TIME-SHARING SYSTEM
TSS/8 Monitor
for the
PDP-8
PDP-8/I

For additional copies order No. DEC-T8-MRFA-D from Program Library, Digital Equipment Corporation, Maynard, Mass. Price $3.00

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

First Printing, September 1968

The reader's attention is invited to the last two pages of this manual. The How To Obtain Revisions and Corrections offers the reader a means of keeping up-to-date with DEC's software. The Reader's Comments card, when filled in and returned, is beneficial to both the reader and DEC. Each card received is considered when documenting subsequent manuals, and where the comments imply or ask for assistance, a knowledgeable DEC representative will contact the reader.

ACKNOWLEDGMENT

We wish to thank Dr. John McCarthy and his associates at the Stanford Computation Center for their permission to use some parts of the THOR User's Manual which describes a time-sharing system for the PDP-1 computer.

CONTENTS

CONTENTS (Cont)

ILLUSTRATIONS (Cont)

HOW TO OBTAIN REVISIONS AND CORRECTIONS

Notification of changes and revisions to currently available Digital software and of new software manuals is available from the DEC Program Library for the PDP-5, 8, 8/S, 8/I, 8/L, LINC-8, the PDP-4, 7, and 9 is currently published in DECUSCOPE, the magazine of the Digital Equipment Computer User's Society (DECUS). This information appears in a section of DECUSCOPE called "Digital Small Computer News".

Revised software products and documents are shipped only after the Program Library receives a specific request from a user.

DECUSCOPE is distributed periodically to both DECUS members and to non-members who request it. If you are not now receiving this information, you are urged to return the request form below so that your name will be placed on the mailing list.

To: Decus Office,
Digital Equipment Corporation,
Maynard, Mass. 01754

☐ Please send DECUS installation membership information.

☐ Please send DECUS individual membership information.

☐ Please add my name to the DECUSCOPE non-member mailing list.

Name _____

Company _____

Address _____

_____
(Zip Code)

CHAPTER 1
INTRODUCTION

TSS/8 (Time-Sharing System for the PDP-8/I Computer) is a general purpose stand-alone, time-sharing system offering each of up to 32 users a comprehensive library of system, utility, and service programs which provide facilities for compiling, assembling, editing, loading, saving, calling, and debugging user programs on-line, and also two conversational languages, FOCAL[1] and BASIC-8[2].

The minimum hardware configuration is designed for eight users and includes the following: a modified PDP-8/I or PDP-8 with 8K words of core memory, a RF08 Disk, a PT8/I high-speed paper tape reader, and Teletype control units for data communication (either four PT08s or one 680/I per system). See Section 1.6 for a complete list of possible optional devices and for the configuration for systems accommodating up to 32 users.

By segregating the central processing operations from the time-consuming interactions with the human users, the computer can in effect work on a number of programs simultaneously. Giving only a fraction of a second at a time to each program or task, the computer can deal with many users seemingly at once, as if each had the computer to himself. The execution of various programs are interspersed without interfering with one another and without detectable delays in the responses to the individual users.

1.1 MONITOR FUNCTIONS

The heart of this time-sharing system is a complex of subprograms called Monitor. Monitor coordinates the operations of the various units, allocates the time and services of the computer to users and controls their access to the system. The allocation function includes scheduling the user's requests, transferring control of the central processor from one user to another, moving (swapping) programs in and out of core memory, and managing the user's private files.

1.2 TSS/8 USER AND CONSOLE

A TSS/8 user is any person running a computer program within TSS/8. He has an account number assigned to him by the person responsible for the system which determines his identity for TSS/8 and identifies his files on a permanent basis. When he is using the system this number also identifies

---

1 FOCAL (FOrmula CALculator) is an on-line conversational interpretive program developed by Digital Equipment Corporation.

2 BASIC-8 is a modified version of the elementary algebraic language developed at Dartmouth College.

his console, his individual disk swapping track (a contiguous 4K section of disk) for temporary storage of active programs, and whatever other facilities he may be using at any given time.

While the user is logged into the system, he owns at least one console (Figure 1-1) and one 4K disk track. The console consists of a keyboard (Figure 1-2), which allows the user to type information to his user programs and Monitor, and an output device, usually a Teletype printer, which furnishes typed copy of program and Monitor output and user input.

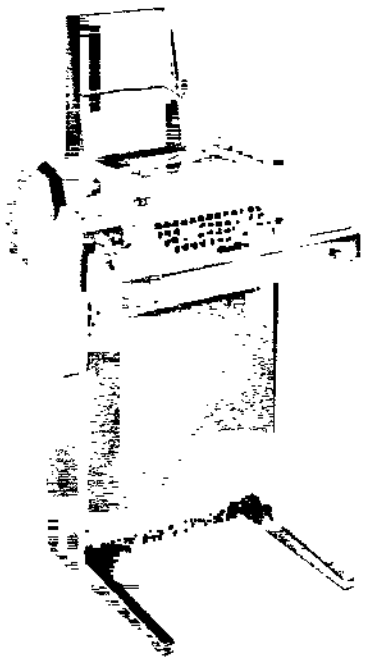

Figure 1-1 User Console

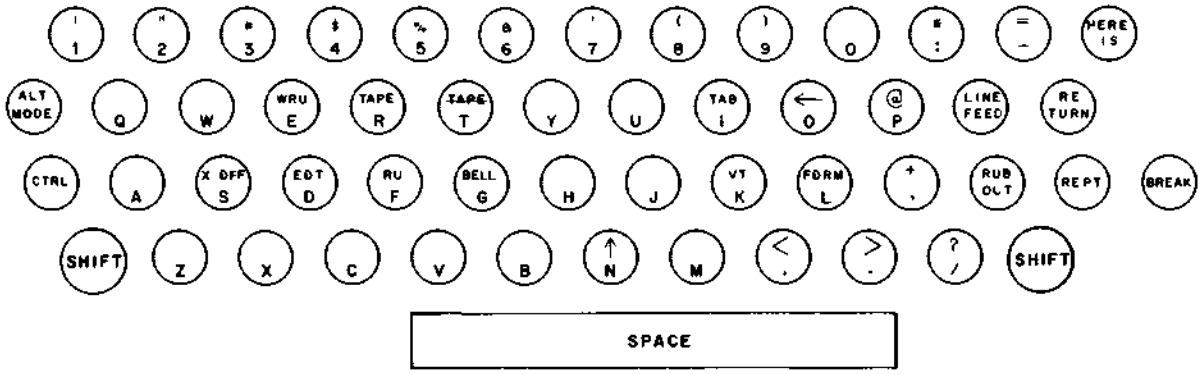

Figure 1-2 Users Console Keyboard

1.3 SYSTEM PROGRAM LIBRARY

A comprehensive library of programs is available to all TSS/8 users. The library provides facilities for assembling, compiling, editing, and debugging user programs. Any of the system, utility, and service programs can be called into use by typing, in response to the dot typed by Monitor, the command R and the assigned name of the programs. For example,

.R FOCAL

brings FOCAL into core from the disk and automatically executes FOCAL so that it begins typing out its Initial Dialogue.

The System Program Library contains the following programs:

a. Binary (BIN) Loader - used to load user programs into core memory from the console.

b. Symbolic Tape Editor - used to prepare, edit, and generate symbolic (source) programs and program tapes from the console.

c. PAL III Symbolic Assembler - used to translate source programs written in the PAL III language into object programs.

d. MACRO-8 Assembler - used to translate source programs written in the MACRO-8 language, containing macros and literals, into object programs.

e. Dynamic Debugging Technique (DDT-8) - used to aid the user in correcting errors in stored user programs by allowing him to execute small sections at a time, to stop execution where he wishes, and to change portions of his program, using the symbolic language of the source program or machine language. The user may create an octal program using the DEPOSIT command.

f. Octal Debugging Technique (ODT-8) - used for the same purpose as DDT-8 (above) except that the user communicates in the octal representation of his program. ODT-8 requires less core area than DDT-8, and can be loaded in either the upper or lower portion of core, depending upon where the user's program is loaded.

g. FORTRAN (4K) - used to compile and operate a user program written in the 4K version of the FORTRAN language; compilation requires only one pass through the compiler.

h. FOCAL (FOrmula CALculator) - an on-line, conversational, interpretive program used to solve complex numerical problems; used as a programming tool by students, engineers, and scientists.

i. BASIC-8 - an elementary algebraic language, is an on-line, service program similar to FOCAL used to solve complex numerical problems.

1.4 USER PROGRAMS

A user program is any program being run in user mode by TSS/8. When a user program is being run by TSS/8 it is swapped into core memory from the user's disk track. Several programs may be run by a regular process of bringing a program into core from the disk, allowing it to execute for a short time, marking the state in which its execution is stopped, returning it to the disk, and picking up the next user program.

User programs are serviced regularly on a "round-robin" basis. After a user program has been executed, it is placed last in the order of user programs waiting to run. Each program is allowed to run one quantum of time and then it is exchanged for the next user program. If only one program is in a condition to run, it is allowed to run without interruption.

There are three ways in which a user can place a binary program on his disk track.

a. He may prepare an octal program at a console using a debugging program such as DDT-8,

b. He may load a previously saved core image, or

c. He may load the binary output file of a previous assembly or compilation.

1.5 USER FILES

Disk storage is divided into logical areas called files. Facilities are available for creating new files and extending, contracting, and destroying old files. Files may contain textual information, binary core images, or data in any standard format. The user can also protect his files against unauthorized access.

A user file directory (see Appendix E) is assigned to each user by the person responsible for the system before the user first logs into the system. The user file directory is associated with the job number assigned to the user by the Monitor each time the user logs in.

Files are composed of segments of disk storage. A segment can consist of from 200_8 to 2000_8 words. Files are at least one segment long and grow by adding additional segments to the end of the file.

Each user may have access to up to four active files at any time. An internal file number, 0 through 3, is associated with each of the user's active files, and commands then operate in terms of the internal file numbers. With this feature, a user may attach one of his files to an internal file number and then load system and utility programs which operate on his file without having to explicitly call the file for each system or utility program used.

1.6 SYSTEM CONFIGURATION

Depending upon the hardware configuration of a particular TSS/8, there can be from 1 to 32 users working with the system simultaneously. Each user has, in addition to system software, the following resources: at least 4K words of core memory for execution of programs, and a corresponding 4K disk track for temporary storage of his core image.

1.6.1 Minimum Hardware Requirements

The minimum equipment requirements of the system are listed below (see Figure 1-3).

