word length of the table, and the right half is the address of the lowest location used. The top of the symbol table is the top of the user's allocated core area, pointed to by the contents of the right half of JOBREL. DDT uses this symbol table for printing and interpreting symbolic values.

The right half of JOBUSY is the beginning address of the list of undefined global symbols. If some symbols are undefined after loading is complete, DDT may be used to define their values. These values are automatically substituted by DDT in all locations referencing them.

CHAPTER 4
USER PROGRAMMING

The central processor operates in one of three modes: executive mode, user I/O mode, or user mode. The Monitor operates in executive mode, which is characterized both by the lack of memory protection and relocation (see Chapter 3) and by normal execution of all defined operation codes. The user I/O mode is a special mode, wherein memory protection and relocation are in effect, as well as the normal execution of all defined operation codes. (This mode is not used by the Monitor, and is not normally available (see TRPSET) to the time-sharing user.) User programs are run in user mode, to guarantee the integrity of both the Monitor and each user program.

4.1 USER MODE

The user mode of the central processor is characterized by the following features:

a. Automatic memory protection and relocation (see Chapter 3)

b. Trap to absolute location 40 on

   (1) Operation codes 40 through 77 and 0;

   (2) Input/output instructions (DATAI, DATAO, BLKI, BLKO, CONI, CONO, CONSZ, and CONSO);

   (3) HALT (i.e., JRST 4,); or

   (4) Any JRST instruction that attempts to enter executive mode or user I/O mode.

c. Trap to relative location 40 on execution of operation codes 001 through 037.

Since user programs run in user mode, the Monitor must perform all input/output operations for the user, as well as any other operations required by the user not available in the user mode.

4.2 PROGRAMMED OPERATORS (UUO's)

Operation codes 000 through 077 are programmed operators (sometimes referred to as UUO's - Unimplemented User Operators); some trap to the Monitor and the rest trap to the user program.

After the effective address calculation is complete, the contents of the instruction register are stored in user or Monitor location 40, along with the effective address, and the instruction in user or Monitor location 41 is executed out of normal sequence. Location 41 must contain a JSR instruction to a routine to interpret the contents of location 40.

4.2.1 Operation Codes 001-037 (User UUO's)

Operation codes 001 through 037 do not effect the mode of the central processor. Thus, when executed in user mode, they trap to user location 40, which allows the user complete freedom in the use of these programmed operators.

4.2.2 Operation Codes 040-077, and 0 (Monitor UUO's)

Operation codes 040 through 077 and 0 trap to absolute location 40, with the central processor in executive mode. These programmed operators are interpreted by the Monitor to perform input/output operations and other control functions for the user's program.

Table 4-1 lists the operation codes and their mnemonics.

4.2.3 Operation Codes 100-127 (Unimplemented Op Codes)

Op code 100-UJEN	Dismisses realtime interrupt from user mode (see 4.3.6.2).
Op codes 101-127	Monitor prints ILL INST AT USER n and stops job.

4.2.3.1 CALL and CALLI - Operation codes 040 through 077 limit the Monitor to 40_8 operations. The CALL operation extends this set by specifying the name of the operation by the contents of the location specified by the effective address, e.g., CALL [SIXBIT/EXIT/] . This provides for indefinite extendability of the Monitor operations, at the overhead cost to the Monitor of a table lookup.

The CALLI operation eliminates the table lookup of the CALL operation by having the programmer perform the lookup once, and specifying an index to the operation in the effective address of the CALLI. Table 4-2 lists the Monitor operations specified by the CALL and CALLI operations.

4.2.4 Illegal Operation Codes

The eight input/output instructions (DATAI, etc.) and JRST instructions attempting to enter executive or user I/O mode from the user mode are interpreted by the Monitor as illegal instructions. The job is stopped and the following error message is printed on the user's console.

ERROR IN JOB n

ILL INST AT USER LOC addr

4.3 PROGRAM CONTROL

4.3.1 Starting

All program starting is accomplished by the Monitor commands RUN, START, CSTART, CONT, CCONT, DDT, and REENTER (see Chapter 2). The starting address is either an argument of the command or stored in the user's job data area (see Chapter 3).

4.3.1.1 CALL AC, [SIXBIT/SETDDT/] or CALLI AC, 2 - This UUO causes the contents of the AC to replace the DDT starting address, which is stored in the protected job data area location, JOBDDT. This starting address is used by the Monitor command, DDT.

4.3.2 Stopping

Any one of the following procedures can stop a running program:

a. ↑C from the user console (see Chapter 2);

b. A Monitor detected error; or

c. Program execution of HALT, CALL [SIXBIT/EXIT], or CALL [SIXBIT/LOGOUT/].

4.3.2.1 Illegal Instructions (700-777, JRST 10, JRST 14) and Unimplemented Op codes (101-127) - Illegal instructions trap to the Monitor, stop the job, and print

ERROR IN JOB
ILL. INST. AT USER n

Note that the program cannot be continued by typing the CONT or CCONT commands.

4.3.2.2 HALT or JRST 4, - The HALT instruction is an exception to the illegal instructions; it traps to the Monitor, stops the job, and prints

ERROR IN JOB
HALT AT USER n

However, the CONT and CCONT commands are still valid and, if typed, will continue the program at the effective address of the HALT instruction. HALT is useful for impossible error returns such as INIT on TTY.

Table 4-1
Monitor Operation Codes

Operation Code	Mnemonic	Function	
040	CALL	Operation code extension	
041	INIT	Initialize I/O device	
042		No operation	Reserved for installation-dependent calls
043		No operation	
044		No operation	
045		No operation	
046		No operation	
047	CALLI	Operation code extension	
050	OPEN	Open file	
051		No operation	Reserved for future expansion
052		No operation	
053		No operation	
054		No operation	
055	RENAME	Rename or delete a file	
056	IN	Input and Skip	
057	OUT	Output and Skip	
060	SETSTS	Set file status	
061	STATO	Skip on file status one	
062	STATUS	Read file status	
063	STATZ	Skip on file status zero	
064	INBUF	Set up input buffer ring	
065	OUTBUF	Set up output buffer ring	
066	INPUT	Read	
067	OUTPUT	Write	
070	CLOSE	Close file	
071	RELEASE	Release device	
072	MTAPE	Position tape	
073	UGETF	Get next free block number	
074	USETI	Set next input block number	
075	USETO	Set next output block number	
076	LOOKUP	Select file	
077	ENTER	Create file	
100	UJEN	Dismiss real-time interrupt	

Table 4-2
CALL and CALLI Monitor Operations

CALLI AC, x	CALL AC, [SIXBIT/y/]	Function
x = 0	y = RESET	Reset I/O devices (See 4.4.2.1)
1	DDTIN	DDT mode console input (See 5.1.2)
2	SETDDT	Set protected DDT starting address (see 4.3.1.1)
3	DDTOUT	DDT mode console output (See 5.1.2)
4	DEVCHR	Get device characteristics (See 5.11)
5	(DDTGT)	No operation
6	(GETCHR)	Same as DEVCHR(4)
7	(DDTRL)	No operation
10	WAIT	Wait until device inactive (See 4.4.6)
11	CORE	Allocate core (See 4.5)
12	EXIT	Release devices, stop job (See 4.3.2.2)
13	UTPCLR	Clear directory (See Table 5-2)
14	DATE	Return date (See 4.3.4.1)
15	LOGIN	Special operation for LOGIN (See 4.3.5.3)
16	APRENB	Enable central processor traps (See 4.3.3.1)
17	LOGOUT	Kill job (See 4.3.2.3)
20	SWITCH	Read processor console switches (See 4.3.6.3)
21	REASSI	Reassign device (See 2.4.4)
22	TIMER	Read clock in ticks (See 4.3.4.2)
23	MSTIME	Read clock in milliseconds (See 4.3.4.3)
24	GETPPN	Read project-programmer pair (See 4.3.5.2)
25	TRPSET	Set trap for user I/O mode (See 4.3.6.1)
26	TRPJEN	Illegal UUO
27	RUNTIM	Return job running time (See 4.3.4.4)
30	PJOB	Return job number (See 4.3.5.1)
31	SLEEP	Stop job for specified time (See 4.3.4.5)
32	(SETPOV)	Set pushdown overflow trap (this command has been superceded by APRENB (16).
33	PEEK	Return specified Monitor location (See 4.3.5.4)
34	GETLIN	Return physical name of attached Teletype console. (See 4.3.5.5)

NOTE: Other CALLI UUOs will be implemented from time to time and will be documented in Software Manual Updates and in revised editions of this manual. Execution of a CALLI UUO with an address higher than the last implemented operator will result in an ILLEGAL UUO message.

4.3.2.3 CALL [SIXBIT/EXIT/] or CALLI 12 - All input/output devices are RELEASed (see Section 4.4.7), and the job is stopped.

EXIT
↑C

is printed on the user's console, which is left in Monitor mode. The CONT or CCONT commands cannot continue the program.

4.3.2.4 CALL [SIXBIT/LOGOUT/] or CALLI 17 - All input/output devices are RELEASed (see Section 4.4.7), and returned to the Monitor pool, along with the allocated core and the job number. The accumulated running time of the job is printed on the user's console, which is left in the detached mode.

4.3.3 Trapping

4.3.3.1 CALL AC, [SIXBIT/APRENB/] or CALLI AC, 16 - APR trapping allows a user to handle any and all traps that occur on the central processor, including illegal memory references, nonexistent memory references, pushdown list overflow, arithmetic overflow, floating point overflow, and clock flag. To enable for trapping a CALL AC, [SIXBIT/APRENB/] or CALLI AC, 16 is executed, where the AC contains the central processor flags to be tested on interrupts, as defined below:

AC Bit		Trap On
19	200000	pushdown overflow*
22	20000	memory protection violation*
23	10000	nonexistent memory flag*
26	1000	clock flag*
29	100	floating point overflow
32	10	arithmetic overflow

When one of the specified conditions occurs while the central processor is in user mode, the state of the central processor is Conditioned Into (CONI) location JOBCNI, and the PC is stored in location JOBTPC in the job data area (see Table 3-1). Then control is transferred to the user trap-answering routine specified by the contents of the right half of JOBAPR, after the arithmetic overflow flag has been cleared. The user program must set up location JOBAPR before executing the CALL AC [SIXBIT/APRENB/] or CALLI AC, 16. To return control to his interrupted program, the user's trap answering routine must execute a JRST 2, @ JOBTPC to restore the state of the processor.

4.3.3.2 Console-Initiated Traps - Program control can be changed from the user's console by use of the ↑C, START, DDT, and REENTER commands (see Chapter 2).

*The Monitor is always enabled for these.

4.3.4 Timing Control

The central processor clock, which generates interrupts at the power-source frequency (60 Hz in North America, 50 Hz in most other countries), keeps time in the Monitor. Each clock interrupt (tick) corresponds to 1/60th (or 1/50th) of a second of elapsed real time. The clock is set initially to the current time of day by console input when the system is started, as is the current date. When the clock reaches midnight, it is reset to zero, and the date is advanced.

4.3.4.1 CALL AC, [SIXBIT/DATE/] or CALLI AC, 14 - A 12-bit binary integer computed by the formula

date=((year-1964)x12+(month-1))x31+day-1

represents the date.

This integer representation is returned right-justified in accumulator AC.

4.3.4.2 CALL AC, [SIXBIT/TIMER/] or CALLI AC, 22 - These return the time of day, in clock ticks (jiffies), right-justified in accumulator AC.

4.3.4.3 CALL AC, [SIXBIT/MSTIME/] or CALLI AC, 23 - These return the time of day, in milliseconds right-justified in accumulator AC.

4.3.4.4 CALL AC, [SIXBIT/RUNTIM/] or CALLI AC, 27 - The accumulated running time, in milliseconds, of the job whose number is in accumulator AC, is returned right-justified in accumulator AC. If the job number in AC is zero, the running time of the currently running job is returned. If the job whose number is in AC does not exist, zero is returned.

4.3.4.5 CALL AC, [SIXBIT/SLEEP/] or CALLI AC, 31 - These stop the job, and continue automatically after an elapsed real time of

$$[c(AC) \times \text{clock frequency}] \text{ modulo } 2^{12} \text{ jiffies.}$$

The contents of the AC are thus interpreted as the number of seconds the job wishes to sleep; however, there is an implied maximum of approximately 68 seconds or one minute.

4.3.5 Identification

4.3.5.1 CALL AC, [SIXBIT/PJOB/] or CALLI AC, 30 - These return the job number right-justified in accumulator AC.

4.3.5.2 CALL AC, [SIXBIT/GETPPN/] or CALLI AC, 24 - These return in AC the project-programmer pair of the job. The project number is a binary number in the left half of AC, and the programmer number is a binary number in the right half of AC. If the program being run is LOGIN or LOGOUT from the system device, the current project-programmer number is changed to 1,2 so that all files are accessible for reading and writing, and a skip return is given if the old project-programmer number is also logged in on another job.

4.3.5.3 CALL AC, [SIXBIT/LOGIN/] or CALLI AC, 15 - This programmed operator is intended for use with the LOGIN command only. Accumulator AC contains XWD -n, TABLE, where TABLE is the first location of n words to be stored in the Monitor's job tables for this user. The first table is project-programmer number (PRJPRG); the second is the job privilege bits (JBTPRV). If LH is less than -2, the extra words are ignored. If LH is -1, only the first table is set.

4.3.5.4 CALL AC, [SIXBIT/PEEK/] or CALLI AC, 33 - This UUO allows a user program to examine any location in the Monitor. Some customers may want to restrict the use of this UUO to project 1.

The call is:

```
        MOVEI AC, exec address          ;TAKEN MODULO 16K
        CALL AC, [SIXBIT/PEEK/]         ;OR CALLI AC, 33
```

This call returns with the contents of the Monitor location in AC. It is used by SYSTAT and could be used for on-line Monitor debugging.

4.3.5.5 CALL AC, [SIXBIT/GETLIN/] or CALLI AC, 34 - This UUO returns the SIXBIT physical name of the Teletype console that the program is attached to.

The call is:

```
        CALL AC, [SIXBIT/GETLIN/]       ;OR CALLI AC, 34
```

The name is returned left justified in the AC.

Example: CTY or TTY3 or TTY30

This UUO is used by the LOGIN program to print the TTY name.

4.3.6 Direct User I/O

The user I/O mode (bits 5 and 6 of PC word = 11) of the central processor allows running privileged user programs with automatic protection and relocation in effect. This mode provides some protection against partially debugged Monitor routines, and permits running infrequently used device service routines as a user job. Direct control by the user program of special devices is particularly important in realtime applications.

To utilize this mode, the job number must be 1. CALL [SIXBIT/RESET/] or CALLI 0 terminates user I/O mode.

4.3.6.1 CALL AC, [SIXBIT/TRPSET/] or CALLI AC, 25 - This UUO is a privileged UUO which temporarily stops time sharing and allows the user program to gain control of the interrupt locations. This UUO is temporary until some "knave-proof" realtime UUOs are implemented which will not stop time sharing and which cannot crash the system. If the user is not job 1, or if AC contains either zero or the left half is not in the range 40 through 57, control returns to the next location after the CALL. Otherwise, all other jobs are stopped and, if AC contains zero, the central processor is placed in user I/O mode and control returns to the second location following the CALL. If the left half of AC contains a number between 40 and 57 inclusive, the contents of the relative location specified in the right half of AC are fetched; the job relocation address is added to the address field, and the result is stored in the absolute location (40-57) specified in the left half of AC; the central processor is placed in the user I/O mode; and control is returned to the second location following the CALL. Thus, the user can set up a priority interrupt trap into his relocated core area.

The call is:

```
MOVE AC, XWD N,ADR
CALL AC, [SIXBIT/TRPSET/]
ERROR RETURN
NORMAL RETURN
```

The Monitor assumes that user location ADR contains either a JSR U or BLKI U, where U is a user address. Consequently, the Monitor will add the job's relocation to the contents of location U to make it an absolute IOWD. Therefore, a user should reset the contents of U before every TRPSET call.

4.3.6.2 UJEN (Op code 100) - This unimplemented op code dismisses a user I/O mode interrupt if one is in progress. If the interrupt is from user mode, a JRST 12, instruction can dismiss the interrupt. If the interrupt was from executive mode, however, this operator must be used to dismiss the interrupt. The program must restore all accumulators, and execute

```
UJEN U
```

where user location U contains the program counter as stored by a JSR instruction when the interrupt occurred.

4.3.6.3 CALL AC, [SIXBIT/SWITCH/] or CALLI AC, 20 - These return the contents of the central processor data switches in AC. Caution must be exercised in using the data switches since they are not an allocated device and are always available to all users.

4.4 INPUT/OUTPUT PROGRAMMING

All user input/output operations are controlled by the use of Monitor programmed operators. These are device independent, in the sense that if an operator is not pertinent to a given device, the operator is treated as a no-operation code. For example, a rewind directed to a line printer does nothing. Devices are referenced by logical names or physical names (see ASSIGN command, Chapter 2), and the characteristics of a device can be obtained from the Monitor. Properly used, these system characteristics permit the programmer to delay the device specification for his program from program-generation until program-run time. I/O is accomplished by associating a device, a file, and a ring buffer or command list with one of a user's I/O channels.

4.4.1 File

A file is an ordered set of data on a peripheral device. Its extent on input is determined by an end-of-file condition dependent on the device. For instance, a file is terminated by reading an end-of-file gap from magnetic tape, by an end-of-file card from a card reader, or by depressing the end-of-file switch on a card reader (see Chapter 5). The extent of a file on output is determined by the amount of information written by the OUT or OUTPUT programmed operators up through and including the next CLOSE or RELEASE operator.

4.4.1.1 Device - To specify a file, it is necessary to specify the device from which the file is to be read or onto which the file is to be written. This specification is made by an argument of the INIT or OPEN programmed operators. Devices are separated into two categories--those with no filename directory, and those with one or more filename directories.

a. Nondirectory Devices - For nondirectory devices, e.g., card reader, line printer, paper tape reader and punch, and user console, the only file specification required is the device name. All other file specifiers, if given, are ignored by the Monitor. Magnetic tape, which is also a nondirectory device, requires, in addition to the name, that the tape be properly positioned. Even though LOOKUP is not required to read and ENTER is not required to write, it is advisable to always use them so that a directory device may be substituted for a nondirectory device at run time (using the Monitor command, ASSIGN). Only in this way can user programs be truly device independent.

b. Directory Devices - For directory devices, e.g., DECtape and disk, files are addressable by name. If the device has a single file directory, e.g., DECtape, the device name and filename are sufficient information to determine a file. If the device has multiple file directories, e.g., disk, the name of the file directory must also be specified. These names are specified as arguments to the LOOKUP, ENTER, and RENAME programmed operators.

4.4.1.2 Data Modes - Data transmissions are either unbuffered (dump) or buffered. The mode of transmission is specified by a 4-bit argument to the INIT, OPEN, or SETSTS programmed operators. Table 4-3 summarizes the data modes.

Table 4-3
Data Modes

Octal Code Buffered Modes	Mnemonic	Meaning
0	A	ASCII. 7-bit characters packed left justified, five characters per word.
1	AL	ASCII line. Same as 0, except that the buffer is terminated by a FORM, VT (vertical tab), LINE-FEED or ALTMODE character.
2-7		Unused.
10	I	Image. A device dependent mode. The buffer is filled with data exactly as supplied by the device.
11-12		Unused.
13	IB	Image binary. 36-bit bytes. This mode is similar to binary mode, except that no automatic formatting or checksumming is done by the Monitor.
14	B	Binary. 36-bit byte. This is a blocked format consisting of a word count, n (the right half of the first data word of the buffer), followed by n 36-bit data words. Checksumming is done for cards and paper tape.
Unbuffered Modes		
15	ID	Image Dump. A device dependent dump mode.
16	DR	Dump respecting record boundaries. Data is transmitted between any contiguous block of core and one record on the device.
17	D	Dump. Same as mode 16, except that record boundaries are ignored on input and arranging the data into records is automatic on output.

a. Unbuffered Data Modes - Data modes 15, 16, and 17 utilize a command list to specify areas in the user's allocated core to be read or written. The effective address of the IN, INPUT, OUT, and OUTPUT programmed operators points to the first word of the command list. Three types of entries may occur in the command list.

(1) IOWD n, loc - Causes n words from loc through loc+n-1 to be transmitted. The next command is obtained from the next location following the IOWD. The assembler pseudo-op IOWD generates XWD -n, loc-1.

(2) XWD0, y - Causes the next command to be taken from location y. Referred to as a GOTO word.

(3) 0 - Terminates the command list.

The Monitor does not return program control to the user until the command list has been completely processed. If an illegal address is encountered while processing the list, an APR trap occurs if the user has enabled the central processor for "illegal memory" references; otherwise, the job is stopped and the Monitor prints

ADDRESS CHECK AT USER LOC addr

on the user's console, leaving the console in Monitor mode.

b. Buffered Data Modes - Data modes 0, 1, 10, 13, and 14 utilize a ring of buffers in the user area and the priority interrupt system to permit the user to overlap computation with his data transmission. Core memory in the user's area serves as an intermediate buffer between the user's program and the device. A ring of buffers consists of a 3-word header block for bookkeeping and a data storage area subdivided into one or more individual buffers linked together to form a ring. During input operations, the Monitor fills a buffer, makes the buffer available to the user's program, advances to the next buffer in the ring and fills it if it is free. The user's program follows along behind, emptying the next buffer if it is full, or waiting for the next buffer to fill. During output operations, the user's program and the Monitor exchange roles, the user filling the buffers and the Monitor emptying them.

(1) Buffer Structure - A ring of buffers consists of a 3-word header block and a data storage area subdivided into one or more individual buffers linked together to form a ring. The ring buffer layout is shown in Figure 4-1, and explained in the paragraphs which follow.

(a) Buffer Header Block - The location of the 3-word buffer header block is specified by an argument of the INIT and OPEN operators. Information is stored in the header by the Monitor in response to user execution of Monitor programmed operators. The user's program finds all the information required to fill and empty buffers in the header. Bit position 0 of the first word of the header is a flag which, if 1, means that no input or output has occurred for this ring of buffers. The right half of the first word is the address of the second word of the buffer currently in use by the user's program. The second word of the header contains a byte pointer to the current byte in the current buffer. The byte size is determined by the data mode. The third word of the header contains the number of bytes remaining in the buffer.

(b) Buffer Data Storage Area - The buffer data storage area is established by the INBUF and OUTBUF operators, or, if none exists when the first IN, INPUT, OUT, or OUTPUT operator is executed, a 2-buffer ring is set up. The effective address of the INBUF and OUTBUF operators specifies the number of buffers in the ring.

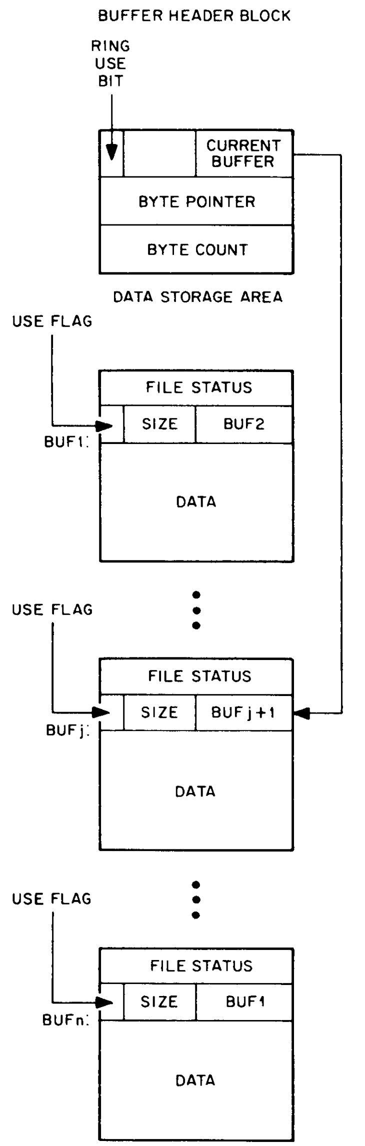


Figure 4-1 User's Ring of Buffers

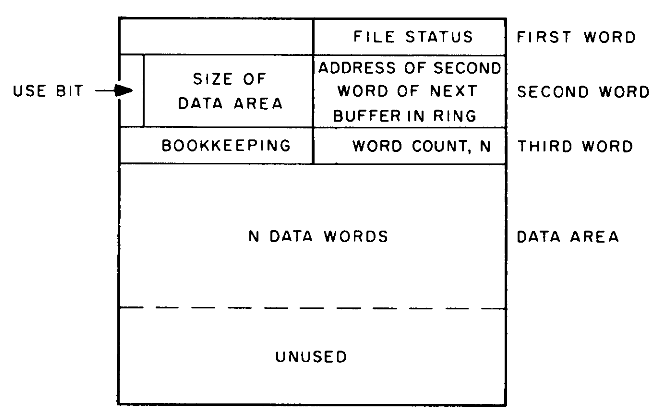


Figure 4-2 Detailed Diagram of Individual Buffer

The location of the buffer storage area is specified by the contents of the right half of JOBFF in the user's job data area. The Monitor updates JOBFF to point to the first location past the storage area.

All buffers in the ring are identical in structure. As Figure 4-2 shows, the right half of the first word contains the file status at the time that the Monitor advanced to the next buffer in the ring. Bit 0 of the second word of a buffer, called the use bit, is a flag that indicates whether the buffer contains active data. This bit is set to 1 by the Monitor when the buffer is full on input or being emptied on output, and set to 0 when the buffer is empty on output or is being filled on input. The use bit prevents the Monitor and the user's program from interfering with each other by attempting to use the same buffer simultaneously. Bits 1 through 17 of the second word of the buffer contain the size of the data area of the buffer which immediately follows the second word. The size of this data area depends on the device. The right half of the second word of the buffer contains the address of the second word of the next buffer in the ring.

The right half of the first word of the data area of the buffer, i.e., the third word of the buffer, is reserved for a count of the number of words (excluding itself) that actually contain data. The left half of this word is reserved for other bookkeeping purposes, depending on the particular device and the data mode.

4.4.1.3 File Status - The file status is a set of 18 bits (right half word), which reflects the current state of a file transmission. The initial status is a parameter of the INIT and OPEN operators. Thereafter, bits are set by the Monitor, and may be tested and reset by the user via Monitor programmed operators. Table 4-4 defines the file status bits. All bits, except the end-of-file bit, are set immediately by the Monitor as the conditions occur, rather than being associated with the buffer that the user is currently working on. However, the file status is stored with each buffer so that the user can determine which bufferful produced an error. A more thorough description of bits 18 through 29 is given in Chapter 5.

Table 4-4
File Status

Bit	Meaning
18	Improper mode, e.g., attempt to write on a write-locked tape.
19	Device detected error, other than hardware checksum or parity. Checksum, and/or parity error detected by hardware and/or software.
20	Data error, e.g., a computed checksum failed or invalid data was received.

Table 4-4 (Cont)
File Status

Bit	Meaning
21	Block too large. A block of data from a device is too large to fit in a buffer, or a block number is too large.
22	End of file.
23	Device is actively transmitting or receiving data.
24-29	Device dependent parameters. (See Chapter 5.)
30	Synchronous input. Stop the device after each buffer is filled.
31	Forces the Monitor to use the word count in the first data word of the buffer (output only). The Monitor normally computes the word count from the byte pointer in the buffer header.
32-35	Data mode. See Table 4-3.