PDP-8/I or PDP-8 with KT08/I Time-Sharing Modifications
MC8/I-A Memory Extension Control and 4096 words
RF08 Disk Control
RS08 Disk
PT08 Asynchronous Line Interfaces, Dual (4)
PT8/I High-Speed Paper Tape Reader
KE8/I Automatic Multiply-Divider
CAB 8/IA Option Cabinet

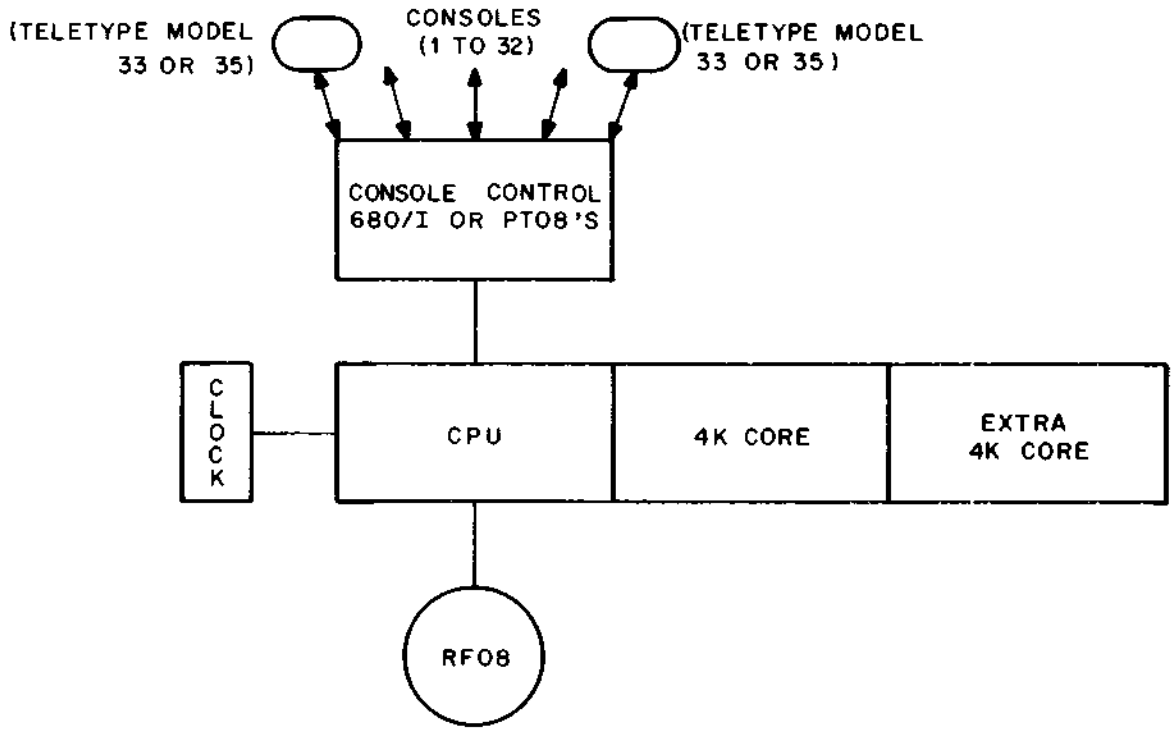

Figure 1-3 System Configuration

1.6.2 Hardware Options

The system can have a maximum of 32K of core memory. As additional fields of core memory are added they permit overlapping the running time of one user program with the swapping time of another, expanding the resident Monitor to buffer a larger number of consoles, and reducing the amount of Monitor overlay required.

With a minimum of 12K of core memory, the following options are available.

a. Up to three DECdisks can be appended to increase the number of active users and their file storage.

b. Up to eight DECtapes can be appended for the use of private DECtapes by users.

c. Memory Parity

d. Extended Arithmetic Element (PDP-8/I only)

e. Power Failure

f. 1 High-Speed Paper Tape Punch

g. 1 Card Reader

h. 1 Incremental Plotter

i. 1 Line Printer

j. 1 DA10 Interface (PDP-10)

k. 1 Bit Synchronous Communication Unit (Type 637)

l. 1 Line Frequency Clock (required with PT08's)

m. 32 Teletype Controls (PT08's or 680/1)

1.6.3 Data Communication System, Type 680/I

The Data Communication System, Type 680/1, is used in full duplex mode. It consists of a Data Line Interface Unit (DL8/1), a Serial Line Multiplexer Unit (685), and other devices connected to form a data link and message switching system between the user consoles and the central processor.

CHAPTER 2
THE TSS/8 MONITOR

2.1 MONITOR SERVICES

The TSS/8 Monitor is composed of the following routines:

IOT Trap Handler	Error Message Handler
Scheduler	Buffer Handler
Storage Allocator	680 Service
Overlay Control	Disk Service
System Interpreter	Optional Device Service
File Control	

With the above routines, Monitor provides services which may be divided into three broad categories:

a. Device Service

b. Scheduling and Activation

c. Communication

On an interrupt basis the Device Service routines receive information from all input devices, parcel that information out to the appropriate buffers, and inform the Activation routine when a user program must be activated to receive its information. The Device Service routines accept output from user programs, buffer it, and send it to output devices whenever they are able to receive it.

Activation is handled by a "round-robin" scheduling algorithm with the exception that programs with disk requests pending are run out of turn to optimize disk usage. The Activation routine decides whether to remove the current program from core, and if so, which program is to be run next.

A user program is, at any point in time, in one of the following states.

a. Running - The user program is in execution. It continues execution until its quantum has expired or until it issues an input/output (I/O) request that cannot be satisfied immediately.

b. Active - The user program is ready to run and will be swapped into core when its turn comes. A program can become active when

(1) An output buffer is almost empty, thus assuring continuous output of information,

(2) An input buffer is almost full,

(3) Input requested by a user has arrived,

(4) A user determined activation condition has been satisfied, or

(5) When the user commands the system to begin execution.

c. Waiting - In this state, the program would be active except that it is waiting for the completion of some I/O request or special condition.

d. Dormant - The program is not being entered into the "round-robin." The user may be communicating with Monitor or the program may be dismissed for a variety of user determined reasons.

A program may be swapped out of core memory for any of the following reasons.

a. The quantum has expired; the program moves from the running state to the active state.

b. The program has requested the quantum be terminated; the program moves from the running state to the active state.

c. The program has filled an output buffer, has requested input and the input buffer is empty, or has requested a special dismissal condition; the program moves from the running state to the waiting state.

d. The program has tried to execute an illegal instruction; the program moves from the running state to the dormant state.

Monitor checks all incoming characters for the call (CTRL/B) character. When the call character is encountered, Monitor routes all subsequent characters up to and including the first carriage return (CR) to its System Interpreter's input buffer. Monitor's System Interpreter routine performs the following services.

a. Verifies the user's name and account number when he logs in and provides him with a disk track to store user programs. It releases facilities owned by the user when he logs out.

b. Parcels out extra consoles and input/output devices.

c. Provides commands for creating, opening, and maintaining the user's files.

d. Allows the user to save all or part of his binary user program for future use and reference, and restores it with its state unchanged.

e. Provides accounting of console time and user program run time.

f. Provides the user with information about the state of his program while running, as well as information about the state of the character control tables.

g. Provides commands for calling the various utility programs.

Communications to the System Interpreter are automatically duplexed. That is, all characters from a keyboard to the System Interpreter appear on that console's printer, regardless of the switchboard setting. (See Section 2.3.)

The System Interpreter may be receiving messages from many keyboards and user programs, therefore, it must be run often. Consequently, it occupies its private position in the "round-robin," being activated whenever there are characters in its input buffer.

Monitor has two Phantom routines: Error and File Control. Phantoms are privileged routines which run in place of a regular user program in the "round-robin." The time the Phantom takes to perform its service is charged to that user program upon which it is servicing.

The Error Phantom prints all error messages for running user programs. Printing a lengthy error message may require several quanta; the use of the Error Phantom punishes the user responsible for the error and no one else.

The File Control Phantom handles such modifications to the file directory as creating, lengthening, renaming, and destroying files. This Phantom is brought into operation when a user program executes certain IOT instructions or when Monitor is acting for the user program.

2.2 CHARACTER AND DATA FLOW

When a user logs into the system his account number is associated with a job number, and that job number is then associated with a disk swapping track and the console(s) he owns. The account number establishes ownership of the input buffer attached to the user program on his disk track and the output buffers attached to the printers of his consoles.

Monitor provides for communication among the users, user programs, and Monitor. Data communication is illustrated in Figure 2-1. Communication is provided through the IOT instructions. When a program executes certain IOT instructions, Monitor picks up locations in the user program as parameters to service routines. These routines may simulate input/output to the on-line device, control or release ownership of devices, handle character communication, or return information to the user program by filling registers within the program. Through IOT instructions, the user program can make its wants known and Monitor can inform the user program of any variation in the time-sharing system.

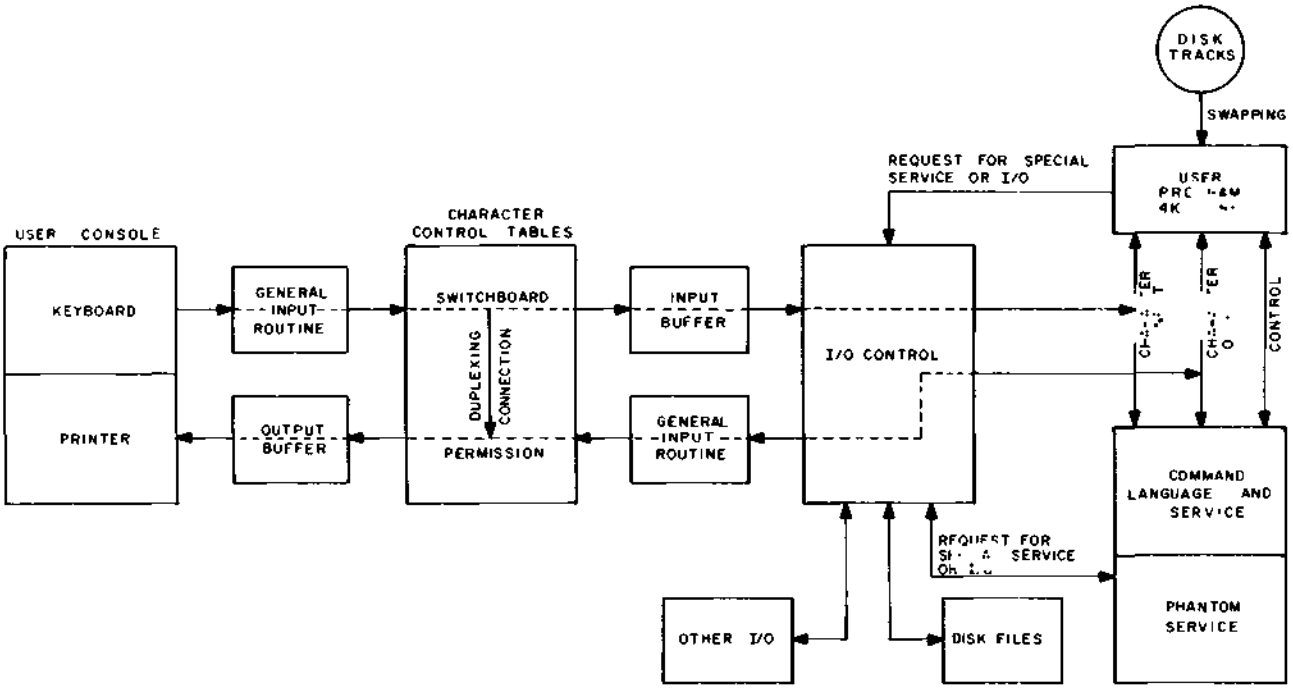

Figure 2-1 Character and Data Flow

2.2.1 Character Transmission

The input/output transfer instructions provide character transmission. Characters are generated by users typing at keyboards or by user programs typing out. Characters go into input buffers to be read by user programs or to output buffers to be printed on Teletype printers. A switchboard device sets up useful character transmission paths, whereby any character source (keyboard and program) may, with permission, send to any character sink (input and output buffer). To insure that unwanted connections are not made, using the appropriate command to Monitor the user owning any character sink may grant or deny permission for connection into that sink.

A console may be thought of as two disassociated devices, a keyboard and a printer. To each printer there is an associated character output buffer numbered like the consoles, O-K. When a character is placed into output buffer number n, Monitor causes that character to be printed on printer number n.

Each job has an associated character input buffer numbered like the job, O-P. When user program number n executes a keyboard input IOT, the next character waiting in input buffer number n is placed in the user program's accumulator (AC). This character may be a data character, the source of the last data character, or the time that the last data character was placed in the input buffer.

2.2.2 General Input Routine

The General Input Routine (GIR) provides and controls character input to user programs. Programs vary in how promptly they must pay attention to incoming characters. In the preparation of a file, characters are added to the input buffer when typed; when the input buffer gets full the program receives all the characters at once and transmits them to the disk. In this case, the user's program is activated only when the input buffer gets full. With a program whose operation is controlled from the keyboard, however, every character typed goes directly to and influences the program's action.

Through IOTs, the user program can specify under which conditions the input characters will cause the program to be activated. These conditions are called delimiters. A delimiter may be some specific type of character appearing in the input or it may be a change in the source of characters; the input buffer becoming full is always a delimiter.

GIR places all incoming characters in the user's input buffer until a delimiter is encountered. When a delimiter appears, the user program is always alerted and usually activated to receive all the characters in the input buffer. This is done by setting the delimiter bit (status register 1) to one. If the Wait Mask (see Chapter 4, Set Wait Mask) is set to one, the user program will be activated.