4.4.2 Initialization

4.4.2.1 Job Initialization - The Monitor programmed operator

CALL [SIXBIT/RESET/] or CALLI 0

should be the first instruction in each program. It immediately stops all input/output transmissions on all devices without waiting for the devices to become inactive. All device allocations made by the INIT and OPEN operators are cleared, and, unless the devices have been assigned by the ASSIGN command (see Chapter 2), the devices are returned to the Monitor facilities pool. The content of the left half of JOBSA (program break) is stored in the right half of JOBFF so that the user buffer area is reclaimed if the program is starting over. The left half of JOBFF is cleared. Any files which have not been closed will be deleted on disk. Any older version having the same filename will remain.

4.4.2.2 Device Initialization

```
OPEN D,SPEC                 INIT D,STATUS
error return                SIXBIT/ldev/
normal return               XWD OBUF, IBUF
      .                     error return
      .                     normal return
SPEC:EXP STATUS
      SIXBIT/ldev/
      XWD OBUF,IBUF
```

The OPEN (operation code 050) and INIT (operation code 041) programmed operators initialize a file by specifying a device, ldev, and initial file status, STATUS, and the location of the input and output buffer headers.

a. Data Channel - OPEN and INIT establish a correspondence between the device, ldev, and a 4-bit data channel number, D. Most of the other input/output operators require this channel number as an argument. If a device is already assigned to channel D, it is released. (See RELEASE in this chapter.) The device name, ldev, is either a logical or physical name, with logical names taking precedence over physical names. (See ASSIGN command, Chapter 2.) If the device, ldev, is not the system device, SYS, and is allocated to another job or does not exist, the error return is taken. If the device is the system device, SYS, the job is stopped in a system device wait queue, and will continue running when SYS becomes available.

b. Initial File Status - The file status, including the data mode, is set to the value of the symbol STATUS. If the data mode is not legal (see Chapter 5) for the specified device, the job is stopped and the Monitor prints

ILL DEVICE DATA MODE FOR DEVICE dev AT USER addr,

where dev is the physical name of the device and addr is the location of the OPEN or INIT operator, on the user's console and leaves the console in Monitor mode.

c. Buffer Header - Symbols OBUF and IBUF, if nonzero, specify the location of the first word of the 3-word buffer header for output and input respectively. Only those headers which are to be used need to be specified. For instance, the output header need not be specified, if only input is to be done. Also, modes 15, 16, and 17 require no header. If either of the buffer headers of the 3-word block starting at location SPEC lies outside the user's allocated core area, an illegal memory violation occurs. If the user has enabled the central processor for illegal memory traps (see APRENB in this chapter), the trap occurs. Otherwise, the job is stopped and the Monitor prints

ADDRESS CHECK FOR DEVICE dev AT USER LOC addr

where addr is the address of the OPEN or INIT operator, on the user's console and leaves the console in Monitor mode.

The first and third words of the buffer header are set to zero. The left half of the second word is set up with the byte pointer size field in bits 6 through 11 for the selected device-data mode combination.

4.4.2.3 Buffer Initialization - Buffer data storage areas may be established by the INBUF and OUTBUF programmed operators, or by the first IN, INPUT, OUT, or OUTPUT operator, if none exists at that time, or the user may set up his own buffer data storage area.

a. Monitor Generated Buffers - Each device has associated with it a standard buffer size (see Chapter 5). The Monitor programmed operators INBUF D, n (operation code 064) and OUTBUF D, n (operation code 065) set up a ring of n standard size buffers associated with the input and output buffer headers, respectively, specified by the last OPEN or INIT operator on data channel D. If no OPEN or INIT operator has been performed on channel D, the Monitor stops the job and prints.

I/O TO UNASSIGNED CHANNEL AT USER LOC addr

where addr is the location of the INBUF or OUTBUF operator, on the user's console leaving the console in Monitor mode.

The storage space for the ring is taken from successive locations, beginning with the location specified in the right half of JOBFF. This is set to the program break, which is the first free location above the program area, by RESET. If there is insufficient space to set up the ring, an "illegal memory" violation occurs, which will cause a trap, if the user has enabled for it (see APRENB in this chapter), or the Monitor will stop the job and print

ADDRESS CHECK FOR DEVICE ldev AT USER LOC addr

where ldev is the physical name of the device associated with channel D and addr is the location of the INBUF or OUTBUF operator, on the user's console and leaves the console in Monitor mode.

The ring is set up by setting the second word of each buffer with a zero use bit, the appropriate data area size, and the link to the next buffer. The first word of the buffer header is set with a 1 in the ring use bit, and the right half contains the address of the second word of the first buffer.

b. User Generated Buffers - The following code illustrates an alternative to the use of the INBUF programmed operator. Analogous code may replace OUTBUF. This user code operates similarly to INBUF. SIZE must be set equal to the greatest number of data words expected in one physical record.

```
GO:         INIT 1, 0                       ;INITIALIZE ASCII MODE
            SIXBIT/MTA0/                    ;MAGNETIC TAPE UNIT 0
            XWD 0, MAGBUF                   ;INPUT ONLY
            JRST NOTAVL
            MOVE 0, [XWD 400000,BUF1+1]     ;THE 400000 IN THE LEFT HALF MEANS THE
                                            ;BUFFER WAS NEVER REFERENCED.
            MOVEM 0, MAGBUF
            MOVE 0, [POINT BYTSIZ,0,35]     ;SET UP NONSTANDARD BYTE SIZE
            MOVEM 0, MAGBUF+1
            JRST CONTIN                     ;GO BACK TO MAIN SEQUENCE
MAGBUF:     BLOCK 3                         ;SPACE FOR BUFFER HEADER
BUF1:       0                               ;BUFFER 1, 1ST WORD UNUSED
            XWD SIZE+1,BUF2+1               ;LEFT HALF CONTAINS BUFFER SIZE,
                                            ;RIGHT HALF HAS ADDRESS OF NEXT BUFFER
            BLOCK SIZE+1                    ;SPACE FOR DATA, 1ST WORD RECEIVES
                                            ;WORD-COUNT. THUS ONE MORE WORD
                                            ;IS RESERVED THAN IS REQUIRED
                                            ;FOR DATA ALONE
BUF2:       0                               ;SECOND BUFFER
            XWD SIZE+1,BUF3+1
            BLOCK SIZE+1
BUF3:       0                               ;THIRD BUFFER
            XWD SIZE +1,BUF1+1              ;RIGHT HALF CLOSES THE RING
            BLOCK SIZE+1
```

4.4.2.4 File Selection - The LOOKUP (operation code 076) and ENTER (operation code 077) programmed operators select a file for input and output respectively. Although these operators are not necessary for nondirectory devices, it is good programming practice to always use them so that directory devices may be substituted at run time. (See ASSIGN, Chapter 2.)

a. LOOKUP D, E

```
        error return
        normal return
        .
        .
        .
E:      SIXBIT/file/            ;filename, 1 to 6 characters.
        SIXBIT/ext/             ;filename extension, 0 to 3 characters.
        0
        XWD project number, programmer number,
```

LOOKUP selects a file for input on channel D. If no device is associated with channel D, 7 is stored in bits 33 through 35 of location E+1, and the error return is taken. If the input side of channel D is not closed (see CLOSE, in this chapter), it is now closed. The output side of channel D is not affected. If the device associated with channel D does not have a directory, the normal return is now taken. If the device has multiple directories, e.g., disk, the Monitor searches the master file directory of the device for the user's file directory whose number is in location E+3 and whose extension is UFD. If E+3 contains zero, the project-programmer pair of the current job is used as the name of the user's file directory. If this file is not found in the master file directory, 1 is stored in bits 33 through 35 of location E+1 and the error return is taken.

The user's file directory or the device directory in the case of a single-directory device (e.g., DECtape) is searched for the file whose name is in location E and whose extension is in the left half of location E+1. If the file is not found, 0 is stored in the right half of E+1 and the error return is taken. If the device is a multiple-directory device (e.g., disk) and the file is found, but is read protected (see File Protection in this chapter), 2 is stored in the right half of location E+1 and the error return is taken. Otherwise, location E+1 through E+3 are filled by the Monitor with the following data concerning the file, and the normal return is taken.

(1) The left half of location E+1 is set to the filename extension.

(2) If the device is a multiple-directory device, bits 24 through 35 of location E+1 are set to the date (in the format of DAYTIME programmed operator) that the file was last referenced.

If the device is a single-directory device, the right half of location E+1 is set to the device block number of the first block of the file.

(3) If the device is a multiple-directory device, bits 0 through 8 of location E+2 are set to the file protection. (See "File Protection," this chapter.)

(4) Bits 9 through 12 of location E+2 are set to the data mode in which the file was written.

(5) Bits 13 through 23 of location E+2 are set to the time, in minutes, and bits 24 through 35 of location E+2 are set to the date (in the format of the DAYTIME programmed operator) of the file's creation, i.e., of the last ENTER or RENAME programmed operator.

(6) If the device is a multiple-directory device, the left half of location E+3 is set to the negative of the number of words in the file, and the right half is unchanged . If the file contains more than 2^{17} words, then the left half contains the positive number of 128-word blocks in the file.

If the device is a single-directory device, location E+3 is used only for SAVed files (see Chapter 3), and contains the IOWD of the core image, i.e., the left half is the negative word length of the file and the right half is the core address of the first word minus 1.

b. ENTER D,E

```
    error return              ;filename, 1 through 6 characters.
    normal return             ;filename, extension, 0 through 3 characters.
    :
    :
E:  SIXBIT/file/
    SIXBIT/ext/
    EXP<TIME>B23+DATE
    XWD project number, programmer number.
```

ENTER selects a file for output on channel D. If no device is associated with channel D, 7 is stored in bits 33 through 35 of location E+1 and the error return is taken. If the output side of channel D is not closed (see CLOSE in this chapter), it is now closed. The input side of channel D is not affected. If the device does not have a directory, the normal return is now taken.

If the device has multiple directories, e.g., disk, the Monitor searches the master file directory of the device for the user's file directory whose name is in location E+3 and whose extension is UFD. If E+3 contains 0, the project-programmer pair of the current job is used as the name of the user's file directory. If this file is not found in the master file directory, 1 is stored in bits 33 through 35 of location E+1, and the error return is taken. If the filename in location E is 0, 0 is stored in bits 33 through 35 of location E+1, and the error return is taken. The user's file directory, or the device file directory in the case of a single-directory device, such as DECtape, is searched for the file whose name is in location E and whose extension is in the left half of location E+1.

If the device is a multiple-directory device and the file is found but is being written or renamed, 3 is stored in bits 33 through 35 of location E+1, and the error return is taken. If the file is write protected (See "File Protection", this chapter), 2 is stored in bits 33 through 35 of location E+1, and the error return is taken.

If the file is found, and is not being written or renamed and is not write protected, then the file is deleted, or marked for later deletion after all read references are completed, and the storage space on the device is recovered.

The Monitor then makes the file entry by recording the following information concerning the file and takes the normal return.

(1) The filename is taken from location E.

(2) The filename extension is taken from the left half of location E+1.

(3) If the device is a multiple-directory device, then

(a) the current date is taken as the date of last reference;

(b) the file protection key is set to 055 (see "File Protection," this chapter);

(c) the current data mode is taken as the mode in which the file is to be written;

(d) the project number of the current job is taken as the file owner's project number; and

(e) if bits 13 through 35 of location E+2 are nonzero, bits 13 through 23 are taken as the time of creation, in minutes, and bits 24 through 35 are taken as the date of creation (in the format of the DAYTIME programmed operator) of the file. Otherwise, the current time and date are used.

If the device is a single-directory device, then, if bits 24 through 35 of location E+2 are nonzero, they are taken as the date of creation; otherwise, the current date is used.

4.4.2.5 File Protection - File protection on nondirectory and single-directory devices is obtained by use of the ASSIGN command (see Chapter 2). Multiple-directory devices have a master file directory for the device which contains entries for each user's file directory. File selection (see LOOKUP and ENTER in this chapter) requires specification of the name of a user's file directory and a filename within that directory. This permits each user to access all files on the device, and necessitates a file protection scheme to prevent unauthorized references. For this purpose users are divided into three categories:

a. The file owner is the user whose project-programmer pair is the same as the filename of the user's file directory in which the file is entered.

b. Project members are users whose project number is the same as that of the file owner.

c. All other users.

There are three types of protection against each of the three categories of users.

a. Protection-protection - the protection cannot be altered

b. Read protection - the file may not be read.

c. Write protection - the file may not be rewritten, RENAMEd, or deleted.

The file protection key, shown in the following figure, is a set of nine bits which specify the three types of protection for each of the categories of users. (Also see Section 5.8.2.4, "Protection".)

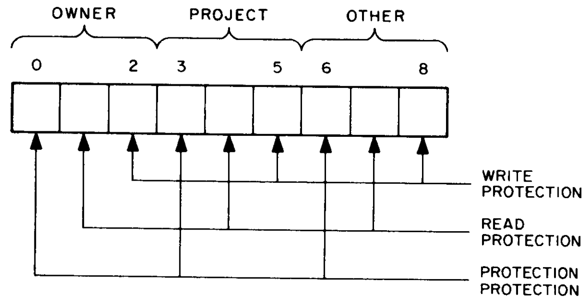


Figure 4-3 File Protection Key

When a file is created by an ENTER programmed operator, the file protection key is set to 055, indicating that the file is protection-protected and write-protected against all users except the owner. The protection key is returned by the LOOKUP D,E programmed operator in bits 0 through 8 of location E+2. It can be changed by the RENAME programmed operator. The owner's protection-protection and read-protection bits are ignored by the Monitor, thereby preventing a file from becoming inaccessible to everyone. However, the LOGIN CUSP sets the protection-protection bit if a user indicates he wishes to selectively protect his file for future logouts. This feature is handled completely by the LOGOUT CUSP.

a. RENAME D,E

error return
normal return
⋮

E: SIXBIT/file/ ;filename, 1 through 6 characters.
SIXBIT/ext/ ;filename extension, 0 through 3 characters.
EXP <PROT>B8+<TIME>B23+DATE
XWD project number, programmer number.