2.3 PERMISSION AND SWITCHBOARD TABLES

Characters originating from any keyboard or from any user program typing out can, with permission, be placed in any or all of the K output buffers or any or all of the P input buffers. The Permission Table is the device with which a user controls which character sources may place characters in which of his buffers; the Switchboard Table controls the actual routing of these characters.

The Permission and Switchboard Tables are organized as matrices (see Figures 2-2 and 2-3). The rows represent character sources, that is, keyboards and programs typing out. The columns represent character sinks, that is, input and output buffers. Along the rows are the K keyboards and P programs. Along the columns are the P input buffers and the K output buffers.

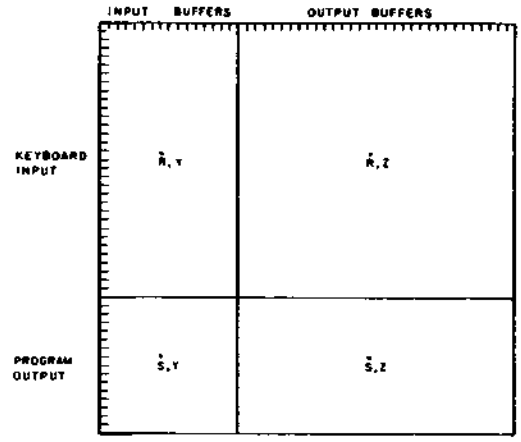

Figure 2-2 Permission Table

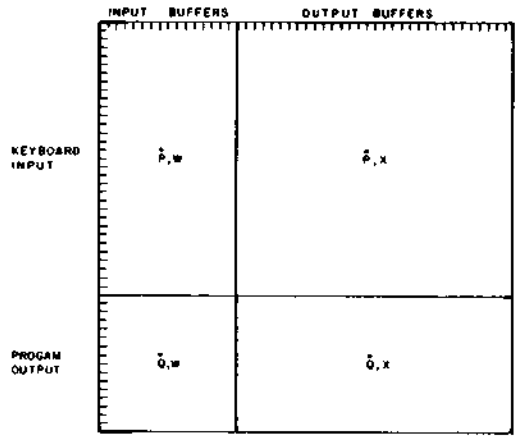

Figure 2-3 Switchboard Table

2.3.1 Permission Table

The Permission Table acts as the matrix of permitted connections. For example, if the bit R, Y is a one, it means that the owner of input buffer number Y allows keyboard number R to place characters in it. Similarly, if bit S,Z is a one, it means that the owner of output buffer number Z (printer Z) has agreed to allow job number S to type out characters onto printer Z. In general, if a bit in the Permission Table is a one, the owner of the entity specified by the column has granted permission for the entity specified by the row to place characters in his sink.

The fact that permission for a given character routing has been established does not necessarily mean that characters automatically flow along that route. A route to a buffer is not established unless the owner of the keyboard or program sets the Switchboard Table.

2.3.2 Switchboard Table

The Switchboard Table acts as a matrix of requests by the owner of a character source that characters emitted by that source appear in some sink. If bit Q,W is a one, characters coming from program number Q typing out will appear in input buffer number W. Similarly, if bit P,X is a one, characters typed on keyboard number P will appear in the output buffer of printer number X. In general, if a bit in the Switchboard Table is a one, characters emitted by the source specified by the row will appear in the sink specified by the column. Attempts to set bits in the Switchboard Table are automatically checked against the Permission Table. Only if the corresponding bit in the Permission Table is a one will the system allow that bit to be set in the Switchboard Table. The Switchboard Table may be regarded as the final arbiter of character routing.

Bits in the Permission and Switchboard Tables may be set, reset, and read by using Monitor commands or by an IOT instruction which the system executes interpretively.

The Permission and Switchboard Tables are automatically set up for normal use when a user logs into the system. This configuration is:

a. The user gives permission for his keyboard to send characters to his user program's input buffer.

b. The user gives permission for his keyboard to send characters to his printer's output buffer (this condition is known as duplexing). Whether the characters actually appear depends upon the corresponding Switchboard setting. Normally, the Switchboard is not set to handle the input before it is printed.

c. The user gives permission for his program to type characters to its own input buffer. (The Switchboard determines whether characters typed out will in fact be sent to the input buffer.)

d. The user gives permission for his program to send characters to his printer's output buffer.

e. The Switchboard is set so that characters from the user's keyboard will be placed in the user program's input buffer.

f. The Switchboard is set so that characters typed out by the user program will be placed in the program's output buffer.

The user is, of course, free to modify and add to these settings.

The Switchboard generality finds application in:

a. Duplexing - Characters typed at a keyboard can be printed on any console without user program intervention.

b. Interprogram Communication - User programs can communicate with each other and with Monitor. Thus, several user programs may run as one system coordinating their separate tasks through character communication.

c. Interconsole Communication - Users at consoles can set up general links for conferences, teaching, monitoring, or for any other reason.

d. Multiple Consoles - User programs may receive characters from and send characters to more than one console. This allows a user program to act as a time-sharing system within TSS/8, controlling its own set of consoles, as in teaching machine monitors.

CHAPTER 3
MONITOR COMMANDS

A Monitor command is a string of characters terminated by a semicolon (;) or a carriage return (RETURN key). Commands to Monitor are typed on the Teletype keyboard by the user or output by the user program, with each command beginning with a command name. In some cases, the command name is the entire command, in which case it is followed directly by a terminator. In other cases, the command name is followed by a space, one or more parameters, and then a terminator. For example,

.TIME	(requesting the time of day. This command was terminated by the RETURN key)
.TIME C1	(requesting the processing time used by job C1. Terminated by the RETURN key)

Only enough characters need be typed in the command name to uniquely identify the command name. For example,

LOGI	for LOGIN
LOGO	for LOGOUT

More than one command may be typed on a line, with all but the last command being terminated by a semicolon; the last command is terminated by the RETURN key. For example,

.OPEN C1 S1 C2; LOAD C1 S1; START C1

(The abbreviated parameters are explained below.)

NOTE

Commands are executed when the RETURN key is typed, which explains why the last command on a line must be terminated by the RETURN key.

As shown in the above examples, each command or line of commands is typed after the dot typed by Monitor. The dot is typed by Monitor when it is ready and available to accept commands from the user.

Parameters may be typed as octal numbers, decimal numbers, character strings, or single letters. In the following descriptions of Monitor commands the parameters are coded as follows.

C1, C2, ...	represent octal numbers
D1, D2, ...	represent decimal numbers
S1, S2, ...	represent character strings
L1, L2, ...	represent single letters

3.1 LOGGING IN AND OUT

Logging in and out of TSS/8 is a function performed by Monitor's System Interpreter routine. When a console is in the free state, a prospective user may attempt to log into the system.

LOGIN C1 S1; This is a request by a user to log into the system. If the console from which the command is typed is free, there is an available disk track, and the two parameters following the command LOGIN form a legitimate account number and password, then the user will be logged into the system.

C1 is the user's account number
S1 is the user's password

At the time of login, the Switchboard is initialized to the normal operation setting. However, Monitor diverts all characters typed to the System Interpreter until the user gives a command that indicates otherwise.

LOGOUT; This is a request by a user to logout of the system. It disconnects all consoles and disk tracks that he owns, places his programs in the free state, and resets the Switchboard. It also writes an account record on the disk showing how much computing time (processing time) and console time was used by the user's program(s).

TIME C1; This is a request for Monitor to type out the computing (processing) time used since login. If job C1 is not specified, the job owning the console is assumed. If requested before login and if no job is specified, the time-of-day is typed. If, at any time, job 0 is specified, the time-of-day is typed.

3.2 CONSOLE MANIPULATION

ASSIGN K C1; This is a request for console number C1 to be added to those consoles already owned by the requesting user.

K denotes console
C1 is the console number

If the console is free, it will be given to the user and the user's Switchboard settings will be augmented by those additional entries which put the new console in the normal state with respect to its own printer and the program on the user's disk track. Cross settings for inter-console connections are left to the user's discretion.

SLAVE C1; This command causes console number C1 to be slaved. This means that console number C1 will be unable to communicate with Monitor, and the system will ignore the call (CTRL/B) character from that console. However, the console may serve as an input/output device for the user's program. Monitor checks to make sure that the requesting user owns the console he wishes to slave. It is illegal to slave every console that a user owns.

UNSLAVE C1;

This command restores a slaved console to normal status. After this command, console number C1 will be able to communicate with Monitor. The user must own the console to unslave it.

RELEASE K C1;

This command releases console number C1 and puts it in the free state. Monitor checks to make sure that the requesting user owns the console he wishes to release. When the console is released, the Switchboard is reset and the input and output buffers are cleared. The user must own the console to release it.

3.3 DEVICE ALLOCATION

ASSIGN L1;

This is a request for access to the device specified by L1.

L1 = R for paper tape reader
P for paper tape punch
C for card reader
L for line printer
I for incremental plotter

If device L1 is not busy, it will be allocated to the user program issuing the command. If device L1 is busy, the job number of the user having possession is returned. A RELEASE L1; or LOGOUT; will release the device.

ASSIGN D C1;

This command is the same as ASSIGN L1;

D denotes DECtape unit
C1 is the unit number of the DECtape unit

RELEASE L1;
RELEASE K C1;
RELEASE D C1;

Each of these commands will annul the current assignment of the specified device.

3.4 FILE CONTROL

OPEN C1 S1 C2;

This command establishes association between an internal file number and a file. After this command is given, the file of account C2 with the name S1 is associated with internal file number C1. The internal file number specified must be between zero and three inclusive. If C2 is not specified, the account number of the current user is assumed.

CLOSE S1;

This command closes the files specified by S1.

S1 is a list of internal file numbers separated by spaces

After this command is given, no writing can be done on the files specified, and the associations between the internal file numbers and the files are broken.

CREATE S1;

This command causes Monitor to create a new file which is to have the name S1, if there is available file storage. At the time of creation, Monitor will enter the name of the new file and the date of creation into the owner's file directory (see Appendix E) and set the protection mask to 17. Example:

CREATE NEWF;	asks Monitor to create a new file named NEWF. If there is no more file storage, Monitor will so inform the user (see Chapter 5).

RENAME C1 S1;

This command renames a file. The file to be renamed must be already open and associated with internal file number C1. Its new name will become S1. Example:

RENAME 1 FOO;	assume that file NEWF has been opened to internal file number 1; this command will change the name of that file from NEWF to FOO.

REDUCE C1 D1;

This command reduces the length of a file. The file which is to be shortened must be open and associated with internal file number C1.

D1 is the number of segments to be removed from the end of the file.

If D1 is greater than or equal to the number of segments in the file, the file is deleted from the directory. Example:

REDUCE 2 2;	assume file TT13 is opened to internal file number 2; this command then removes 2 segments from its end and returns those segments to free storage.

EXTEND C1 D1;

This command extends the length of a file. The file which is to be lengthened must be open and associated with internal file number C1.

D1 is the number of segments to be added to the end of the file.

If there is space available, Monitor will lengthen the file as requested.

PROTECT C1 C2;

This command changes the file protection mask of a file. The file to be protected must be open and associated with internal file number C1.

C2 is the new file protection mask.

For file protection, the 12-bit account number is partitioned into a project number (high order 7 bits) and a programmer number (low order 5 bits). Examples:

C2 = 1	read protect against users whose project number differs from owner's

C2 = 2 write protect against users whose project number differs from owner's

C2 = 4 read protect against users whose project number is same as owner's

C2 = 10 write protect against users whose project number is same as owner's

C2 can be the sum of any of the above values.

This command is illegal for all users except the owner. Example:

PROTECT 1 3; read and write protect internal file 1 against access by any user whose project number differs from the owner's.

F C1; This command causes Monitor to print out the current state of the association of the user's internal file number C1 with the file. The response format is

C1 S1 C2 D1

where

C1 is the owner's account number
S1 is the file name
C2 is the protection mask
D1 is the number of segments

3.5 CONTROL OF USER PROGRAMS

START C1; This command begins execution of a user program at location C1. In addition, the command resets the Switchboard so that all characters typed from either a keyboard or program directed into the user program's input buffer are no longer intercepted by the System Interpreter. The accumulator (AC) and link are cleared and the user's interrupt system is turned off.

START; This command restarts a user program. If Monitor has been called during the execution of the user program, the complete state of the program is saved including the location of the next instruction to be executed. When the START command is given, the program's state is restored and the program continues execution where it left off. As in the START C1; command, characters intended for the user program's input buffer are no longer intercepted by the System Interpreter.

DEPOSIT C1 C2 ... Cn; This command stores C2 in location C1, C3 in location C1+1, ..., Cn in location C1+n-1.

n is equal to or less than 10 decimal.

A user can load a binary user program using the DEPOSIT command, although it is much easier using DDT-8. This command is useful when making small patches to stored programs.

EXAMINE C1 D1; This command causes Monitor to type the contents of the D1 locations starting at location C1.

D1 is equal to or less than 10 decimal.

If D1 is not specified, the contents of location C1 is typed.

3.5.1 Saving and Restoring Binary User Programs

The SAVE, LOAD, R and RUN commands leave file S1 open on internal file 3, turn the user's interrupt system off, and reset the Wait Mask to its initial value.

SAVE S1;
SAVE S1 C1;
SAVE S1 C1 C2;
SAVE S1 C1 C2 C3; These commands write portions of the user's core image onto a file whose name is S1.

- C1 is the file address of the first word to be written; if not specified, the entire 4K is written on the first 4096 words of the file.
- C2 is the core address of the first word to be written; if not specified, all 4096 words are written.
- C3 is the core address of the last word to be written; if not specified, 7777 is assumed.

Example:

SAVE NEWF; writes core words 0 through 7777 on words 0 through 7777 of file NEWF.

LOAD C1 S1;
LOAD C1 S1 C2;
LOAD C1 S1 C2 C3;
LOAD C1 S1 C2 C3 C4; These commands read certain portions of the file whose owner's account number is C1, and whose file name is S1 into core.

- C2 is the file address of the first word to be read; if not specified, words 0 through 7777 are read into words 0 through 7777.
- C3 is the core address of the first word to be loaded; if not specified, all 4096 words are loaded.
- C4 is the core address of the last word to be loaded; if not specified, 7777 is assumed.

Example:

LOAD NEWF 5 10 17; loads words 5 through 14 into words 10 through 17 respectively.

R S1; This command is equivalent to

OPEN 3 S1 2; LOAD 2 S1; START 0

which loads program S1 from the System Library (account 2) and starts the program running. The accumulator (AC) and link are cleared.

RUN S1;

This command is equivalent to

OPEN 3 S1; LOAD S1; START 0

The accumulator (AC) and link are cleared.

RUN C1 S1;

This command is equivalent to

OPEN 3 S1 C1; LOAD C1 S1; START 0

The accumulator (AC) and link are cleared.

S;

This command stops the execution of the user program, saves its complete state, and sets the Switchboard so that all characters directed to the program's input buffer will be intercepted by System Interpreter.

WHERE;

This command causes Monitor to type out the current state of a user program's location counter, accumulator, link, and switch register.

USER;

This command causes Monitor to type out the number of the job and devices owned by the user.

USER C1;

This command causes Monitor to type out the numbers of the devices owned by user C1. If job 0 is specified, the numbers of unassigned devices are typed.

SWITCH C1;

This command sets the user's switch register to C1.

3.6 SWITCHBOARD

Using Monitor commands the user may set, reset, and read any given bit in either the Permission Table or the Switchboard Table. See Section 2.2.1 Character Transmission and Section 2.3 Permission and Switchboard Tables for a description of the meaning of bits in these tables.

SET L1 L2 C1 L3 C2;

This command sets a bit in either the Permission or Switchboard Table to a one.

L1 =	P	denotes Permission Table
	S	denotes Switchboard Table
L2 =	K	denotes keyboard
	P	denotes user program
C1		is the octal number of the keyboard or program
L3 =	I	denotes input buffer
	O	denotes output buffer
C2		is the octal number of the buffer which is to receive the characters from the character source.

Monitor checks for appropriate ownership to decide whether it will allow the connection to be made. The ownership of the character source is used to determine legality in setting the Switchboard Table while the ownership of the character sink is used in setting the Permission Table. Examples:

SET P K 13 0 12; requests that keyboard number 13 be given permission to write on the printer (output buffer) of console 12.

SET S P 4 I 16; informs Monitor that job number 4 would like to type out characters into the input buffer of user program 16. This request will be granted only if user 16 has previously given permission by SET P P 4 I 16; or by the equivalent IOT instruction (SSP).

RESET L1 C2 C1 L3 C2; This command is identical in all respects to the SET command above except that the bit indicated by the parameters will be set to zero instead of one.

READ L1 L2 C1 L3 C2; This command uses the same parameters to specify a bit in either the Permission or Switchboard Table as in the SET and RESET commands above. Monitor will type out the value of the bit.

DUPLEX; This command is a shorthand command to set the Switchboard Table so that characters typed on the keyboard from which this command is issued will appear on the printer of that console. Example:

DUPLEX; if the keyboard issuing this command is number 16, then this command is equivalent to SET S K 16 0 16;

ALLOW C1; This command is shorthand to indicate that a user is given permission for keyboard number C1 to place characters in the output buffer of his console. Example:

ALLOW 4 assume that this command was issued from keyboard number 27, then it is equivalent to SET P K 4 0 27;

LINK C1; This command is given by a user who wishes to communicate with a console other than the one at which he is setting. This command is legal only if the owner of the console has set the Permission Table so that he will accept characters from the requesting console into his printer's output buffer. The command is shorthand for the command ALLOW C1; followed by the command to set the Switchboard so that characters from the requesting console will be placed in output buffer number C1. Example:

LINK 32; assume that the console which issued this command was number 7, then this command would be legal only if the owner of console number 32 had previously set the Permission Table to allow keyboard 7 access to that output buffer. He could have done this with ALLOW 7; or by SET P K 7 0 32; or by

an equivalent IOT instruction executed by his program. If that permission has been granted, the LINK command is equivalent to the following two Switchboard commands: SET P K 32 0 7; (ALLOW 32) and SET S K 7 0 32;

3.7 INTER- SYSTEM COMMUNICATION

In those TSS/8 systems having a local connection to a PDP-10 Time-Sharing System or a Synchronous Data Communication System (Type 637), the input to the user's input buffer and program output is scanned for the character sequences CTRL/B CTRL/X and CTRL/B CTRL/Y, respectively. All characters up to the next CTRL/B are diverted to the PDP-10 or 637 System, whichever the case may be. Characters from the PDP-10 and 637 System are directed into the user's input buffer.

CHAPTER 4
INPUT/OUTPUT TRANSFER INSTRUCTIONS

Whenever a user program executes an input/output transfer (IOT) instruction (an instruction of the form 6XXX) the system traps the instruction and transfers control to a system service or simulation routine. These routines accomplish special tasks, set up parameters for the system, and perform input/output for the user program.

IOT instructions may be separated by function into three types:

a. Input/output instructions available on the standard PDP-8/I without a time-sharing monitor - when a user program executes one of these IOTs, TSS/8 simulates an input/output function similar to the function of the IOT instruction on the standard PDP-8/I. Some standard PDP-8/I IOT instructions are illegal in TSS8 (see Appendix C for a summary of legal IOTs).

b. IOT instructions to request input/output service from TSS/8 unavailable on the standard PDP-8/I - these include requests for DECdisk, DECtape, high-speed paper tape reader and punch, card reader, and console character handling.

c. IOT instructions which call subroutines to set user parameters or to alter the time-sharing environment for a particular user program - an alteration may, for example, include requests to add or release facilites or to change mode of character handling.

An IOT instruction usually acts as a subroutine call. Therefore, depending upon the specific IOT instruction, parameters may be loaded into the accumulator (AC) before execution of the IOT. In some cases, the parameter in the AC acts as a pointer to a parameter block in the user's program. If the system has to return information to the user's program, it returns that information in the AC or in a block of locations in the user's program (the beginning address of this block is in the AC).

4.1 PROGRAM CONTROL

Check Status (CKS) Octal Code: 6200

Operation: There are three status registers, STR0, STR1, and STR2. All three registers are read by the CKS instruction, and the accumulator (AC) is cleared. Before executing CKS, load the AC with the address of a 3-word block. Executing CKS will store in

Word 1: STR0
Word 2: STR1
Word 3: STR2

The formats of these registers are:

STR0 Bits

0	Run Bit	Unconditional run (this bit is turned on when the program is started by the System Interpreter)
1	PI Enable	ION has been executed by user program
2	Source	Input buffer source is being recorded

STR0 Bits

3	Time	Input buffer time is being recorded
4	Binary	Card reader mode: alphanumeric (0), binary (1)
5	JSIOT	Non-resident IOT call (system use only)
6	JSIOTC	Copy IOT results to user (system use only)
7 through 11	Error Code	Bits 7 through 11 specify a system error code

STR1 Bits

0	Timer	Time is up
1	File 0	Internal file 0 is not busy
2	File 1	Internal file 1 is not busy
3	File 2	Internal file 2 is not busy
4	File 3	Internal file 3 is not busy
5	Delimiter	There is a delimiter in the input buffer
6	Keyboard	There is a character in the input buffer
7	Teleprinter	Output buffer is not full
8	Reader	Character in reader buffer
9	Punch	Punch buffer is not full
10	Error	System error detected, code in bits 7 through 11 of STR0
11	Wait	Job is not waiting

STR2 Bits

0	Plotter	Plotter flag
1	DT0 Flag	DECtape flag unit 0
2	DT0 Error	DECtape error flag unit 0
3	DT1 Flag	DECtape flag unit 1
4	DT1 Error	DECtape error flag unit 1
5		Unused
6	Card Reader	Character in card reader buffer
7	CR No Ready	Card reader not ready
8	CR End File	Card reader end of file
9	LP Error	Line printer error
10	LP Done	Line printer done
11		Unused

Set Interrupt Mask (SIM) Octal Code: 6000

Operation: TSS/8 simulates the PDP-8 program interrupt system for the user. A user program may request this mode of operation by executing the ION instruction. A simulated interrupt will cause an immediate transfer to location 1 of the user program. The user's current location (Program Counter) before the interrupt, is stored in location 0 and the interrupt system

is disabled (bit 1 of STR0 = 0). If the instruction following the ION is a JMP or JMS instruction, it is simulated by the system before returning control to the user's program.

The SIM instruction enables the user program to set those conditions which will result in simulated program interrupts. The system provides a mask called the Program Interrupt Mask, which has the same format as the two status registers STR1 and STR2 (see CKS).

Before executing the SIM instruction, load the AC with the address of a 2-word block containing the mask. After executing the SIM instruction, a bit match between the status registers and the Program Interrupt Mask will result in a simulated interrupt if the interrupt system is enabled. The user's program may do a CKS instruction to determine the cause of the interrupt.

The Program Interrupt Mask is normally set to

0074
7677

and is reset only on LOGOUT. The AC is cleared by SIM.

Read Interrupt Mask (RIM) Octal Code: 6003

Operation: The Program Interrupt Mask is read into two locations starting at the location whose address is in the AC, and the AC is cleared.

Interrupt Turn On (ION) Octal Code: 6001

Operation: This instruction enables TSS/8 to respond to a user program interrupt request by setting bit 1 of STR0 = 1. If the user interrupt is disabled (bit 1 of STR0 = 0) when this instruction is given and the following instruction is a JMP or JMS instruction, TSS/8 executes it, then enables the interrupt by returning control to the user program. This instruction has no effect if given when the user interrupt is enabled.

Interrupt Turn Off (IOF) Octal Code: 6002

Operation: This instruction disables the user program interrupt system by setting bit 1 of STR0 = 0.