The RENAME programmed operator (operation code 055) is used to alter the filename, the filename extension, and the file protection key, or to delete a file associated with channel D on a directory device.

If no device is associated with channel D, 7 is stored in bits 33 through 35 of location E+1, and the error return is taken. If the device is a nondirectory device, the normal return is taken. If no file is currently selected on channel D, 5 is stored in bits 33 through 35 of location E+1, and the error return is taken.

If the device has multiple directories, e.g., disks, the Monitor searches the master file directory of the device for the user's file directory whose name is in location E+3 and whose extension is UFD. If E+3 contains 0, the project-programmer pair of the current job is used as the name of the user's file directory. If this file is not found in the master file directory, 1 is stored in bits 33 through 35 of location E+1, and the error return is taken. The user's file directory, or the device file directory in the case of a single-directory device, is searched for the file currently selected on channel D. If the file is not found, 0 is stored in bits 33 through 35 of location E+1, and the error return is taken.

If the device is a multiple-directory device and the file is found, but is being written or renamed, 3 is stored in bits 33 through 35 of location E+1, and the error return is taken. If the file is owner write-protected or if the protection key is being modified, i.e., bits 0 through 8 of location E+2 differ from the current protection key, and the file is owner protection-protected, 2 is stored in bits 33 through 35 of location E+1, and the error return is taken.

If the new filename in location E is 0, the file is deleted, or marked for deletion, after all read references are completed, and the normal return is taken. If the filename in location E and the filename extension in the left half of location E+1 are the same as the current filename and filename extension, respectively, the protection key is set to the contents of bits 0 through 8 of location E+2, and the normal return is taken.

If the new filename in location E and/or the filename extension in the left half of location E+1 differ from the current filename and/or filename extension, the user's file directory (or the device directory) is searched for the new filename and extension, as in LOOKUP. If a match is found, 4 is stored in bits 33 through 35 of location E+1, and the error return is taken. If no match is found, the file is changed to the new name in location E, the filename extension is changed to the new filename extension in the left half of location E+1, the protection key is set to the contents of bits 0 through 8 of location E+2, the access date is set to the current date, and the normal return is taken.

4.4.2.6 Examples

General Device Initialization

```
INIDEV:     0                       ;JSR HERE
            INIT 3, 14              ;BINARY MODE, CHANNEL 3
            SIXBIT/DTA5/            ;DEVICE DECTAPE UNIT 5
            XWD OBUF, IBUF          ;BOTH INPUT AND OUTPUT
            JRST NOTAVL             ;WHERE TO GO IF DTA5 IS BUSY
;FROM HERE DOWN IS OPTIONAL DEPENDING ON THE DEVICE AND PROGRAM
;REQUIREMENTS
            MOVE 0, JOBFF
            MOVEM 0, SV JBFF        ;SAVE THE FIRST ADDRESS OF THE BUFFER
                                    ;RING IN CASE THE SPACE MUST BE
                                    ;RECLAIMED.
```

```
            INBUF 3,4            ;SET UP 4 INPUT BUFFERS
            OUTBUF 3,1           ;SET UP 1 OUTPUT BUFFER
            LOOKUP 3, INNAM      ;INITIALIZE AN INPUT FILE
            JRST NOTFND          ;WHERE TO GO IF THE INPUT FILE NAME IS
                                 ;NOT IN THE DIRECTORY
            ENTER 3, OUTNAME     ;INITIALIZE AN OUTPUT FILE
            JRST NOROOM          ;WHERE TO GO IF THERE IS NO ROOM IN
                                 ;THE DIRECTORY FOR A NEW FILE NAME.
            JRST@INIDEV          ;RETURN TO MAIN SEQUENCE
OBUF:       BLOCK 3              ;SPACE FOR OUTPUT BUFFER HEADER
IBUF:       BLOCK 3              ;SPACE FOR INPUT BUFFER HEADER
INNAM:      SIXBIT/NAME/         ;FILE NAME
            SIXBIT/EXT/          ;FILE NAME EXTENSION (OPTIONALLY 0),
                                 ;RIGHT HALF WORD RECEIVES THE
                                 ;FIRST BLOCK NUMBER
            0                    ;RECEIVES THE DATE
            0                    ;UNUSED FOR NONDUMP I/O
OUTNAM:     SIXBIT/NAME/         ;SAME INFORMATION AS IN INNAME
            SIXBIT/EXT/
            0
            0
```

4.4.3 Data Transmission

The programmed operators

INPUT D,E and IN D,E
normal return
error return

transmit data from the file selected on channel D to the user's core area. The programmed operators

OUTPUT D,E and OUT D,E
normal return
error return

transmit data from the user's core area to the file selected on channel D.

If no OPEN or INIT operator has been performed on channel D, the Monitor stops the job and prints

I/O TO UNASSIGNED CHANNEL AT USER LOC addr

where addr is the location of the IN, INPUT, OUT, or OUTPUT programmed operator, on the user's console leaving the console in Monitor mode. If the device is a multiple-directory device and no file is selected on channel D, bit 18 of the file status is set to 1, and control returns to the user's program. Control always returns to the location immediately following an INPUT (operation code 066) and an OUTPUT (operation code 067). A check of the file status for end-of-file and error conditions must then be made by another programmed operator. Control returns to the location immediately following an IN (operation code 056) and an OUT (operation code 057), if no end-of-file or error condition exists, i.e., if bits 18 through 22 of the file status are all 0. Otherwise, control returns to the second location following the IN or OUT. Note that IN and OUT UUOs are the only ones in which the error return is a skip and the normal return is not a skip.

4.4.3.1 Unbuffered (Dump) Modes - In data modes 15, 16, and 17, the effective address E of the INPUT, IN, OUTPUT, and OUT programmed operators is the address of the first word of a command list (see Section 4.4.1). Control does not return to the program until the transmission is terminated or an error is detected.

Example -

Dump Output

Dump input is similar to dump output. This routine outputs fixed-length records.

```
DMPINI:     0                         ;JSR HERE TO INITIALIZE A FILE
            INIT 0, 16                ;CHANNEL 0, DUMP MODE
            SIXBIT/MTA2/              ;MAGNETIC TAPE UNIT 2
            0                         ;NO RING BUFFERS
            JRST NOTAVL               ;WHERE TO GO IF UNIT 2 IS BUSY
            JRST@DMPINI               ;RETURN
DMPOUT:     0                         ;JSR HERE TO OUTPUT THE OUTPUT AREA
            OUTPUT 0,OUTLST           ;SPECIFIES DUMP OUTPUT ACCORDING
                                      ;TO THE LIST AT OUTLIST
            STATZ 0, 740000           ;CHECK ERROR BITS
            CALL [SIXBIT/EXIT/]       ;QUIT IF AN ERROR OCCURS
            JRST @DMPOUT              ;RETURN
DMPDON:     0                         ;JSR HERE TO WRITE AN END OF FILE
            CLOSE 0,                  ;WRITE THE END OF FILE
            STATZ 0, 740000           ;CHECK FOR ERROR DURING WRITE
                                      ;END OF FILE OPERATION
            CALL [SIXBIT/EXIT/]       ;QUIT IF ERROR OCCURS
            RELEAS 0,                 ;RELINQUISH THE DEVICE
            JRST@DMPDON               ;RETURN
OUTLST:     IOWD BUFSIZ,BUFFER        ;SPECIFIES DUMPING A NUMBER OF
                                      ;WORDS EQUAL TO BUFSIZ, STARTING
                                      ;AT LOCATION BUFFER
            0                         ;SPECIFIES THE END OF THE COMMAND
                                      ;LIST
BUFFER:     BLOCK BUFSIZ              ;OUTPUT BUFFER, MUST BE CLEARED
                                      ;AND FILLED BY THE MAIN PROGRAM
```

4.4.3.2 Buffered Modes - In data modes 0, 1, 10, 13, and 14 the effective address E of the INPUT, IN, OUTPUT, and OUT programmed operators may be used to alter the normal sequence of buffer reference. If E is 0, the address of the next buffer is obtained from the right half of the second word of the current buffer. If E is nonzero, it is the address of the second word of the next buffer to be referenced. The buffer pointed to by E can be in an entirely separate ring from the present buffer. Once a new buffer location is established, the following buffers are taken from the ring started at E.

a. Input - If no input buffer ring is established when the first INPUT or IN is executed, a 2-buffer ring is set up. (See INBUF, Section 4.4.2.3.)

Buffered input may be performed synchronously or asynchronously at the option of the user. If bit 30 of the file status is 1, each INPUT and IN programmed operator

(1) Clears the use bit in the second word of the buffer whose address is in the right half of the first word of the buffer header, thereby making it available for refilling by the Monitor;

(2) Advances to the next buffer by moving the contents of the second word of the current buffer to the right half of the first word of the 3-word buffer header;

(3) If an end-of-file or an error condition exists, control is returned to the user's program. Otherwise, the Monitor starts the device which fills the buffer and stops transmission;

(4) Computes the number of bytes in the buffer from the number of words in the buffer (right half of the first data word of the buffer) and the data mode, and stores the result in the third word of the buffer header;

(5) Sets the position and address fields of the byte pointer in the second word of the buffer header, so that the first data byte is obtained by an ILDB instruction; and

(6) Returns control to the user's program.

Thus, in synchronous mode, the position of a device, such as magnetic tape, relative to the current data is easily determined. The asynchronous input mode differs in that once a device is started, successive buffers in the ring are filled at the interrupt level without stopping transmission until a buffer whose use bit is 1 is encountered. Control returns to the user's program after the first buffer is filled. The position of the device relative to the data currently being processed by the user's program depends on the number of buffers in the ring and when the device was last stopped.

Example -

General Subroutine to Input One Character

```
GETCHR:     0                         ;JSR HERE AND STORE PC
GETCNT:     SOSG   IBUF+2             ;DECREMENT THE BYTE COUNT
            JRST   GETBUF             ;BUFFER IS EMPTY (OR FIRST CALL AFTER
                                      ;INIT

GETNXT:     ILDB AC, IBUF+1           ;GET NEXT CHAR FROM BUFFER
            JMPN AC, @GETCHR          ;RETURN TO CALLER IF NOT NULL CHAR[1]
            JRST   GETCNT             ;IGNORE NULL AND GET NEXT CHAR
```

[1]For some devices in ASCII mode, the item count provided will always be a multiple of five characters. Since the last word of a buffer may be partially full, user programs which rely upon the item count should always ignore null characters.

```
GETBUF:     IN      3,              ;CALL MONITOR TO REFILL THIS BUFFER
            JRST    GETNXT          ;RETURN HERE WHEN NEXT BUFFER IS
                                    ;FULL (PROBABLY IMMEDIATELY)
            JRST    ENDTST          ;RETURN HERE ONLY IF ERROR OR EOF

ENDTST:     STATZ 3, 740000         ;CHECK FOUR ERROR BITS FIRST
            JRST    INERR           ;WHERE TO GO ON AN ERROR
            JRST    ENDFIL          ;WHERE TO GO ON AN END OF FILE
```

b. Output - If no output buffer ring has been established, i.e., if the first word of the buffer header is 0, when the first OUT or OUTPUT is executed, a 2-buffer ring is set up (see OUTBUF, this chapter). If the ring use bit (bit 0 of the first word of the buffer header) is 1, it is set to 0, the current buffer is cleared to all 0s, and the position and address fields of the buffer byte pointer (the second word of the buffer header) are set so that the first byte is properly stored by an IDPB instruction. The byte count (the third word of the buffer header) is set to the maximum of bytes that may be stored in the buffer, and control is returned to the user's program. Thus, the first OUT or OUTPUT initializes the buffer header and the first buffer, but does not result in data transmission.

If the ring use bit is 0 and bit 31 of the file status is 0, the number of words in the buffer is computed from the address field of the buffer byte pointer (the second word of the buffer header) and the buffer pointer (the first word of the buffer header), and the result is stored in the right half of the first data word of the buffer. If bit 31 of the file status is 1, it is assumed that the user has already set the word count in the right half of the first data word. The buffer use bit (bit 0 of the second word of the buffer) is set to 1, indicating that the buffer contains data to be transmitted to the device. If the device is not currently active, i.e., not receiving data, it is started. The buffer header is advanced to the next buffer by setting the buffer pointer in the first word of the buffer header. If the buffer use bit of the new buffer is 1, the job is put into a wait state until the buffer is emptied at the interrupt level. The buffer is then cleared to all 0s, the buffer byte pointer and byte count are initialized in the buffer header, and control is returned to the user's program.

Example -

<u>General Subroutine to Output One Character</u>

```
PUTCHR:     0                       ;JSR HERE AND STORE PC
            SOSG    OBUF+2          ;INCREMENT BYTE COUNT
            JRST    PUTBUF          ;NO MORE ROOM (OR FIRST CALL AFTER INIT)

PUTNXT:     IDPB AC, OBUF+1         ;STORE THIS CHARACTER
            JRST    @PUTCHR         ;AND RETURN TO CALLER

PUTBUF:     OUT     3,              ;CALL MONITOR TO EMPTY THIS BUFFER
            JRST    PUTNXT          ;RETURN HERE WHEN NEXT BUFFER IS
                                    ;EMPTY (PROBABLY IMMEDIATELY)
            JRST    OUTERR          ;RETURN HERE ONLY IF OUTPUT ERROR

OUTERR:     GETSTS  3,AC            ;GET THE ERROR STATUS TO LOOK AT
              .
              .
              .
```

4.4.4 Status Checking and Setting

The file status (see Table 4-1) is manipulated by the GETSTS (operation code 062), STATZ (operation code 063), STATO (operation code 061) and SETSTS (operation code 060) programmed operators. In each case the accumulator field of the instruction selects a data channel. If no device is associated with the specified data channel, the Monitor stops the job and prints,

I/O TO UNASSIGNED CHANNEL AT USER LOC addr

where addr is the location of the GETSTS, STATZ, STATO, or SETSTS programmed operator, on the user's console leaving the console in Monitor mode.