Set Wait Mask (SWM) Octal Code: 6005

Operation: The user program may request to be dismissed until some condition has been met. This is done by setting a mask through which the system will observe the status register of the user program. Load the AC with the address of a 2-word block containing 1 bits in the desired

wait conditions. The two words are inclusively ORed with the wait mask and the AC is cleared. In general, flag test IOTs whose skip condition is not met and requested I/O operations which cannot be performed immediately, cause the program to be dismissed until the condition is met.

For example, suppose it is desired that the user program be dismissed until any character is typed, then

```
        TAD MASK
        SWM
        KSF
         .
         .
         .
MASK,   0040
        0000
```

would have the desired effect.

The normal setting of the Wait Mask is

```
0001
7767
```

and is reset on LOGIN, SAVE, LOAD, R and RUN commands.

Clear Wait Mask (CWM) Octal Code: 6006

Operation: Same as Set Wait Mask (SWM) except that the selected bits are cleared and the AC is cleared.

Read Wait Mask (RWM) Octal Code: 6007

Operation: The contents of the user's Wait Mask are read into two successive locations beginning at the word whose address is in the AC, and the AC is cleared.

Wait (WAIT) Octal Code: 6410

Operation: The user's program is dismissed until it is re-scheduled by some status bit/Wait Mask match. The user may read the status registers to determine the restart conditions.

User (USE) Octal Code: 6421

Operation: Return in the AC the number of the current job.

Console (CON) Octal Code: 6422

Operation: Return in the AC the smallest unit number of the unslaved consoles assigned to the job whose number is in the AC. If the AC is 0, the number of an unassigned console is returned; if it does not exist, -1 is returned.

User Run Time (URT) Octal Code: 6411

Operation: Returns the total run time of the user program in the two locations starting at the location whose address is in the AC. The AC is cleared.

Time-Of-Day (TOD) Octal Code: 6412

Operation: Returns the value of the System Clock in the two locations starting at the location whose address is in the AC. The AC is cleared.

Return Clock Rate (RCR) Octal Code: 6413

Operation: The number of milliseconds per clock tick is returned in the AC.

Date (DATE) Octal Code: 6414

Operation: Returns the date in the AC. The format of this 12-bit number is

DATE=((YEAR-1964)*12*(MONTH-1))*31+DAY-1

Skip On TSS/8 (TSS) Octal Code: 6420

Operation: This instruction is used by programs which run under both TSS/8 and a standard PDP-8/I. Under TSS/8, the instruction following TSS will be skipped. On a standard PDP-8/I, the IOT has the effect of a NOP instruction.

Halt (HLT) Octal Code: 7402

Operation: This instruction is used to stop the user program and to pass control to the System Interpreter. Executing HLT is exactly equivalent to typing CTRL/B S RETURN. HLT may be micro-coded with other group 2 IOT's (see Small Computer Handbook, C-800).

Quantum Synchronization (SYN) Octal Code: 6415

Operation: Upon execution of this instruction, the system will dismiss the user program and set it in the run state so that it will be run again next time through the "round-robin." Ordinarily, this instruction is used to insure a full time quantum to perform some critical operation.

Set Timer (STM) Octal Code: 6416

Operation: The system provides a clock time for each user program. By means of this IOT, the timer may be set to "fire" after a specified number of clock ticks have elapsed.

Load the AC with the time in ticks to prime the timer. Upon execution of the STM instruction, the system sets the timer to "fire" in the specified number of ticks, and turns the timer bit (bit 0) in status register 1 to 0 and clears the AC. After the specified time has elapsed, the system turns bit 0 back to 1.

The timer mechanism may be used three ways: 1) The user may set the time going and use the CKS IOT to find out when the time is up 2) The timer may be used with the Wait feature.

```
        TAD C5000       /Prime timer with 5000(8) ticks
        STM
        TAD MASK        /Set the Wait Mask to check bit 0
        SWM             /of status register 1
          .
          .
          .
MASK,   4000
        0000
C5000,  5000
```

Here the user program will be dismissed until 5000 ticks have elapsed 3) The time may also be used in conjunction with the interrupt system. In this case, an interrupt will occur after the specified number of ticks.

Assign Device (ASD) Octal Code: 6440

Operation: If the device specified by the contents of the AC is available, it will be assigned to the user program and the AC cleared. Otherwise, the number of the user program having the device is placed in the AC. If the device does not exist, 7777 is returned in the AC.

The left half of the AC specifies the device type and the right half specifies the unit number.

AC Bits 0-5	AC Bits 6-11	Device
0	K	Console K
40	0	Paper tape reader
40	1	Paper tape punch
40	2	Line printer
40	3	Plotter
40	4	Card Reader
40	N+5	DECtape unit N

This assignment is in effect until a corresponding REL instruction or LOGOUT is encountered.

Release Device (REL) Octal Code: 6442

Operation: The device specified by the contents of the AC is released. If the device was not assigned by ASD, the release is a NOP. Otherwise, the device is made available to other users. The AC is cleared.

Slave A Console (SLV) Octal Code: 6402

Operation: SLV enables a user program to slave a console belonging to the user. When a console is slaved, the call (CTRL/B) character is treated as an ordinary character rather than as the System Interpreter call character. This means that the slaved console may not call the System Interpreter.

Load the AC with the number of the console to be slaved, then execute SLV. If the console specified does not belong to the user, this IOT will be considered illegal and the AC will be non-zero. Otherwise, the AC is cleared.

Unslave A Console (UNS) Octal Code: 6403

Operation: UNS removes a slaved console from the slave state. The call character resumes its power to call the System Interpreter.

Load the AC with the number of the console to be released from the slave state, then execute UNS. If the console specified is not a slave or does not belong to the user, this IOT will be considered a NOP. The AC is cleared.

Set Switchboard and Permission Tables (SSP) Octal Code: 6404

Operation: SSP is used to set, reset, and read bits in the Switchboard and Permission Tables. Its use mirrors the format of the Monitor commands for the same purpose.

Load the AC with the beginning address of a 3-word block:

Word 1: Bits 0 and 1 contain 0, 1, 2, or 3

- 0 reset the bit in the table
- 1 read the bit in the table - the value of the bit is returned in the AC after executing SSP
- 2 set the bit in the table
- 3 link - set both tables

Bit 9 contains 0 or 1

- 0 reference the Switchboard Table
- 1 reference the Permission Table

Bit 10 contains 0 or 1

- 0 reference the keyboard rows of the table
- 1 reference the programs typing out rows of the table

Bit 11 contains 0 or 1

- 0 reference the input columns of the table
- 1 reference the output columns of the table

Word 2: contains the number of the keyboard or program typing out to be referenced

Word 3: contains the number of the input or output buffer to be referenced

If SSP is executed, the AC is set to 0, otherwise, to 7777. As an example of the use of SSP to duplicate the effect of the Monitor command SET P K 13 0 7; (set the Permission Table to give permission for the keyboard of console 13 to type into the output buffer of console 7) execute the following sequence:

```
        TAD ARRAY
        SSP
         .
         .
         .
ARRAY,  .+1
        4005
        13
        7
```

As an example of console linking, the Monitor command LINK 3; (set the Permission Table to give permission for the keyboard of console 3 to type into the output buffer of this job and set the Switchboard Table to route keyboard input to this job to the output buffer of console 3) is equivalent to:

```
        USE
        CON
        DCA TABLE+2
        TAD TABLE
        SSP
         .
         .
         .
TABLE,  .+1
        6000
        0
        3
```

Set Switch Register (SSW) Octal Code: 6430

Operation: Each user program has a switch register corresponding to the PDP-8/I switch register. This IOT stores the contents of the AC in the user's switch register, and the AC is cleared.

OR With Switch Register (OSR) Octal Code: 7404

Operation: The contents of the user's switch register are inclusively ORed into the AC. OSR may be micro-coded with other group 2 IOT's (see Small Computer Handbook, C-800).

4.2 FILE CONTROL

All bulk data storage for user programs is on file on the disk. This area is referred to collectively as file storage. A user program may read or write a file to handle bulk input/output of text or binary information.

Files are composed of sequential segments of file storage. The size of a file segment is an integer multiple of disk records specified at system initialization time. A record consists of 128 12-bit machine words. Files are at least one segment long and grow by adding additional segments to the end of the file. Each file is registered in the owner's file directory by a name which may be any string of not more than six 6-bit characters.

Each user may refer to four files simultaneously. An internal file number (0, 1, 2, or 3) is associated with each of the user program's active files. All references to file storage is by internal file numbers. The act of associating an internal file number with a file referenced by an account number and name is called "opening" the file.

All file IOTs that are successfully completed return to the user with the AC cleared. A non-zero AC indicates that an error was detected and the IOT was not performed. The error messages are as follows:

AC = 4000	The internal file specified is not open
AC = 4400	The file is open to another user
AC = 5000	The directory is full
AC = 6000	File protection violation
AC = 7000	File not found
AC = 7400	Disk is full

Read File (RFILE) and Write File (WFILE) Octal Codes: 6603 and 6605

Operation: Once the association of a file with an internal file number has been made, these IOTs allow the actual file reference to be made. They are illegal on a file that has not been opened (associated with an internal file number).

To read or write a file, load the AC with the address of a 6-word block, then execute the IOT. The format for the 6-word block is:

Word 1:	contains the high-order file word address
Word 2:	contains the internal file number
Word 3:	contains the negative of the number of words for the operation. This number will be either the number of words to be read or the number of words to be written.
Word 4:	contains a pointer to the beginning address -1 of a buffer located in the user program. On a read operation this buffer will receive the information from the file; on a write operation this buffer holds the information that is to be sent to the file.
Word 5:	contains the least significant 12 bits of the initial file word address to begin the operation.
Word 6:	contains an error code:

0 if no error
1 if parity error
2 if file shorter than word count
3 if file not open
4 if protection violated

The read or write begins at the word specified by words 1 and 5. For example:

```
        TAD X
        WFILE
          .
          .
          .
X,      .+1
        0
        1
        -200
        6477
        200
```

means, write 200 words starting at word 200 of the file that is associated with internal file number 1 from a buffer starting at location 6500.

After the IOT is given, the file request is placed in the file request queue, and the corresponding file-inactive bit is cleared in status register 1 for the user program. Execution of the user program continues immediately without waiting for the file request to be completed. If the Wait Mask contains a 1 in the bit corresponding to the inactive bit for that file, however, the user program will be dismissed until the file request has been completed.

The file queue is one deep for each of the user program's internal file numbers, therefore, four different file requests may be overlapped if they refer to the four different internal file numbers. An attempt to overlap a file request on the same internal file number will result in the user program being dismissed until the first request on that internal file number has been completed. After completion of any file request, the corresponding file inactive bit in status register 1 is set to 1 and the word count (word 3) and core address (word 4) are updated, and the error code is set in word 6.

Rename A File (REN) Octal Code: 6600

Operation: REN is used to change the name of a file. Load the AC with the beginning address of a 4-word block, where

Word 1:	contains the internal file number associated with the file whose name is to be changed, that is, to rename a file one must first open it, otherwise, REN is illegal.
Words 2 through 4:	contain the new name. This name is in 6-bit characters packed two to a word.

Open A File (OPEN) Octal Code: 6601

Operation: OPEN is used to associate a file with an internal file number. This process, called "opening", is necessary since all file operations are in terms of the internal file numbers. Before executing the OPEN IOT, load the AC with the beginning address of a 5-word block, where

Word 1:	contains the internal file number.
Word 2:	contains the account number of the owner of the file. If 0, the account number of the current user is specified.
Word 3 through 5:	contain the name of the file to be opened. This name is in 6-bit characters packed two to a word.

If there was another file associated with the internal file number before the execution of the OPEN IOT, it will be closed automatically before the new file is associated with the internal file number.

Close A File (CLOS) Octal Code: 6602

Operation: CLOS terminates the association between files and their internal file numbers. Before executing CLOS, load the AC with a selection pattern for the internal file numbers whose associated files are to be closed. The file is closed if bit I is 1, where I = bit 0, 1, 2, or 3.

Protect A File (PROT) Octal Code: 6604

Operation: The owner of a file may protect his file from unauthorized attempts to access it by using this instruction. Before executing PROT, load the AC with

Bits 5 through 6	Internal file number of the reserved file to be protected.
Bit 7	Write protect against owner.
Bit 8	Write protect against users whose project number is same as owner's.
Bit 9	Read protect against users whose project number is same as owner's.
Bit 10	Write protect against users whose project number differs from owner's.
Bit 11	Read protect against users whose project number differs from owner's.

A file must be opened before it can be protected. PROT is legal only when performed by the file owner, that is, the user who created the file. All attempts to access the file which violate any of the protection flags will be considered illegal.

File Information (FINF) Octal Code: 6613

Operation: FINF enables a user program to determine what file, if any, is associated with an internal file number. Load the AC with the beginning address of a 7-word block, where

Word 1:	contains the internal file number for which the user program wishes information.

Words 2 through 7 will contain the information that the system returns after executing FINF.

Word 2:	contains the account number of the owner or zero if no file is associated with the internal file number, that is, the file is not open.
Words 3 through 5:	contain the name of the file in 6-bit code.

Word 6: contains

Bit=1	Means
7	write protected against owner
8	write protected against users whose project number is same as owner's
9	read protected against users whose project number is same as owner's
10	write protected against users whose project number differs from owner's
11	read protected against users whose project number differs from owner's

Word 7: contains the number of segments that the file contains.

Create A File (CRF) Octal Code: 6610

Operation: The user may request the system to create a new file of one segment. The user program provides the new name for the file. Load the AC with the beginning address of a 3-word block, where

Words 1 through 3: contain the 6-character name.

If there is some reason why the request cannot be granted, the system will return a non-zero error code in the AC. The protection of a newly created file is 17.

Extend A File (EXT) Octal Code: 6611

Operation: To extend the length of an existing file, that file must be currently open. Load the AC with the beginning address of a 2-word block, where

Word 1: contains the internal file number of the file to be extended.

Word 2: contains the number of segments the system should append to the file.

If for some reason the request to extend a file cannot be granted, the AC will contain 4000, 4400, 6000, or the number of segments it failed to append.

Reduce A File (RED) Octal Code: 6612

Operation: To reduce the length of an existing file, that file must be currently open. Load the AC with the beginning address of a 2-word block, where

Word 1: contains the internal file number of the file to be reduced.

Word 2: contains the number of segments to be removed.

This request will always be granted.

Segment Size (SIZE) Octal Code: 6614

Operation: The number of words per segment is returned in the AC.

Account (ACT) Octal Code: 6617

Operation: The account number of the job whose number is in the AC is returned in the AC. If the AC is 0, the current job is assumed. If 0 is returned in the AC, the specified job does not exist.

Who (WHO) Octal Code: 6616

Operation: The account number and password of the current job are returned to the three-word block whose address is in the AC, and the AC is cleared.

4.3 INPUT BUFFER CONTROL

Set Keyboard Break (KSB) Octal Code: 6400

Operation: Rather than activate the user program to receive each character as it is placed in the input buffer, TSS/8 saves time by activating the program only on certain conditions specified by the user. These activation conditions are called delimiters. Before executing KSB, load the AC with a 12-bit delimiter table. The format of the table is as follows: (where each bit = 1)

Bit	Specifies				
0	change of character source				
1	all letters				
2	all numerals				
3	space	horizontal tab			
4	line feed	vertical tab			
	form feed	carriage return			
5	!	"	'	,	
	;	:	?		
6	[	]	(	)	
7	&	*	-	.	↑
	/	<	=	>	←
8	@	\	#	$	%
9	RUBOUT				
10	ALT MODE				
11	all characters not specified in bits 1 through 10				

When a bit in the table is set to 1, the user's program is activated if the condition the bit represents is met by the incoming character, or if the input buffer becomes full. The delimiter table is initialized to 3777 on LOGIN and ASSIGN. KSB clears the AC.

Set Buffer Control Flags (SBC) Octal Code: 6401

Operation: SBC permits the user program to clear his input and output buffers and control the recording of character time and source in his input buffer. Before executing SBC, load the AC with the desired flags whose functions are

Bit	
0	Clear output buffer
1	Clear input buffer
2	Record character source
3	Record character time

If bit 2=1, the source of each character in the input buffer is recorded in the following character position. The source from which the character was generated is indicated by the 8-bit character as follows:

0XX	Console XX keyboard
1XX	User program XX typing out
172	637 typing out
174	PDP-10 typing out
177	System Interpreter typing out

Bit 2 of status register 0 is set to 1. If AC bit 2=0, source recording in the input buffer is suppressed and bit 2 of status register 0 is set to 0.

If AC bit 3=1, the time of receipt of each input character is recorded in the two successive character positions following the character or following the source. If the time is being recorded, bit 3 of status register 0 is set to 1.

If AC bit 3=0, time recording in the input buffer is suppressed and bit 3 of status register 0 is set to 0. The AC is cleared by SBC.

Skip On Keyboard Flag (KSF) Octal Code: 6031

Operation: The keyboard flag is sensed, and if it contains a binary 1, the contents of the PC is incremented by one so that the next sequential instruction is skipped. The keyboard flag is bit 6 of status register 1, and it has a value of 1 whenever the input buffer is not empty. If the keyboard flag is 0 and the keyboard wait flag (bit 6 of wait register 1) is 1, the job is placed in the active state until the keyboard flag is 1. If the keyboard wait flag is 0, control returns to the calling program immediately.

Clear Keyboard Flag (KCC) Octal Code: 6032

Event Time: 2

Operation: The AC is cleared, and if the input buffer is empty, the keyboard flag is cleared.

Read Keyboard Buffer Static (KRS) Octal Code: 6034

Event Time: 3

Operation: If the input buffer is not empty, the next character from the input buffer is inclusively ORed into bits 4 through 11 of the AC. This is a static command in that neither the AC nor the keyboard flag is cleared. If fhe input buffer is empty and the keyboard wait flag is 0, no operation is performed and control returns immediately to the user's program. If the keyboard wait flag is 1, the job is placed in the active state until the keyboard flag is 1.

Read Keyboard Buffer Dynamic (KRB) Octal Code: 6036

Event Time: 2, 3

Operation: Identical to KCC followed by KRS.

Read Keyboard String (KSR) Octal Code: 6030

Operation: Before executing KSR, load the AC with the address of a 2-word block, where

Word 1: contains the negative of the number of characters to read.

Word 2: contains the address -1 of the first word of the buffer.

If the keyboard wait flag is 1, control does not return to the user program until a break condition (see KSB) is satisfied or until the word count in word 1 is reduced to zero. If the keyboard wait flag is 0, control returns to the user program immediately; control returns to the location following the KSR, the negative of the number of characters remaining to be transferred is in word 1, and word 2 points to the last character stored. The AC is cleared.

4.4 OUTPUT BUFFER CONTROL

Send A String (SAS) Octal Code: 6040

Operation: To send a string, load the AC with the address of a 2-word block, where

Word 1: contains the negative of the number of characters to be sent.

Word 2: contains the address -1 of the first word of the string.

The characters are stored one per word right justified starting at the address specified by word 2. Upon execution of SAS, the system takes only as many characters as will fit in the output buffer. It then makes the appropriate adjustment to word 2 to indicate a new starting address and to word 1 to indicate the reduced character count; it returns to the instruction following the SAS. If the character count is reduced to zero, the instruction following the SAS is skipped. The instruction following the SAS may contain a JMP .-1 to continue the block transfer of Teletype characters. In this case, the user program will dismiss until the Teletype output buffer is nearly empty. If the user wishes to avoid dismissal, he can place a jump to another part of his program in the instruction following SAS. The AC is cleared by SAS.

Skip on Teleprinter Flag (TSF) Octal Code: 6041

Event Time: 1

Operation: The teleprinter flag is sensed, and if it contains a binary 1, the contents of the PC is incremented by one so that the next sequential instruction is skipped. The teleprinter flag is bit 7 of status register 1, and has a value of 1 if the output buffer is not full. If the teleprinter wait flag is 0, control returns to the user program immediately, otherwise, control returns when the teleprinter flag is 1.

Clear Teleprinter Flag (TCF) Octal Code: 6042

Event Time: 2

Operation: The teleprinter flag is cleared to zero.

Load Teleprinter And Print (TPC) Octal Code: 6044

Event Time: 3

Operation: The character in bits 4 through 11 of the AC is typed out and routed through the switchboard to all buffers connected to the user's program character output.

Load Teleprinter Sequence (TLS) Octal Code: 6046

Event Time: 2, 3

Operation: Same as TPC.

4.5 HIGH-SPEED PAPER TAPE READER

Skip On Reader Flag (RSF) Octal Code: 6011

Event Time: 1

Operation: The reader flag is sensed, and if it contains a binary 1, the contents of the PC is incremented by one so that the next sequential instruction is skipped. The reader flag is bit 8 of status register 1, and has a value of 1 if the reader buffer is not empty. If the reader wait flag in the wait mask is 1, the program is dismissed until the reader flag is 1, otherwise, control returns to the user program immediately.

Read Reader Buffer (RRB) Octal Code: 6012 or 6016

Event Time: 2

Operation: The contents of the reader buffer is transferred into bits 4 through 11 of the AC and the reader flag is cleared if the reader buffer is empty. This instruction does not clear the AC. If the reader buffer is empty and the reader wait flag is 1, the user program is dismissed until the reader flag is 1. If the reader buffer is empty and the reader wait flag is 0, this IOT is a NOP.

Reader Fetch Character (RFC) Octal Code: 6014

Event Time: 3

Operation: The reader flag and the Monitor reader buffer are both cleared, the reader is started to fill the Monitor reader buffer and the reader flag is set after the first character is read.

Read Reader String (RRS) Octal Code: 6010

Operation: RRS allows a user program to read a string of characters from the paper tape reader. Before executing RRS, load the AC with the beginning address of a 2-word block, where

Word 1: contains the negative of the number of characters to read.

Word 2: contains the address -1 of the first word of a buffer.

If the reader wait flag in the wait mask is 1, the program is dismissed until the transfer is completed, otherwise, control returns to the user program immediately. The reader flag is set to 1 when the transfer is complete. The AC is cleared by RRS.

4.6 HIGH-SPEED PAPER TAPE PUNCH

Skip On Punch Flag (PSF) Octal Code: 6021

Event Time: 1

Operation: The punch flag is sensed, and if it contains a binary 1, the contents of the PC is incremented by one so that the next sequential instruction is skipped. The punch flag is bit 9 of status register 1, and has a value of 1 if the punch buffer is not full. If the punch flag is 1 and the punch wait flag is 1, the program is dismissed until the punch flag is 1. If the punch flag is 0 and the punch wait flag is 0, no operation is performed and control returns to the user program immediately.

Clear Punch Flag (PCF) Octal Code: 6022

Event Time: 2

Operation: If the punch buffer is not full, the punch flag is cleared in preparation for punching the next character from the computer. If the punch buffer is full and the punch wait flag is 1, the program is dismissed until the punch flag is 1. If the punch buffer is full and the punch wait flag is 0, no operation is performed and control returns to the user program immediately.

Load Punch Buffer and Punch Character (PPC) Octal Code: 6024

Event Time: 3

Operation: An 8-bit character is transferred from bits 4 through 11 of the AC into the punch buffer and then this character is punched. PPC does not clear the punch flag. If the punch buffer is full and the punch wait flag is 1, the program is dismissed until the punch flag is 1. If the punch buffer is full and the punch wait flag is 0, no operation is performed and control returns to the user program immediately.

Load Punch Buffer Sequence (PLS) Octal Code: 6026

Event Time: 2, 3

Operation: Identical to PCF followed by PPC.

Punch String (PST) Octal Code: 6020

Operation: PST allows a user program to punch a string of characters. Before executing PST, load the AC with the beginning address of a 2-word block, where

Word 1: contains the negative of the number of characters to be punched.

Word 2: contains the beginning address -1 of the string to be punched; the characters should be right justified one per word.

Upon execution of PST, the system takes only as many characters as will fit in the punch buffer; it then makes the appropriate adjustment to word 2 to indicate a new starting address and to word 1 to indicate the reduced character count. It returns to the instruction following the PST. If the character count is reduced to zero, the instruction following PST is skipped. The AC is cleared by PST.