GETSTS D,E stores the file status of data channel D in the right half and 0 in the left half of location E.

STATZ D,E skips, if all file status bits selected by the effective address E are 0.

STATO D,E skips, if any file status bit selected by the effective address E is 1.

SETSTS D,E waits until the device on channel D stops transmitting data and replaces the current file status, except bit 23, with the effective address E. If the new data mode, indicated in the right four bits of E, is not legal for the device, the job is stopped and the Monitor prints

ILL DEVICE DATA MODE FOR DEVICE dev AT USER addr

where dev is the physical name of the device and addr is the location of the SETSTS operator, leaving the console in Monitor mode. If the data mode is changed by SETSTS, the byte pointers in the buffer headers are changed appropriately.

4.4.5 Terminating A File (CLOSE)

File transmission is terminated by the CLOSE D,N (operation code 070) programmed operator. If no device is associated with channel D or if bits 34 and 35 of the instruction are both 1, control returns to the user's program immediately.

If bit 34 is 0 and the input side of data channel D is open, it is now closed. In data modes 15, 16, and 17, the effect is to execute a device dependent function and clear the end-of-file flag, bit 22 of the file status. Data modes 0, 1, 10, 13, and 14 have the additional effect, if an input buffer ring exists, of setting the ring use bit (bit 0 of the first word of the buffer header) to 1, setting the buffer byte count (the third word of the buffer header) to 0 and setting the buffer use bit (bit 0 of the second word of the buffer) of each buffer to 0.

If bit 35 of the instruction is 0 and the output side of channel D is open, it is now closed. In data modes 15, 16, and 17, the effect is to execute a device dependent function. In data modes 0, 1, 10, 13, and 14, if a buffer ring exists, all buffers that have not yet been transmitted to the device are now written, device dependent functions performed, the ring use bit is set to 1, the buffer byte count is set to 0, and control returns to the user after transmission is complete.

Example:

Terminating A File

```
DROPDV:  0                    ;JSR HERE
         CLOSE 3,             ;WRITE END OF FILE AND TERMINATE
                              ;INPUT
         STATZ 3, 740000      ;RECHECK FINAL ERROR BITS
         JRST OUTERR          ;ERROR DURING CLOSE
         RELEAS 3,            ;RELINQUISH THE USE OF THE
                              ;DEVICE, WRITE OUT THE DIRECTORY
         MOVE 0, SVJBFF
         MOVEM 0, JOBFF       ;RECLAIM THE BUFFER SPACE
         JRST @ DROPDV        ;RETURN TO MAIN SEQUENCE
```

4.4.6 Synchronization of Buffered I/O (CALL D, [SIXBIT/WAIT/]

In some instances, such as recovery from transmission errors, it is desirable to delay until a device completes its input/output activities. The programmed operators,

CALL D, [SIXBIT/WAIT/] and CALLI D,10

return control to the user's program when all data transfers on channel D have finished. This UUO does not wait for a Magtape spacing operation, since no data transfer is in progress. An MTAPE D, 0 (see Section 5.7.2) should be used to wait for spacing and I/O activity to finish on Magtape. If no device is associated with data channel D, control returns immediately. After the device is stopped, the position of the device relative to the data currently being processed by the user's program can be determined by the buffer use bits.

4.4.7 Relinquishing A Device (RELEASE)

When all transmission between the user's program and a device is finished, the program must relinquish the device by performing a

RELEASE D,

RELEASE (operation code 071) returns control immediately, if no device is associated with data channel D. Otherwise, both input and output sides of data channel D are CLOSEd and the correspondence between channel D and the device, which was established by the INIT or OPEN programmed operators, is terminated. If the device is neither associated with another data channel nor assigned (see ASSIGN, Chapter 2) by command, it is returned to the Monitor's pool of available facilities. Control is returned to the user's program.

4.5 CORE CONTROL (CALL AC, [SIXBIT/CORE/])

CALL AC, [SIXBIT/CORE/]	CALLI AC,11
error return	error return
normal return	normal return

These programmed operators provide a user program with the ability to expand and contract its core size as its memory requirements change. Accumulator AC should contain the desired highest relative address. The Monitor will set JOBREL to this new value before returning to the user, provided that the request can be satisfied. If AC contains 0, the number of free 1024-word blocks is returned right-justified in AC, and the error return is taken. If core is being increased, the error return is taken, and the current allocation remains in effect if the request cannot be satisfied. Otherwise, core is appended to or removed from the top of the user's current core area, and the normal return is taken. In all cases the number of free 1024-word blocks is returned right-justified in AC.

CHAPTER 5

DEVICE DEPENDENT FUNCTIONS

This chapter explains the unique features of each standard I/O device.[1] All devices accept the programmed operators explained in Chapter 4 unless otherwise indicated. Buffer sizes are given in octal and include two bookkeeping words. Table 5-1 is a summary of the characteristics of all devices.

Table 5-1
Device Summary

Physical Name	Name	Hardware Type Number	Prog. Op.	Data Modes	Buffer[2] Size (octal)
CTY	Console Teletype	626 Models 33,35, 37	INPUT, IN OUTPUT, OUT	A, AL	23
TTY1, TTY2, ..., TTY77	Teletype	630, 680, or DC10	INPUT, IN OUTPUT, OUT	A, AL	25
PTR	Paper Tape Reader	760	INPUT, IN	A, AL, IB, B, I	43
PTP	Paper Tape Punch	761	OUTPUT, OUT	A, AL, IB, B, I	43
LPT	Line Printer	646, LP10	OUTPUT, OUT	A, AL	34
CDR	Card Reader	461, CR10	INPUT, IN	A, AL, B, I	36
DTA1, DTA2, ...,DTA0,	DECtape	551/555, TD10/TD55	INPUT, IN OUTPUT, OUT LOOKUP ENTER USETO USETI UGETF CALL [SIXBIT/UTPCLR/]	A, AL, IB, B, I, DR, D	202
MTA0, MTA1, ..., MTA7	Magnetic Tape	516, TM10, TU20, TU79	INPUT, IN OUTPUT, OUT MTAPE	A, AL, IB, B, I, DR, D	203
DSK	Disk		INPUT, IN OUTPUT, OUT LOOKUP ENTER RENAME USETO	A, AL, I, IB, B, DR, D	203
DIS	Display	30, 340	INPUT OUTPUT	ID	Dump only

[1]The user may determine the physical characteristics associated with a logical device name by executing a DEVCHR UUO. See 5.11.

[2]Buffer sizes are subject to change and should be calculated rather than assumed by user programs. A dummy INBUF or OUTBUF may be employed for this purpose.

5.1 TELETYPE

Device Name - TTY0, TTY1, ..., TTY76, TTY77, CTY

Line number n of the Type 630 Data Communications System, Data Line Scanner DC10, or PDP-8 680 System is referred to as TTYn. The console Teletype is CTY. The Time-Sharing Monitor automatically gives the logical name, TTY, to the user's console whenever a job is initialized.

Teletype device names are assigned dynamically. For interconsole communication by program, it is necessary for one of the two users to type DEASSIGN TTY in order to make his Teletype available to the other user's program as an output or input device. Typing ASSIGN TTYn is the only way to reassign a Teletype that has been deassigned. Also see TALK command, Section 2.4.6.

Buffer Size - 25_8 words.

5.1.1 Data Modes

5.1.1.1 A (ASCII) - All characters typed in appear in the input buffer as typed, with the following exceptions:

RUBOUT	Erases the previous character. Successive RUBOUTs erase characters to the left until the beginning of the current bufferful. For each character erased, a back-slash is typed. If there is no character to erase, a carriage-return/line-feed is performed.
RETURN (carriage-return)	Followed automatically with line-feed, both of which appear in the input buffer; however, the input buffer is not terminated. See "AL (ASCII Line)."
↑U (CTRL U)	Types back as ↑U followed by a carriage-return/line-feed. This character deletes the entire current input bufferful of ASCII characters.
↑O (CTRL O)	Similar to ↑U but has special action during output; it deletes all output until next input or INIT on TTY.
↑P (CTRL P)	Does not appear in the input buffer. Some Teletype units (usually Models 35 and 37) have horizontal tab, vertical tab, and form feed mechanisms while other units (usually Model 33s) do not. The Monitor assumes that all Teletype units in the system either do or do not have these mechanisms depending upon how the system was built (System Builder). If the user finds that his particular Teletype unit is different from the Monitor's assumption, he should type ↑P. Otherwise, tabs will not be printed at all or spaces will be substituted for a tab depending upon the Monitor's assumption. Alternate uses of ↑P simulate hardware tabs with multiple spaces on and off.

↑Z (CTRL Z)	Types as ↑Z and appears in the buffer as 032. This character serves as an end-of-file and terminates the input buffer.
↑B	Suppresses echoing of characters typed in. The user must type this character in the rare case that he is using a self echoing full-duplex Teletype unit.
TAB	Echoes as an appropriate number of spaces to place tab "stops" at every eighth space unless ↑P is typed.
VT	(vertical tab) types as four line-feeds.
FORM	(form-feed) types as eight line-feeds.
↑C (CTRL C)	Types as ↑C followed by carriage-return/line-feed. This character places the console in the Monitor mode, ready to accept Monitor commands.

On output, all characters are typed just as they appear in the output buffer with the exceptions, TAB, VT, and FORM, which are processed the same as on type in.

If, during output operations, an echo-check failure occurs (the transmitted character was not the same as the intended character), the I/O routine suspends output until the user types the next character. If that character is ↑C, the console is placed in Monitor mode. If it is ↑O, all Teletype output buffers that are currently full are ignored, thus cutting the output short. All other characters cause the service routines to continue output. The user may cause a deliberate echo check by typing in while typeout is in progress. For example, to return to Monitor control mode while typeout is in progress, the user must type any character ("X", for example) until an echo check occurs and output is suspended; then and only then he types ↑C.

The buffer is terminated when it fills up or when the user types ↑Z.

5.1.1.2 AL (ASCII Line) - Same as ASCII mode (usually preferred) with the addition that the input buffer is terminated by a CR/LF pair, FF, VT, or ALTMODE.

5.1.2 DDT Submode

To allow a user's program and the DDT debugging program to use the same Teletype without interfering with one another, the Teletype service routine provides the DDT submode. This mode does not affect the Teletype status if it is initialized with the INIT operator. It is not necessary to use INIT in order to do I/O in the DDT submode. I/O in DDT mode is always to the user's Teletype and not to any other device.

In the DDT submode, the user's program is responsible for its own buffering. Input is usually one character at a time, but if the typist types characters faster than they are processed, the Teletype service routine supplies bufferfuls of characters at a time.

To input characters in DDT mode, use the sequence

```
MOVEI AC,BUF
CALL AC, [SIXBIT/DDTIN/]
```

BUF is the first address of a 21-word block in the user's area. The DDTIN operator delays, if necessary, until one character is typed in. Then all characters (in 7-bit packed format) typed in since the previous occurrence of DDTIN are moved to the user's area in locations BUF, BUF+1, etc. The character string is always terminated by a null character (000). RUBOUTs are not processed by the service routine but are passed on to the user. The special control characters ↑O and ↑U have no effect. Other characters are processed as in ASCII mode.

To perform output in DDT mode, use the sequence

```
MOVEI AC,BUF
CALL AC, [SIXBIT/DDTOUT/]
```

BUF is the first address of a string of packed 7-bit characters terminated by a null (000) character. The Teletype service routine delays until the previous DDTOUT operation is complete, then moves the entire character string into the Monitor, begins to output the string, and restarts the user's program. Character processing is the same as for ASCII mode output.

5.2 PAPER TAPE READER

Device Mnemonic - PTR

Buffer Size - 43_8 words

5.2.1 Data Modes (Input Only)

> NOTE: To initialize the paper tape reader, the input tape must be threaded through the reading mechanism and the FEED button depressed.

5.2.1.1 A (ASCII) - Blank tape (000), RUBOUT (377), and null characters (200) are ignored. All other characters are truncated to seven bits and appear in the buffer. The physical end of the paper tape serves as an end-of-file and results in the character 032 ↑Z) appearing in the buffer.

5.2.1.2 AL (ASCII Line) - Character processing is the same as for the A mode. The buffer is terminated by LINE FEED, FORM, or VT.

5.2.1.3 I (Image) - There is no character processing. The buffer is packed with 8-bit characters exactly as read from the input tape. Physical end of tape is the end-of-file indication but does not cause a character to appear in the buffer.

5.2.1.4 IB (Image Binary) - Characters not having the eighth hole punched are ignored. Characters are truncated to six bits and packed six to the word without further processing. This mode is useful for reading binary tapes having arbitrary blocking format.

5.2.1.5 B (Binary) - Checksummed binary data is read in the following format. The right half of the first word of each physical block contains the number of data words that follow and the left contains half a folded checksum. The checksum is formed by adding the data words using 2s complement arithmetic, then splitting the sum into three 12-bit bytes and adding these using 1s complement arithmetic to form a 12-bit checksum. The data error status flag (IODERR) is raised if the checksum miscompares. Because the checksum and word count appear in the input buffer, the maximum block length is 40. The byte pointer, however, is initialized so as not to pick up the word count and checksum word.

Again, physical end of tape is the end-of-file indication but does not result in putting a character in the buffer.

5.3 PAPER TAPE PUNCH

Device Mnemonic - PTP

Buffer Size - 43_8 words

5.3.1 Data Modes

5.3.1.1 A (ASCII) - The eighth hole is punched for all characters. Tape-feed without the eighth hole (000) is inserted after form-feed. A rubout is inserted after each vertical or horizontal tab. Null characters (000) appearing in the buffer are not punched.