4.7 DECTAPE

Load Status Register A (DTXA) Octal Code: 6764

Operation: DTXA allows a user program to read and write records (128-word blocks) on DECtape. Load the AC with the beginning address of a 4-word block, where

Word 1: contains

Bit=1	
0-2	contain the transport unit select number
6-8 = 2	for read data function
4	for write data function
9 = 0	to disable DECtape control flag (DTCF) and error flag from causing a program interrupt.
1	to enable DECtape control flag (DTCF) and error flag to cause program interrupt
10-11	unused

Word 2: contains the beginning DECtape block number

Word 3: contains the negative of the number of records to transfer

Word 4: contains the beginning core address -1 of record buffer

After DTXA is given, the DECtape request is placed in the DECtape request queue and the DECtape and error flags are cleared in status register 2. Execution of the user program continues immediately without waiting for the DECtape request to be completed. If the wait

mask contains a 1 bit corresponding to the DECtape flag of the selected unit, the program will be dismissed until the DECtape request has been completed. The DECtape queue is one deep for each unit, therefore, four different DECtape requests may be overlapped if they refer to different units. An attempt to overlap a DECtape request on the same unit will result in the user program being dismissed until the first request has been completed. After the completion of any DECtape request, the corresponding DECtape flag in status register 2 is turned on. The AC is cleared by DTXA.

Skip On Flags (DTSF) Octal Code: 6771

Event Time: 1

Operation: The contents of both the error flag and the DECtape flag is sampled, and if either flag contains a binary 1, the contents of the PC is incremented by one to skip the next sequential instruction. If both flags are zero and if either of the corresponding wait flags are 1, the user program is dismissed until the skip condition is satisfied.

Read Status Register B (DTRB) Octal Code: 6772

Event Time: 2

Operation: The contents of DECtape status register B is loaded into the AC by an OR transfer. The AC bit assignments are:

0	=	error flag
1	=	mark track error
2	=	end of tape
3	=	select error
4	=	parity error
5	=	timing error
6 through 10	=	unused
11	=	DECtape flag

4.8 AUTOMATIC LINE PRINTER

The Automatic Line Printer and Control (Type 645) IOT instructions are simulated precisely as they appear in the Small Computer Handbook. The line printer error flag is bit 10 of status register 2, and the done flag is bit 11 of the same register. The wait mask and priority interrupt mask have the same significance as with other devices.

Skip On Line Printer Error (LSE) Octal Code: 6651

Event Time: 1

Operation: The contents of line printer error flag is sensed, and if it contains a binary 1, indicating that an error has been detected, the contents of the PC is incremented by one so that the next sequential instruction is skipped.

Symbol: If Line Printer Error Flag = 1, then PC+1 → PC

Clear Printer Buffer (LCB) Octal Code: 6652

Event Time: 2

Operation: Both sections of the line printer buffer are cleared in preparation for receiving new character information.

Symbol: 0 → LPB

Load Printer Buffer (LLB) Octal Code: 6654

Event Time: 3

Operation: A section of the printer buffer is loaded from the contents of bits 6 through 11 of the AC, then the AC is cleared.

Symbol: AC 6 through 11 → LPB, then 0 → AC

Skip On Line Printer Done Flag (LSD) Octal Code: 6661

Event Time: 1

Operation: The contents of the line printer done flag is sensed, and if it contains a binary 1, the contents of the PC is incremented by one so that the next sequential instruction is skipped.

Symbol: If Line Printer Done Flag = 1, then PC + 1 → PC

Clear Line Printer Flags (LCF) Octal Code: 6662

Event Time: 2

Operation: The line printer done and error flags are cleared.

Symbol: 0 → Line Printer Done Flag
0 → Line Printer Error Flag

Clear Format Register (LPR) Octal Code: 6664

Event Time: 3

Operation: The line printer format register (FR) is cleared and then loaded from the contents of bits 9 through 11 of the AC, and the AC is cleared. The line contained in the section of the printer buffer (LPB) loaded last is printed. Paper is advanced in accordance with the selected channel of the format tape if the contents of AC8 is a 1. If AC8 is a 0, paper advance is inhibited.

4.9 INCREMENTAL PLOTTER

The Incremental Plotter and Control (Type 350B) IOT instructions are simulated precisely as they appear in the Small Computer Handbook. The plotter flag is bit 0 of status register 2. The wait mask and priority interrupt mask have the same significance as with other devices.

Skip On Plotter Flag (PLSF) Octal Code: 6501

Event Time: 1

Operation: The plotter flag is sensed, and if it contains a 1 the contents of the PC is incremented by one so that the next sequential instruction is skipped.

Symbol: If Plotter Flag = 1, then PC + 1 → PC

Clear Plotter Flag (PLCF) Octal Code: 6502

Event Time: 2

Operation: The plotter flag is cleared in preparation for issuing a plotter operation command.

Symbol: 0 → Plotter Flag

Pen Up (PLPU) Octal Code: 6504

Event Time: 3

Operation: The plotter pen is raised from the surface of the paper.

Pen Right (PLPR) Octal Code: 6511

Event Time: 1

Operation: The plotter pen is moved to the right in either the raised or lowered position.

Drum Up (PLDU) Octal Code: 6512

Event Time: 2

Operation: The plotter paper drum is moved upward. This instruction can be combined with the PLPR and PLDD instructions.

Drum Down (PLDD) Octal Code: 6514

Event Time: 3

Operation: The plotter paper drum is moved downward.

Pen Left (PLPL) Octal Code: 6521

Event Time: 1

Operation: The plotter pen is moved to the left in either the raised or lowered position.

Drum Up (PLUD) Octal Code: 6522

Event Time: 2

Operation: The plotter paper drum is moved upward. This instruction is similar to PLDU except that it can be combined with PLPL or PLPD instructions.

Pen Down (PLPD) Octal Code: 6524

Event Time: 3

Operation: The plotter pen is lowered to the surface of the paper.

4.10 CARD READER

Skip on Card Reader Flag (CRSF) Octal Code: 6632

Event Time: 2

Operation: The contents of the card reader flag is sensed, and if it contains a 1 (indicating that the card column is present for reading), the contents of the PC is incremented by one so that the next sequential instruction is skipped. The card reader flag is bit 6 of status register 2 and it has the value 1 if a card is in the card buffer. If the card reader flag is 0 and the card reader wait flag is 1, the program is dismissed until the card reader flag is 1; otherwise, control returns to the user program immediately.

Read Card Equipment Status (CERS) Octal Code: 6634

Event Time: 3

Operation: The contents of the card reader flag and status flags are transferred into the contents of bits 7 through 9 of the AC. The AC bit assignments are:

7	card reader flag (1 if card in TS8 card reader buffer)
8	not ready (covers not in place, power is off, start pushbutton has not been pressed, hopper is empty, stacker is full, a card is jammed, a validity check error has been detected, or the read circuit is defective)
9	end of file (EOF) (hopper is empty and operator has pushed EOF button)

Read Card Reader Buffer (CRRB) Octal Code: 6671

Event Time: 1

Operation: The contents of the card column buffer is transferred into the AC and the card reader flag is cleared if the card reader buffer is empty. One CRRB reads either alphanumeric or binary information. If the card reader buffer is empty and the card reader wait flag is 1, the program is dismissed until the card reader flag is 1. If the card reader buffer is empty and the card reader wait flag is 0, no operation is performed and control returns to the user program immediately.

Select Alphanumeric (CRSA) Octal Code: 6672

Event Time: 2

Operation: The card reader alphanumeric mode is selected and a card is read into the card reader buffer. Information read into the card reader buffer is in 6-bit alphanumeric form (the Hollerith code representing the decoded 12-row character in one column). If the card reader wait flag is 1, the user program is dismissed until the card reader flag is 1; otherwise, control returns to the user program immediately.

Select Binary (CRSB) Octal Code: 6674

Event Time: 3

Operation: The card reader binary mode is selected and a card is read into the card reader buffer in 12-bit binary form. If the card reader wait flag is 1, the user program is dismissed until the card reader flag is 1; otherwise, control returns to the user program immediately.

Upon instruction to read the card reader buffer, the contents of the 12-bit column is transferred into the AC. In the alphanumeric mode a 6-bit Hollerith code is transferred into bits 6 through 11 of the AC, and bits 0 through 5 of the AC are cleared. In the binary mode, the binary contents of the 12 bits (or rows) in a card column are transferred into the AC so that row X is read into AC bit 0, row Y into AC bit 1, row 0 into AC bit 2, ..., and row 9 into AC bit 11. The mode is specified by either the CRSA or CRSB instruction.

Read Card String (RCS) Octal Code: 6630

Operation: RCS allows the user program to read a string of characters from the card reader. Before executing RCS, load the AC with the beginning address of a 2-word block, where

Word 1: contains the negative of the number of columns to be read.

Word 2: contains the address -1 of the first word of a buffer.

If the transfer is terminated before the column count has been reduced to zero, the card reader flag is turned on the words 2 and 1 contain the address -1 of the next buffer location and the negative of the number of columns remaining to be read, respectively. Columns are packed right justified, one per word independent of mode.

CHAPTER 5
ERROR DIAGNOSTICS

5.1 USER PROGRAM

The following error messages are typed on the user's printer by the Error Phantom when error conditions occur in a running user program. If the user program is not enabled for system error interrupts, the error messages are typed in the following format.

```
S1 FOR USER C1
AC=C2,  L=C3,  PC=C4
INSTR=C5
```

S1 is a string, describing the nature of the illegal instruction C5 which user program C1 has executed at location C4. At the time the illegal instruction was executed the value of the accumulator was C2 and the value of the link was C3. For example,

ILLEGAL IOT	The user program has executed an IOT which the system regards as illegal. The illegality may be for one of two reasons: 1. The IOT itself may be illegal. 2. The parameters to a legal IOT may invalidate it.

5.2 SYSTEM INTERPRETER

The following error messages result from illegal requests to the System Interpreter. They are printed by the System Interpreter on the printer.

S1 ?	The System Interpreter does not understand the command. S1 = command
ILLEGAL REQUEST	The user has requested an illegal service. This error usually results when some parameter has been given an incorrect value or the request refers to a facility not owned by the user.
SWITCHBOARD ERROR	The user has attempted to set a Permission Table bit not owned by him; or the user has attempted to make a connection in the Switchboard Table for which permission has not been granted.
SAVE LIMIT ERROR	The user has given invalid bounds in the command to save binary core image. The first bound must be lower than the second.
CONSOLE IN USE LOGOUT TO RELEASE	The user has tried to log in on a console which is already in use; or he has attempted to add a console to those he already owns.
LOGIN PLEASE	The user has attempted to use a console which is not logged in.

UNAUTHORIZED ACCOUNT	The user has attempted to log into the system with an invalid account number or name.
FULL	This message may appear on an attempt to log into TSS/8 from a console. It means that all disk swapping tracks are in use.
C1 HAS S1	Job C1 had device S1; the request for its acquisition cannot be granted. C1 = job number S1 = device name

APPENDIX A
TSS/8 CHARACTER SET

TSS/8 accepts 8-bit ASCII character only. ASCII is an abbreviation for USA Code for Information Interchange. The acceptable characters and their 6- and 8-bit octal equivalence are listed below.