5.3.1.2 AL (ASCII Line) - The same as A mode. Format control must be performed by the user's program.

5.3.1.3 I (Image) - Eight-bit characters are punched exactly as they appear in the buffer with no additional processing.

5.3.1.4 IB (Image Binary) - Binary words taken from the output buffer are split into six 6-bit bytes and punched with the eighth hole punched in each line. There is no format control or checksumming performed by the I/O routine. Data punched in this mode is read back by the paper tape reader in the IB mode.

5.3.1.5 B (Binary) - Each bufferful of data is punched as one checksummed binary block as described for the paper tape reader. Several blank lines are punched after each bufferful for visual clarity.

5.3.2 Special Programmed Operator Service

The first output programmed operator of a file causes about two fanfolds of blank tape to be punched as leader. Following a CLOSE, an additional fanfold of blank tape is punched as trailer. No end-of-file character is punched automatically.

5.4 LINE PRINTER

Device Mnemonic - LPT

Buffer Size - 34_8 words

5.4.1 Data Modes

5.4.1.1 A (ASCII) - ASCII characters are transmitted to the line printer exactly as they appear in the buffer. See the PDP-10 System Reference Manual, for a list of the vertical spacing characters.

5.4.1.2 AL (ASCII Line) - This mode is exactly the same as A and is included for programming convenience. All format control must be performed by the user's program; this includes placing a RETURN, LINE-FEED sequence at the end of each line.

5.4.2 Special Programmed Operator Service

The first output programmed operator of a file and the CLOSE at the end of a file cause an extra form-feed to be printed to keep files separated.

5.5 CARD READER

Device Mnemonic - CDR

Buffer Size - 36_8 words

5.5.1 Data Modes

5.5.1.1 A (ASCII) - All 80 columns of each card are read and translated to 7-bit ASCII code. Blank columns are translated to spaces. At the end of each card a carriage-return/line-feed is appended. A card with the character 12-11-0-1 punched in column 1 is an end-of-file card. Columns 2 through 80 are ignored, and an end-of-file character 032 appears as the last character in the input buffer. The end-of-file button on the card reader has the same effect as the end-of-file card. As many complete cards as can fit are placed in the input buffer, but cards are not split between two buffers. Using the standard-sized buffer, only one card is placed in each buffer. The left arrow character 137 appears in each column containing an invalid punch.

5.5.1.2 AL (ASCII Line) - Exactly the same as the A mode.

5.5.1.3 I (Image) - All 12 punches in all 80 columns are packed into the buffer as 12-bit bytes. The first 12-bit byte is column 1. The last word of the buffer contains columns 79 and 80 as the left and middle bytes respectively. The end-of-file card and the end-of-file button are processed the same as in the A mode with the character 0032 appearing in the buffer as the last character of the file. Cards are not split between two buffers.

5.5.1.4 B (Binary) - Card column 1 must contain a 7-9 punch to verify that the card is in binary format. The absence of the 7-9 punch results in raising the IOIMPM (improper mode) flag in the card reader status word. Card column 2 must contain a 12-bit checksum as described for the paper tape reader binary format. Columns 3 through 80 contain binary data, 3 columns per word for 26 words. Cards are not split between two buffers. The end-of-file card and the end-of-file button are processed the same as in the A mode with a word containing 003200000000 appearing as the last word in the file.

5.6 DECTAPE

Device Mnemonic - DTA0, DTA1, ..., DTA7

Buffer Size - 202_8 words

5.6.1 Data Modes

5.6.1.1 A (ASCII) - Data is written on DECtape exactly as it appears in the buffer. No processing or checksumming of any kind is performed by the service routine. The self-checking of the DECtape system is sufficient assurance that the data is correct. See the description of DECtape format below for further information concerning blocking of information.

5.6.1.2 AL (ASCII Line) - Same as A.

5.6.1.3 I (Image) - Same as A. Data consists of 36-bit words.

5.6.1.4 IB (Image Binary) - Same as I.

5.6.1.5 B (Binary) - Same as I.

5.6.1.6 DR (Dump Records) - This mode is accepted but actually functions as dump mode 17.

5.6.1.7 D (Dump) - Data is read into or written from anywhere in the user's core area without regard to the standard buffering scheme. Control for read or write operations must be via a command list in core memory. The command list format is as described in Chapter 4, "Unbuffered (Dump) Modes;" any positive number appearing in a command list terminates the list. Dump data is automatically blocked into standard-length DECtape blocks by the DECtape control. Unless the number of data words is an exact multiple of the standard length of a DECtape block (128_{10}), after each output programmed operator, the remainder of the last block written is wasted. The input programmed operator must specify the same number of words that the corresponding output programmed operator specified in order to skip over the wasted fractions of blocks.

5.6.2 DECtape Block Format

A standard reel of DECtape consists of 578 (1102_8) prerecorded blocks each capable of storing 128 (200_8) 36-bit words of data. Block numbers which label the blocks for addressing purposes are recorded between blocks. These block numbers run from 0 to 1101_8. Blocks 0, 1, and 2 are normally not used during time-sharing and are reserved for a bootstrap loader. Block 100_{10} (144_8) is the directory block which contains the names of all files on the tape and information relating to each file. Blocks 1_{10} through 99_{10} (1-143_8) and 101_{10} through 577_{10} (145-1101_8) are usable for data.

If in the process of DECtape I/O, the I/O service routine is requested to use a block number larger than 1101_8 or smaller than 0, the Monitor sets the Block Too Large flag (bit 21) in the file status and returns.

5.6.3 DECtape Directory Format

The directory block (block 100_{10}) of a DECtape contains directory information for all files on that tape; a maximum of 22 files can be stored on any one DECtape.

Words 0 through 82_{10}	The first 83 words of the directory contain "slots," each "slot" representing one of the 577 (blocks 1 through 1101_8 are represented in these 83 words) blocks on the DECtape. Each slot occupies five bits (seven slots are stored per word) and contains the number of the file ($1-26_8$) to which the block the slot represents is assigned.
Words 83 through 104_{10}	The next 22 words contain the filenames of the 22 files residing on the DECtape. Word 83 contains the filename for file #1, word 84 the filename for file #2, etc. Filenames are stored in 6-bit code.
Words 105 through 126_{10}	The next 22 words contain the extension names and dates of the 22 files, in the same relative order as their filenames above.

Bits 0 through 17_{10}	The extension name of the file (in 6-bit code)
Bits 18 through 23_{10}	Number of 1K blocks minus 1 needed to load the file (maximum value=53). This information is stored for SAVEd files only.
Bits 24 through 35_{10}	The date the file was last updated, according to the formula: ((year-1964)*12+(month-1))*31+day-1

Word 127_{10}	Unused.

The message

BAD DIRECTORY FOR DEVICE DTAn: EXEC CALLED FROM USER LOC n

is produced whenever any of the following conditions are detected.

a. A parity error while reading the directory block.

b. No "slots" are assigned to the file number of the file.

c. The tape block which may possibly be the first block of the file (i.e., the first block for the file encountered while searching backwards from the directory block) cannot be read.

5.6.4 DECtape File Format

A file consists of any number of DECtape blocks. Each block contains:

Word 0	Left half	The link. The link is the block number of the next block in the file. If the link is zero, this block is the last in the file.
	Right half	Bits 18 through 27: The block number of the first block of the file.
		Bits 28 through 35: A count of the number of words in this block which are used (maximum 177_8)

Words 1 through 177_8	Data packed exactly as the user placed in his buffer, or in Dump Mode files, the next 127 words of memory.[1]

5.6.5 Special Programmed Operators Service

Several programmed operators are provided for manipulating DECtape. These allow the user to manipulate block numbers and to handle directories.

In addition to the operators above, INPUT, OUTPUT, CLOSE, and RELEAS have special effects. When performing nondump input operations, the DECtape service routine reads the links in each block to determine the next block to read and when to raise the end-of-file flag.

When an OUTPUT is given, the DECtape service routine examines the left half of the first data word in the output buffer (the word containing the word count in the right half). If this half word contains -1, it is replaced with a 0 before being written out, and the file is thus terminated. If this half word is greater than -1, it is not changed and the service routine uses it as the block number for the next OUTPUT.

Table 5-2
DECtape Programmed Operators

Programmed Operator	Effect
USETI D, E	Sets the DECtape on device channel D to input block E next. Input operations on this DECtape must not be active because otherwise the user has no way of determining which buffer contains block E.
USETO D, E	Similar to USETI but sets the output block number. USETO waits until the device is inactive before setting up the new output block number.
UGETF D, E	Places the number of the first free block of the file in user's location E.
ENTER D, E	User's locations E, E+1, E+2, and E+3, must be reserved for a directory entry. The DECtape service routine searches the directory for a filename and extension that match the contents of E and the left half of E+1. If no match is found and there is room in the directory, the service routine places the first free block number into the right half of E+1, places the date in E+2 (unless already nonzero), and places the necessary information into the directory. If a match is found, similar actions occur, but the new entry replaces the old. If there is no room in the directory, ENTER returns to the next location. Otherwise, ENTER skips one location.

[1]The Monitor compresses the user's core image by squeezing out blocks of two or more consecutive zeroes before creating the SAVed files; files with extension .SAV may be read in Dump Mode, but must be re-expanded before being run. The Monitor takes this action after input on a RUN or GET.

Table 5-2 (Cont)
DECtape Programmed Operators

Programmed Operator	Effect
LOOKUP D, E error return	Similar to ENTER but sets up an input file. The contents of E and E+1 are matched against the filenames and extension names in the DECtape directory. If a match is found, information about the file is read from the directory into the appropriate portions of the 4-word block beginning at E. The first block of the file is then found as follows. 1. The first 83 words of the DECtape directory are searched in a backwards manner, beginning with the slot immediately prior to the directory block, until the first slot containing the desired file number is found. 2. The block associated with this slot is then read in and bits 18 through 27 of the first word of the block (these bits contain the block number of the first block of the file) are checked. If they are equal to the block number of this block, then this block is the first block of the file; if not, then the block with that block number is read as the first block of the file. LOOKUP then skips one location. If no match is found, LOOKUP returns to the user's program at the next location.
CALL D, [SIXBIT/UTPCLR/]	UTPCLR clears the directory of the DECtape on device channel D. A cleared directory has zeroes in the first 83 words except in those slots related to blocks 0, 1, 2, and 100_{10} and nonexistent blocks 1102 through 1105_8. Only the directory block (block 100) is affected by UTPCLR; the other blocks are unaffected. This programmed operator does nothing if the device on channel D is not DECtape.

For both INPUT and OUTPUT, block 100 (the directory) is treated as an exception case. If the user's program gives

USETI D, 144_8

to read block 100, it is treated as a 1-block file.

The CLOSE operator places a -1 in the left half of the first word in the last output buffer, thus, terminating the file.

The RELEAS operator writes the copy of the directory which is normally kept in core onto block 100, but only if any changes have been made. Certain console commands, such as KJOB or CORE 0, perform an implicit RELEAS of all devices and, thus, write out a changed directory even though the user's program failed to give a RELEAS.

5.6.6 Special Status Bits

If an attempt is made to write on a unit with the WRITE-LOCK switch on, the improper mode flag (bit 18) is set in the file status word.

5.6.6.1 Special DECtape Status Bits - An INIT or SETSTS to a DECtape with bit 29 ON informs DTASER (the DECtape service routine) that the DECtape is in nonstandard format. This implies that no file-structured operations will be performed on that tape. Blocks will be read or written sequentially; no links will be generated (output) or recognized (input). The first block to be read or written must be set by a USETI or USETO. In Dump Mode, 200_8 data words per block will be read or written (as opposed to the normal 177_8 words). No "dead reckoning" will be used on a search for a block number, as the tape may be composed of blocks shorter than 200 words. The ENTER, LOOKUP, and UTPCLR UUOs are treated as no-ops. Block 0 of the tape may not be read or written in Dump Mode if bit 29 is ON, as the data must be read in a forward direction and block 0 normally cannot be read forward.

5.6.7 Important Considerations

The DECtape service routine reads the directory from a tape the first time it is required to perform a LOOKUP, ENTER, or UGETF; the directory image remains in core until a new ASSIGN command is executed from the console. To inform the DECtape service routine that a new tape has been mounted on an assigned unit, the user must use an ASSIGN command. The directory from the old tape could be transferred to the new tape, thus destroying the information on that tape unless the user reassigns the DECtape transport every time he mounts a new reel.

5.7 MAGNETIC TAPE

Magnetic tape format is industry compatible, 7-channel 200, 556, and 800 bpi and is not described here.

Device Mnemonic - MTA0, MTA1,...,MTA7

Buffer Size - 203_8 words

5.7.1 Data Modes

5.7.1.1 A (ASCII) - Data is written on magnetic tape exactly as it appears in the buffer. No processing or checksumming of any kind is performed by the service routine. The parity checking of the magnetic tape system is sufficient assurance that the data is correct. Normally, all data, both binary and ASCII, is written with odd parity and at 556 bits per inch. A maximum of 200 words per record is standard. The word-count is not written on the tape.

5.7.1.2 AL (ASCII Line) - Same as A.

5.7.1.3 I (Image) - Same as A but data consists of 36-bit words.

5.7.1.4 IB (Image Binary) - Same as I.

5.7.1.5 B (Binary) - Same as I.

5.7.1.6 DR (Dump Records) - Variable length records are read into or written from anywhere in the user's core area without regard to the standard buffering scheme. Control for read or write operations must be via a command list in core memory. The command list format is as described in Chapter 4, "Unbuffered (Dump) Modes." For input operations a new record is read for each word in the command list (except GOTO words); if the record terminates before the command word is satisfied, the service routine skips to the next command word. If the command word runs out before the record terminates, the remainder of the record is ignored. For each output command word, exactly one record is written. See Section 4.4.1.2 for command list format.

5.7.1.7 D (Dump) - This mode is accepted but actually functions as DR mode 16.

5.7.2 Special Programmed Operator Service

CLOSE performs a special function for magnetic tape. When an output file is closed (both dump and nondump), the I/O service routine automatically writes two end-of-file marks and backspaces over one of them. If another file is now opened, the second end-of-file is wiped out leaving one end-of-file between files. At the end of the in-use portion of the tape, however, there appears a double end-of-file character which is defined as the logical end of tape. When an input dump file is closed, the I/O service routine automatically skips to the next end-of-file.

A special programmed operator called MTAPE provides for such tape manipulation functions as rewind, backspace record, backspace file, etc. The format is

MTAPE D, FUNCTION

where D is the device channel on which the magnetic tape unit is initialized. FUNCTION is selected according to the following table:

Table 5-3
MTAPE Functions

Function	Action
0	No operation; wait for spacing and I/O to finish
1	Rewind to load point
11	Rewind and unload[1]
7	Backspace record
17	Backspace file
3	Write end of file
6	Skip one record
13	Write 3 inches of blank tape
16	Skip one file
10	Space to logical end of tape

[1]On the 516 Control, this function is not currently implemented as such, but is treated as a Rewind function only.

MTAPE waits for the magnetic tape unit to complete whatever action is in progress before performing the indicated function , including no operation (0). Bits 18 through 25 of the status word are then cleared, the indicated function is initiated, and control is returned to the user's program immediately. It is important to remember that when performing buffered input/output, the I/O service routine can be reading several blocks ahead of the user's program. MTAPE affects only the physical position of the tape and does not change the data that has already been read into the buffers.

5.7.3 Special Status Bits

Special bits of the status word are reserved for selecting the density and parity mode of the magnetic tape. Table 5-4 lists the bits that are set and cleared by INIT or SETSTS.

Table 5-4
Magnetic Tape Special Status Bits

Bit	Action
18[1]	Improper mode. When set to one during an output operation means that the write protect ring is out.
24[1]	I/O Beginning of Tape. The tape is at the load point.

[1]These bits indicate special magnetic tape conditions and are set by the magnetic tape service routine when the conditions occur.

Table 5-4 (Cont)
Magnetic Tape Special Status Bits

Bit	Action
25[1]	I/O Tape END. The tape is at or past the end point.
26	I/O Parity. 0 for odd parity, 1 for even parity.[1]
27-28	I/O Density. 00 or 10 = 556 bpi 01 = 200 bpi 11 = 800 bpi
29	I/O No Read Check. Suppress automatic error correction if bit 29 is a 1. Normal error correction is to repeat the desired operation 10 times before setting an error status bit.

[1] Odd parity is preferred. Even parity should be used only when creating a tape to be read in BCD (Binary Coded Decimal) on another computer.

5.8 DISK

Device Mnemonic - DSK

Buffer Size - 203_8 words (of which 200_8 words are data)

5.8.1 Data Modes

5.8.1.1 A (ASCII) - Data is written on the disk exactly as it appears in the buffer. Data consists of 36-bit words.

5.8.1.2 AL (ASCII Line) - Same as A.

5.8.1.3 I (Image) - Same as A.

5.8.1.4 IB (Image Binary) - Same as I.

5.8.1.5 B (Binary) - Same as I.

5.8.1.6 DR (Dump Records) - Functions exactly the same as D.

5.8.1.7 D. Dump - Data is read into or written from anywhere in the user's core area without regard to the normal buffering scheme. Control for read or write operations must be via a command list in core memory. The command list format is as described in Chapter 4, "Unbuffered (Dump) Modes." The disk control automatically measures dump data into standard-length disk blocks of 200 octal words. Unless the number of data words is an exact multiple of the standard length of a disk block (200 words) after each command word in the command list, the remainder of that block is wasted.

5.8.2 Structure of Files on Disk

The file structure of the disk system has been designed to minimize the number of disk seeks for sequential or random accessing using either buffered or dump mode I/O. The assignment of physical space for data is performed automatically by the Monitor as logical files are written or deleted by user programs. Files may be of any length, and each user may have as many files as he wishes, as long as disk space is available. No initial estimate of file length or number of files need be given by users or their programs. Files may be simultaneously read by more than one user at a time, thus allowing data sharing. A new version of a file may be recreated by one user while other users continue to read the old version, thus allowing for smooth replacement of shared programs and data files. Finally, one user may selectively update portions of a file, rather than creating a new one (see "General Notes," 5.8.3.3).

5.8.2.1 Addressing by Monitor - The file structure described in this section is generally transparent to the user, and a detailed knowledge of this material is not essential for effective user-mode use of the disk. There are two programs in the Time-Sharing Monitor that service the disk, DSKSER and DSKINT. DSKSER is the device service routine for a disk and references a disk by symbolic addressing only. This routine is essentially independent of what physical disk is attached to the system. DSKINT serves only two functions: 1) that of translating the logical addressing used elsewhere in the system to the physical addressing of the particular disk being utilized, and 2) controlling the physical disk. The monitor can be thought of as seeing all disks in the same manner; a change of disks requires only a change in DSKINT to provide the proper software interface between the physical device and the rest of the system.

All references made herein to addresses on the disk refer to the logical or relative addresses used by the system and not to any physical addressing scheme involving records, sectors, tracks, etc., that may pertain to a particular physical device. The basic unit which may be addressed is a logical disk block which consists of 200_8 36-bit words.

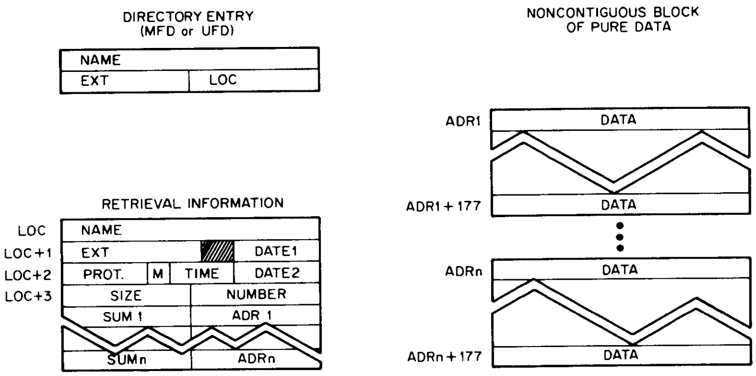
5.8.2.2 Storage Allocation Table (SAT) Blocks - There is a storage allocation table on the disk, which reflects the current status of every addressable block on the disk. These SAT blocks are contained in a file with the name "*SAT* .SYS". This file may be used by any user, but can only be modified by the Monitor. Each addressable block on the disk is represented by one particular bit within the SAT blocks.

If a particular bit is on, it indicates that the corresponding block is filled with data (all blocks on the disk are filled when any information is written on them); if the bit is off, it indicates that the corresponding block is empty or available to be written on. The disk can be wiped out by zeroing the SAT blocks (which is exactly what is done when the disk is refreshed). The disk may optionally be "refreshed" whenever the Monitor is reloaded.

5.8.2.3 File Directories - There are two levels of directories on the disk; one is referenced mainly by the system and the other is referenced by individual users. There is only one higher level directory, known as the Master File Directory (MFD). One of the functions of the MFD is to serve as a directory for individual User's File Directories (UFD's). A UFD is a particular user's own directory and will contain the names of files he has written on the disk. The UFD itself is a file like any other file except that its filename is a binary number combination (project-programmer) rather than a 6-bit code and its extension is always UFD in SIXBIT. The binary combination consists of a left half, which is the project number, and a right half, which is called the programmer number. When a user is logged in under a specific project-programmer number and references the disk, he is actually referencing his own area through the UFD having his project-programmer number as its name. He may, of course, specifically code his routine to reference files listed in the UFD's of other users or the MFD; whether he is successful or not will then depend upon the type of protection that has been specified for the file he is trying to reference.

5.8.2.4 File Format - All disk files (including MFD and UFDs) are composed of two parts: 1) pure data, and 2) information needed by the system to retrieve this data. Each data block contains exactly 200 (octal) words. If a partially filled buffer is output to the disk by a user, a full block is written with trailing zeros filling in to make 200_8 words. Word counts associated with individual blocks are not retained by the system. If such a partial block is input later, it will appear to have a full 200_8 data words.

There are three links in the chain by which the system references data on the disk. The first link is the 2-word directory entry in the UFD, which points to the Retrieval Information block(s), which in turn points to the individual pure data blocks. This chain is transparent to the user, who may look upon the directory as having 4-word entries analogous to DECtapes.



Size = n

Directory Entry

NAME - Filename in 6-bit ASCII, unless the directory is the MFD and the file is a UFD; in that case, NAME is a project-programmer number in binary.

EXT - Filename extension in 6-bit ASCII; if NAME is a project-programmer number, EXT is UFD.

LOC - Address of the first block on the disk that contains Retrieval Information for this file.

Retrieval Information

NAME and EXT as above; used to check hardware for possible read error, and to check against software malfunctions. (A failure to match NAME and EXT results in the message "INCORRECT RETRIEVAL INFORMATION".)

DATE1 - In format of DATE UUO; date file last referenced (RENAME, or ENTER, or INPUT done). (Bits 24-35)

DATE2 - Same format as DATE1; date file originally created (ENTER) (bits 24-35).

PROT. - Protection; see below (bits 0-8).

M - Data Mode (ASCII, Binary, Dump, etc.) (bits 9-12).

TIME - 24-hour time (in minutes) that file was originally created (bits 13-23).

SIZE - If negative, this portion indicates the number of words in the file, where all blocks with the possible exception of the last are assumed to contain a full 200_8 words. If positive, this is a count of the number of 200_8-word blocks contained in the file. For files of less than 2^{17} words, the negative word count is used; for larger files, the positive block count is used instead.

NUMBER - Programmer Number.

SUM 1, ...SUMn - Checksum; two's complement, end-around-carry, sum of data in data-block whose disk address is ADR 1.

ADR1, ...ADRn - Address of data block (logical block number on disk).

Protection

The first nine bits of the third word of a file's retrieval information are used to specify the protection of the file. This is a necessary procedure since the disk is shared by many users, who may each desire to keep certain files from being written over, read, or deleted by other users.

The total number of users is divided into three categories:

a. Owner of file; (person whose programmer number is the same as that in the left half of the name of the UFD in which the file is entered).

b. Project members; (users whose project number is the same as that in the right half of the name of the UFD in which the file is entered).

c. All other users.

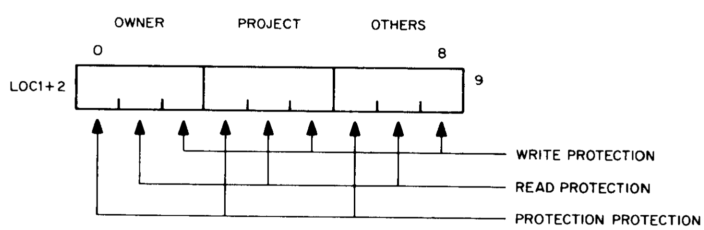
There are three types of protection against each of the three categories of users:

(1) Protection - The protection itself cannot be altered.

(2) Read protection - The file may not be read.

(3) Write Protection - The file may not be rewritten, renamed, or deleted.

The protection mask (see above) consists of the first nine bits of the third word of retrieval information; each bit (when on) represents a particular type of protection against a specific category of user, according to the following scheme. However, owner protection-protection and owner read-protection are ignored lest the file become totally inaccessible.



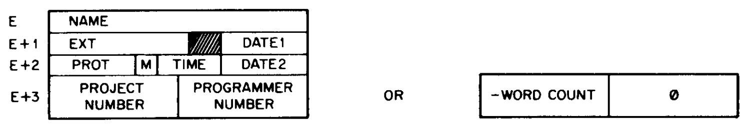
All files created with an ENTER are given the protection, 055_8 by the Monitor; if some other protection mask is desired, the RENAME UUO must be employed by the user. (Also see Section 4.4.2.5, "File Protection".)

5.8.3 User Programming for the Disk

5.8.3.1 Format - The actual file structure of the disk is generally transparent to the user. In programming for input/output on the disk, a format analogous to that of DECtapes is used; that is, the user assumes a 4-word directory entry similar in form to the first four words of retrieval information. The UUO format is approximately the same as for DECtapes:

UUO D, E

Where UUO is an input/output programmed operator and D specifies the user channel associated with this device. E points to a 4-word directory entry in the user's program which has the following format:



(Note that E+3 differs from the fourth word of retrieval information)
(See Retrieval Information, Paragraph 5.8.2.4 for description)

5.8.3.2 Special Functions of Programmed Operators (UUO's) -

ENTER D,E
error return

Causes the Monitor to store away the 4-word directory entry for later entry into the proper UFD when user channel D is CLOSEd or RELEASed.

NAME - The filename must be non-zero, if not, an error return results.

EXT - The file extension may be zero; if so, the Monitor will leave it zero.

DATE1 - The correct date is always filled in by the Monitor.

PROT - The protection is always supplied by the Monitor as 055. The RENAME may be used to change protection after file has been completely written and a CLOSE done.

M - The data mode is supplied by the Monitor as set by the user in the last INIT, or SETSTS UUO on channel D.

TIME, DATE2 - If both of these are 0, the Monitor supplies the current date and time as the creation date and time for the file. If either is non-zero, the Monitor will use the TIME and DATE2 supplied by the user in E+2; thus files may be copied without changing the original creation time and date.

PROJECT-NUMBER, PROGRAMMER-NUMBER - If both of these are 0, the project-number and programmer-number (binary) under which the user is logged-in is supplied by the Monitor. Otherwise the Monitor will use the project-number and programmer-number supplied by the user in E+3, however, it is generally not possible to create (ENTER) files in another user's area of the disk, since UFDs are usually write-protected against all but the owner.

With certain types of error returns peculiar to the disk, the right half of E+1 is set to a specific number to indicate which type of error caused the return. These numbers have the following significance:

0 - E contained a zero file name

1 - E+3 contained an incorrect (or nonexistent) project-programmer number.

2 - File already exists, but is write-protected.

3 - File was being created, recreated, updated, or renamed.

No user, except an administrator with project number 1, may create a UFD, since the MFD is universally write-protected. The LOGIN CUSP (running under the administrator project number) creates a UFD for any user the first time he logs into the system.

When an ENTER is executed by the Monitor on a file that already exists, a new file by that name is written, and those bits in the SAT blocks that correspond to the blocks of the old file are zeroed when the CLOSE (or RELEASE) UUO is executed thereby retrieving space and making it available to any other user.

LOOKUP D, E
error return

Causes the Monitor to read the appropriate UFD. If a later version of the file is being written, the old version pointed to by the UFD will be read.

NAME - The filename in SIXBIT

EXT - The file extension in SIXBIT. A zero extension is not treated in any special manner.

DATE1, PROT, M, TIME, DATE2 are ignored. The Monitor returns these quantities to the user in E+1 and E+2.

PROJECT-NUMBER, PROGRAMMER-NUMBER - If both of these are 0, the project-number and programmer-number (binary) under which the user is logged-in is supplied by the Monitor. Otherwise the Monitor will use the project-number, programmer-number supplied by the user in E+3. Thus, it is possible to read files in other user's directories, provided that the file's protection mask permits reading. The Monitor returns the negative word count (or positive block count for large files) in the LH of E+3, 0 in RH of E+3.

The numbers placed by the Monitor in the right half of E+1 upon an error return have a significance analogous to that described for the ENTER UUO:

0 - File was not found

1 - Incorrect project-programmer number in E+3

2 - Protection failure

3 - File was being created (no earlier version existed).

If the file is currently being recreated, the old file is used.

RENAME D, E
error return

This programmed operator is used to alter the name, extension, and/or protection of a file or to delete a file from the disk. Locations E through E+3 are as described above. RENAME is the only UUO that can set the protection of a file to that specified in E+2. To be RENAMEd a file must first be CLOSEd on channel D, in order to identify for the RENAME UUO. RENAME then seeks out this file and enters the information specified in E through E+2 into the retrieval information and proper directory. If the contents of E is zero, RENAME has the effect of deleting the file.

The error return numbers in the right half of E+1 are the same as for ENTER, with the added possibilities:

4 - Tried to RENAME file to already-existing name.

5 - Neither LOOKUP nor ENTER has been done to identify the file to be renamed.

USETO D, A

USETI D, A

These programmed operators are treated identically by the disk service routines. Their function is to notify the service routine that a particular block is to be used on the next INPUT or OUTPUT on channel D. A is a number that designates a particular block relative to the beginning of the file. If A is greater than the current size of the file (in blocks), the next OUTPUT will write a block immediately after the file; the next INPUT will cause the end-of-file flag to be set. Block 1 is the first block of file (0 is equivalent to 1).

If A = 0 or if no previous LOOKUP or ENTER has been done, this UUO will set the improper mode error bit (see bit 18, Table 4-4, and Section 4.4.4).

5.8.3.3 General Notes - Three types of "writing" on the disk may be distinguished. If a user does an ENTER with a filename which did not previously exist in his UFD, he is said to be "creating" that file. If the filename did previously exist in his UFD, he is said to be "recreating" that file; the old version of the file stays on the disk (and is available to anyone who wants to read it) until the user does the output CLOSE (at this point, his UFD is changed to point to the new version of the file and the old version is either deleted immediately or marked for deletion later if someone is currently reading it; the space occupied by deleted files is always reclaimed in the SAT tables - see Section 5.8.2.2). Finally, if a user does a LOOKUP followed by an ENTER (the order is important) on the same filename on the same user channel, he will be able to modify selected blocks of that file, using USETO and USETI UUOs, without creating an entirely new version of it; this third type of writing is called "updating" and eliminates the need to copy a file when making only a small number of changes.

As a standard practice, user programs should read, create, and recreate (new file with same filename) files on different user channels. However, for compatibility with DECtapes, it is possible to read and create, or read and recreate, two files on the same user channel as long as all OUTPUTs and the CLOSE output are done before any INPUTs and a CLOSE input, or vice versa.

When issuing a RENAME UUO, the user must insure that the status at locations E through E+3 are as he desires them to be. Since an ENTER or LOOKUP, as well as CLOSE, must have preceded the RENAME; the contents of E through E+3 will have been altered, or filled if the E is the same for all UUO's.

> CALL [SIXBIT/RESET/] - Any files which are in the process of being written, but have not be CLOSEd or RELEASed, will be deleted and the space reclaimed. If a previous version of the file with the same name and extension existed, it will remain on the disk (and in the UFD) unchanged.
>
> If the programmer wants to retain the newly created file and have the older version deleted, he must CLOSE or RELEASE the file before doing a RESET UUO.

5.9 INCREMENTAL PLOTTER

Device Mnemonic - PLT

Buffer Size - 43 (octal) words

5.9.1 The plotter takes 6-bit characters with the bits of each character decoded as follows:

Pen Raise	Pen Lower	-X Drum Up	+X Drum Down	+Y Carriage Left	-Y Carriage Right

Do not combine pen raise or lower with any of the position functions. (For more details on the incremental plotter, see the PDP-10 System Reference Manual, DEC-10-HGAA-D.)

5.9.2 Data Modes

5.9.2.1 A (ASCII) — Five, 7-bit characters per word are transmitted to the plotter exactly as they appear in the buffer. Since the plotter is a 6-bit device, the leftmost bit of each character is ignored.

5.9.2.2 AL (ASCII LINE) — This mode is identical to the A mode.

5.9.2.3 I (IMAGE) — Six, 6-bit characters per word are transmitted to the plotter exactly as they appear in the buffer.

5.9.2.4 B (BINARY) — This mode is identical to the I mode.

5.9.2.5 IB (IMAGE BINARY) — This mode is identical to the I mode.

5.9.2.6 DR (DUMP RECORDS) — Not available.

5.9.2.7 D (DUMP) — Not available.

5.9.3 The first OUTPUT operator causes the plotter pen to be lifted from the paper before any user data is sent to the plotter. The CLOSE operator causes the plotter pen to be lifted after all user data is sent to the plotter. These two pen-up commands are the only modifications the monitor makes to the user output file.

5.10 DISPLAY WITH LIGHT PEN (TYPE 30 and TYPE 340)

Device Mnemonic - DIS

Buffer Size - None (uses device-dependent dump mode only - 15)

5.10.1 Data Words

5.10.1.1 ID (Image Dump - 15)

An arbitrary length area in the user area may be displayed on the scope. The command list format is as described in Chapter 4, "Unbuffered (Dump) Modes," with the addition for the Type 30 display, that, if RH = 0, and LH $\neq$ 0, then LH specifies the intensity for the following data (4 to 13).

5.10.2 Background

The purpose of the monitor service routine for the VR-30 is to maintain a flicker-free picture on the display during time-sharing. To do this, the picture data must be available for display at least every two jiffies. This necessitates that the display data remain in core. At present, this means that the user program must also remain in core. To minimize swapping of other programs and to make available a larger block of free core for othér users, the user program is shuffled toward the top of core between pictures.

5.10.3 Display UUO's

The input/output UUO's for both displays operate as follows:

```
INIT D, 15              ;MODE 15 ONLY
SIXBIT /DIS/            ;DEVICE NAME
0                       ;NO BUFFERS USED
ERROR RETURN            ;DISPLAY NOT AVAILABLE
NORMAL RETURN

CLOSE D,                ;STOPS DISPLAY AND
    or                  ;RELEASES DEVICE AS
RELEAS D,               ;DESCRIBED IN MANUAL
```

5.10.3.1 INPUT D, ADR

If a light pen hit has been detected since the last INPUT command, then ADR = location of last light pen hit.

If no light pen hit has been detected since last INPUT command, then ADR = -1.

5.10.3.2 OUTPUT D, ADR

ADR specifies the first address of a table of pointers. This table is composed of pointers with the following format:

0 — 17	18 — 35
LH	RH

For the VR-30 Display:

If LH = 0 and RH = 0, then this is the end of the command list.

If LH $\neq$ 0 and RH = 0, then LH is the desired intensity for the following data or commands. The intensity ranges from 4 to 13, where 4 is the dimmest and 13 is the brightest.

If LH = 0 and RH $\neq$ 0, then RH is the address of the next pointer. Successive pointers are interpreted beginning at RH.

If LH $\neq$ 0 and RH $\neq$ 0, then -LH words beginning at address RH+1 are output as data to the display. The format of the data word is the following:

0 – 7	8 – 17	18 – 25	26 – 35
	y-coord		x-coord

For the 340 Display:

If RH = 0, then this is the end of the command list.

If LH = 0 and RH $\neq$ 0, then RH is the address of the next pointer. Successive pointers are interpreted beginning at RH.

If LH $\neq$ 0 and RH $\neq$ 0, then -LH words beginning at address RH+1 are output as data to the display. The format of the data word is described in the 340 programming manual.

An example of a valid pointer list for the VR-30 Display is:

```
        OUTPUT    D, LIST       ;OUTPUT DATA
                                ;POINTED TO BY LIST
LIST:   XWD       5, 0          ;INTENSITY 5 (DIM)
        IOWD      1, A          ;PLOT A
        IOWD      5,SUBP1       ;PLOT SUBPICTURE 1
        XWD       13,0          ;INTENSITY 13 (BRIGHT)
        IOWD      1,C           ;PLOT C
        IOWD      2,SUBP2       ;PLOT SUBPICTURE 2
        XWD       0,LIST1       ;TRANSFER TO LIST 1
LIST1:  XWD       10,0          ;INTENSITY 10 (NORMAL)
        IOWD      1,B           ;PLOT B
        IOWD      1,D           ;PLOT D
        XWD       0,0           ;END OF COMMAND LIST
A:      XWD       6,6           ;Y =    6, X = 6
B:      XWD       70,105        ;Y =   70, X = 105
C:      XWD       105,70        ;Y =  105, X = 70
D:      XWD       1000,200      ;Y = 1000, X = 200
SUBP1:  BLOCK     5             ;SUBPICTURE 1
SUBP2:  BLOCK     2             ;SUBPICTURE 2
```

An example of a valid pointer list for the 340 Display is:

```
        OUTPUT    D, LIST       ;OUTPUT DATA POINTED
                                ;TO BY POINTER IN LIST
LIST:   IOWD      1,A           ;SET STARTING POINT TO (6,6)
        IOWD      5,SUBP1       ;DRAW A CIRCLE
        IOWD      1,C           ;SET STARTING POINT TO (70, 105)
        IOWD      5,SUBP1       ;DRAW A CIRCLE
```

```
        IOWD    1,B        ;SET STARTING POINT TO (105, 70)
        IOWD    2,SUBP2    ;DRAW A TRIANGLE
        XWD     0,LIST1    ;TRANSFER TO LIST1
LIST1:  IOWD    1,D        ;SET STARTING POINT TO (1000, -200)
        IOWD    5,SUBP1    ;DRAW A CIRCLE
        IOWD    1,A        ;SET STARTING POINT TO (6,6)
        IOWD    2,SUBP2    ;DRAW A TRIANGLE
        XWD     0,0        ;STOP
A:      X = 6       Y = 6
B:      X = 105     Y = 70
C:      X = 70      Y = 105
D:      X = 1000    Y = -200
SUBP1:  BLOCK   5          ;DRAW A CIRCLE
SUBP2:  BLOCK   2          ;DRAW A TRIANGLE
```

The example shows the flexibility of this format. The user can display a subpicture by merely setting up a pointer to it. He can also display the same subpicture in many different places by setting up pointers to the subpicture, each preceded by a pointer to commands for the display to reset its coordinates.

5.11 CALL AC, [SIXBIT/DEVCHR/] or CALLI AC, 4

The user may determine the physical characteristics associated with a logical device name by executing a DEVCHR UUO. The DEVCHR UUO returns the following information in the AC referred.

$(AC)_L$:	1	Device can do output
	2	Device can do input
	4	Device has a directory (DTA or DSK)
	10	Device is a TTY
	20	Device is a magnetic tape
	40	Device is available to this job or is already assigned to this job
	100	Device is a DECtape
	200	Device is a paper tape reader
	400	Device is a paper tape punch
	1000	Device has a long dispatch table (that is, UUO's other than INPUT, OUTPUT, CLOSE, and RELEASE perform real actions)
	2000	Device is a display
	4000	TTY in use as an I/O device
	10000	TTY in use as a user console (even if detached)
	20000	TTY attached to a job
	40000	Device is a line printer

	100000	Device is a card reader
	200000	Device is a disk
	400000	DECtape directory is in core (this bit is cleared by an ASSIGN or DEASSIGN command to that unit)
$(AC)_R$:	400000	Device assigned by a console command
	200000	Device assigned by program (INIT UUO)

Remaining Bits: If bit 35-n contains a 1, then mode n is legal for the device.

NOTE

The mode number (0 through 17) must be converted to decimal; for example, mode 17_8 is represented by bit $35-15_{10}$ or bit 20.

READER'S COMMENTS

Digital Equipment Corporation maintains a continuous effort to improve the quality and usefulness of its publications. To do this effectively, we need user feedback: your critical evaluation of this manual and the DEC products described.

Please comment on this publication. For example, in your judgment, is it complete, accurate, well-organized, well-written, usable, etc? ____________________

Did you find this manual easy to use? ____________________

What is the most serious fault in this manual? ____________________

What single feature did you like best in this manual? ____________________

Did you find errors in this manual? Please describe. ____________________

Please describe your position. ____________________

Name ____________________ Organization ____________________

Street ____________________ State ____________________ Zip ____________

Fold Here

Do Not Tear - Fold Here and Staple

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Digital Equipment Corporation
Software Quality Control
Building 12
146 Main Street
Maynard, Mass. 01754

DIGITAL EQUIPMENT CORPORATION □ MAYNARD, MASSACHUSETTS

PRINTED IN U.S.A.