Character	6-Bit* Octal	8-Bit Octal	Character	6-Bit* Octal	8-Bit Octal
Space	00	240	@	40	300
!	01	241	A	41	301
"	02	242	B	42	302
#	03	243	C	43	303
$	04	244	D	44	304
%	05	245	E	45	305
&	06	246	F	46	306
'	07	247	G	47	307
(	10	250	H	50	310
)	11	251	I	51	311
*	12	252	J	52	312
+	13	253	K	53	313
,	14	254	L	54	314
-	15	255	M	55	315
.	16	256	N	56	316
/	17	257	O	57	317
0	20	260	P	60	320
1	21	261	Q	61	321
2	22	262	R	62	322
3	23	263	S	63	323
4	24	264	T	64	324
5	25	265	U	65	325
6	26	266	V	66	326
7	27	267	W	67	327
8	30	270	X	70	330
9	31	271	Y	71	331
:	32	272	Z	72	332
;	33	273	[	73	333
<	34	274	\	74	334
=	35	275	]	75	335
>	36	276	↑	76	336
?	37	277	←	77	337

* Used to store passwords and filenames only

APPENDIX B
SUMMARY OF COMMANDS

A Monitor command is a string of characters terminated by a semicolon (;) or a carriage return (RETURN key). Parameters to commands may be octal numbers, decimal numbers, character strings, or single letters. In the following summary, parameters are coded as follows:

C1, C2, ...	represent octal numbers
D1, D2, ...	represent decimal numbers
S1, S2, ...	represent character strings
L1, L2, ...	represent single letters

Logging In and Out

LOGIN C1 S1;	Request to login; C1 = user's account number S1 = user's password
LOGOUT;	Request to logout; processing and console time is typed out
TIME C1;	Request typeout of processing time; C1 = job number Without C1, current job is assumed; Before logging in and without C1, time-of-day is typed out; If C1=job 0, time-of-day is typed out.

Console Manipulation

ASSIGN K C1;	Request console; K = console C1= console number
SLAVE C1;	Slave an owner console; C1 = console number
UNSLAVE C1;	Unslave an owned console; C1 = console number
RELEASE K C1;	Release console; K = console C1 = console number

Device Allocation

ASSIGN L1;	Access device; L1 = R for paper tape reader P for paper tape punch C for card reader L for line printer I for incremental plotter

Device Allocation (Cont)

ASSIGN D C1;
Access DECtape unit;
D = DECtape
C1 = device number

RELEASE L1;
RELEASE K C1;
RELEASE D C1;
Release device;
L1 = R, P, C, L, or 1, (see ASSIGN L1;)
K = console
D = DECtape unit
C1 = console or DECtape number

File

OPEN C1 S1 C2;
Establish association between internal file number and file;
C1 = internal file number
S1 = file name
C2 = account number

CLOSE S1;
Close files;
S1 = list of internal file numbers

CREATE S1;
Create new file;
S1 = name of new file

RENAME C1 S1;
Rename a file;
C1 = internal file number
S1 = new name of file

REDUCE C1 D1;
Reduce length of file;
C1 = internal file number
D1 = number of segments to be removed from end of file

EXTEND C1 D1;
Extend length of file;
C1 = internal file number
D1 = number of segments to be added to end of file

PROTECT C1 C2;
Protect a file;
C1 = internal file number
C2 = new file protection mask
1 read protect against users with different project number
2 write protect against users with different project number
4 read protect against users with same project number
10 write protect against users with same project number
Or the sum of any combination

F C1;
Print out association between internal file numbers and files
C1 = internal file number

Control Of User Programs

START C1;
Execute user program;
C1 = starting location

Control of User Programs (Cont)

START;	Restart user program;
DEPOSIT C1 C2...Cn;	Store in core memory; C1 = location C2 = contents to be stored ⋮ Cn = location C1+n-1 n ≤ 10 decimal
EXAMINE C1 D1;	List specified contents; C1 = first location D1 = number of location to be listed D1 ≤ 10 decimal

Saving and Restoring Binary User Programs

SAVE S1; SAVE S1 C1; SAVE S1 C1 C2; SAVE S1 C1 C2 C3;	Save core image; S1 = name of file C1 = file address of first word to be saved; if not specified, entire 4K is saved C2 = core address of first word to be saved; if not specified, entire core is saved C3 = core address of last word to be saved; if not specified, entire core is saved
LOAD C1 S1; LOAD C1 S1 C2; LOAD C1 S1 C2 C3; LOAD C1 S1 C2 C3 C4;	Load core image; C1 = owner's account number S1 = name of file C2 = file address of first word to be loaded; if not specified, entire 4K is loaded C3 = core address of first word to be loaded, if not specified, entire core is loaded C4 = core address of last word to be loaded; if not specified, entire core is loaded
R S1;	Run System file; S1 = name of file
RUN S1;	Run user file; S1 = name of file
RUN C1 S1;	Run user file; C1 = owner's account number S1 = name of file
S;	Stop execution
WHERE;	Type out contents of location counter, accumulator, link, and switch register
USER;	Type out number of the job and devices owned
USER C1;	Type out device numbers; C1 = user's account number

Saving and Restoring Binary User Programs (Cont)

SWITCH C1;	Set switch register; C1 = word to be set

Switchboard

SET L1 L2 C1 L3 C2;	Set bit in Permission or Switchboard Table; L1 = P for Permission Table S for Switchboard Table L2 = K for keyboard P for user program C1 = keyboard or program number L3 = I for input buffer O for output buffer C2 = receiving buffer number
RESET L1 L2 C1 L3 C2;	Clear bit in Permission or Switchboard Table; parameters are as in SET, above
READ L1 L2 C1 L3 C2;	Read bit in Permission or Switchboard Table; parameters are as in SET, above
DUPLEX;	Echo typed characters on printer;
ALLOW C1;	Permit console C1 to output on this console; C1 = console number
LINK C1;	Output to console C1; C1 = console number

APPENDIX C
SUMMARY OF IOT INSTRUCTIONS

PROGRAM CONTROL

6000	SIM	Set Interrupt Mask
6001	ION	Interrupt Turn On
6002	IOF	Interrupt Turn Off
6003	RIM	Read Interrupt Mask
6005	SWM	Set Wait Mask
6006	CWM	Clear Wait Mask
6007	RWM	Read Wait Mask
6200	CKS	Check Status
6402	SLV	Slave A Console
6403	UNS	Unslave A Console
6404	SSP	Set Switchboard and Permission Tables
6410	WAIT	Wait
6411	URT	User Run Time
6412	TOD	Time Of Day
6413	RCR	Return Clock Rate
6414	DATE	Date
6415	SYN	Quantum Synchronization
6416	STM	Set Timer
6420	TSS	Skip On TS8
6421	USE	User
6422	CON	Console
6430	SSW	Set Switch Register
6440	ASD	Assign Device
6442	REL	Release Device
7402	HLT	Halt
7404	OSR	OR With Switch Register

FILE CONTROL

6600	REN	Rename File
6601	OPEN	Open File
6602	CLOS	Close File
6603	RFILE	Read File
6604	PROT	Protect File
6605	WFILE	Write File
6610	CRF	Create File
6611	EXT	Extend File
6612	RED	Reduce File
6613	FINF	File Information
6614	SIZE	Segment Size
6617	ACT	Account Number
6616	WHO	Who

INPUT BUFFER CONTROL

6030	KSR	Read Keyboard String
6031	KSF	Skip On Keyboard Flag
6032	KCC	Clear Keyboard Flag
6034	KRS	Read Keyboard Buffer Static
6036	KRB	Read Keyboard Buffer Dynamic
6400	KSB	Set Keyboard Break
6401	SBC	Set Buffer Control Flags

OUTPUT BUFFER CONTROL

6040	SAS	Send A String
6041	TSF	Skip On Teleprinter Flag
6042	TCF	Clear Teleprinter Flag
6044	TPC	Load Teleprinter and Print
6046	TLS	Load Teleprinter Sequence

HIGH-SPEED PAPER TAPE READER AND CONTROL (TYPE PC02)

6010	RRS	Read Reader String
6011	RSF	Skip On Reader Flag
6012	RRB	Read Reader Buffer
6014	RFC	Reader Fetch Character

HIGH-SPEED PAPER TAPE PUNCH AND CONTROL (TYPE PC03)

6020	PST	Punch String
6021	PSF	Skip On Punch Flag
6022	PCF	Clear Punch Flag
6024	PPC	Load Punch Buffer and Punch Character
6026	PLS	Load Punch Buffer Sequence

DECTAPE CONTROL (TYPE TC01)

6764	DTXA	Load Status Register A
6771	DTSF	Skip On Flags
6772	DTRB	Read Status Register B

AUTOMATIC LINE PRINTER AND CONTROL (TYPE 645)

6651	LSE	Skip On Line Printer Error
6652	LCB	Clear Printer Buffer
6654	LLB	Load Printer Buffer
6661	LSD	Skip On Line Printer Done Flag
6662	LCF	Clear Line Printer Flags
6664	LPR	Clear Format Register

INCREMENTAL PLOTTER AND CONTROL (TYPE 350B)

6501	PLSF	Skip On Plotter Flag
6502	PLCF	Clear Plotter Flag
6504	PLPU	Pen Up
6511	PLPR	Pen Right
6512	PLDU	Drum Up

INCREMENTAL PLOTTER AND CONTROL (TYPE 350B) (Cont)

6514	PLDD	Drum Down
6521	PLPL	Pen Left
6522	PLUD	Drum Up
6524	PLPD	Pen Down

CARD READER AND CONTROL (TYPE 451)

6630	RCS	Read Card String
6632	CRSF	Skip On Card Reader Flag
6634	CERS	Read Card Equipment Status
6671	CRRB	Read Card Reader Buffer
6672	CRSA	Select Alphanumeric
6674	CRSB	Select Binary

APPENDIX D

REQUIRED MODIFICATIONS

The PDP-8/I or PDP-8 computer must have the following KT08/I time-sharing modifications to be used in a TSS/8 system.

a. USER FLAG REGISTER (UF) - UF is a 1-bit register that specified Monitor (UF=0) or user (UF=1) mode. In Monitor mode all instructions are legal; in user mode HALT, OSR, and IOT instructions are trapped via the program interrupt system. UF is cleared by the LOAD ADDress switch.

b. SAVE FIELD REGISTER EXTENSION (SF6) - When a program interrupt occurs, this bit is cleared and then loaded from the UF.

c. USER BUFFER REGISTER (UB) - The UB serves as a 1-bit input buffer for the UF. All transfers into the UF are made through the UB, except transfers from the computer console switches. The Change User Flag (CUF) instruction loads the UB with the value contained in the Memory Buffer (MB). The Restore Memory Field (RMF) instruction transfers the contents of SF6 into the UB to restore the UF to the condition that existed prior to a program interrupt. The UF is cleared by the LOAD ADDress switch.

The following machine instructions have been changed to operate as indicated.

a. Name and Mnemonic: CHANGE USER FLAG (CUF)

Octal Codes: 6264 and 6274

Execution Time: 1.5 μs

Operation: 6264 sets the UF to 0; 6274 sets the UF to 1. The next JMP or JMS instruction causes the appropriate mode to be entered.

Symbol: MB8 → UB

b. Name and Mnemonic: SKIP ON USER IOT (SKIOT)

Octal Code: 6245

Execution Time: 1.5 μs

Operation: HALT, OSR, and IOTs when executed with UF=1, sets the user IOT (UIOT) flag and, if the program interrupt is enabled, causes an interrupt.

c. Name and Mnemonic: CLEAR USER IOT (CIOT)

Octal Code: 6204

Execution Time: 1.5 μs

Operation: Clears the user IOT (UIOT) flag.

Symbol: 0 → UIOT

d. Name and Mnemonic: READ INTERRUPT BUFFER (RIB)

Octal Code: 6234

Execution Time: 1.5 μs

Operation: The contents of the Instruction Field, Data Field, and User Flag held in the Save Field Register during a program interrupt are transferred into bits 6 through 8, 9 through 11, and 5, respectively.

Symbols: SF0-2 → AC6-8, SF3-5 → AC9-11, SF6 → AC5

e. Name and Mnemonic: RESTORE MEMORY FIELD (RMF)

Octal Code: 6244

Execution Time: 1.5 μs

Operation: This instruction is used upon exit from the program interrupt subroutine in another field. The Instruction Field, Data Field, and User Flag that were interrupted by the subroutine are restored by transferring the contents of the Save Field Register into the Instruction Buffer and Data Field Registers and the User Buffer.

Symbols: SF0-2 → IB, SF3-5 → DF, SF6 → UB

APPENDIX E
STORAGE ALLOCATION

E.1 STORAGE MAP

A mental picture of the system's storage allocation is illustrated below.

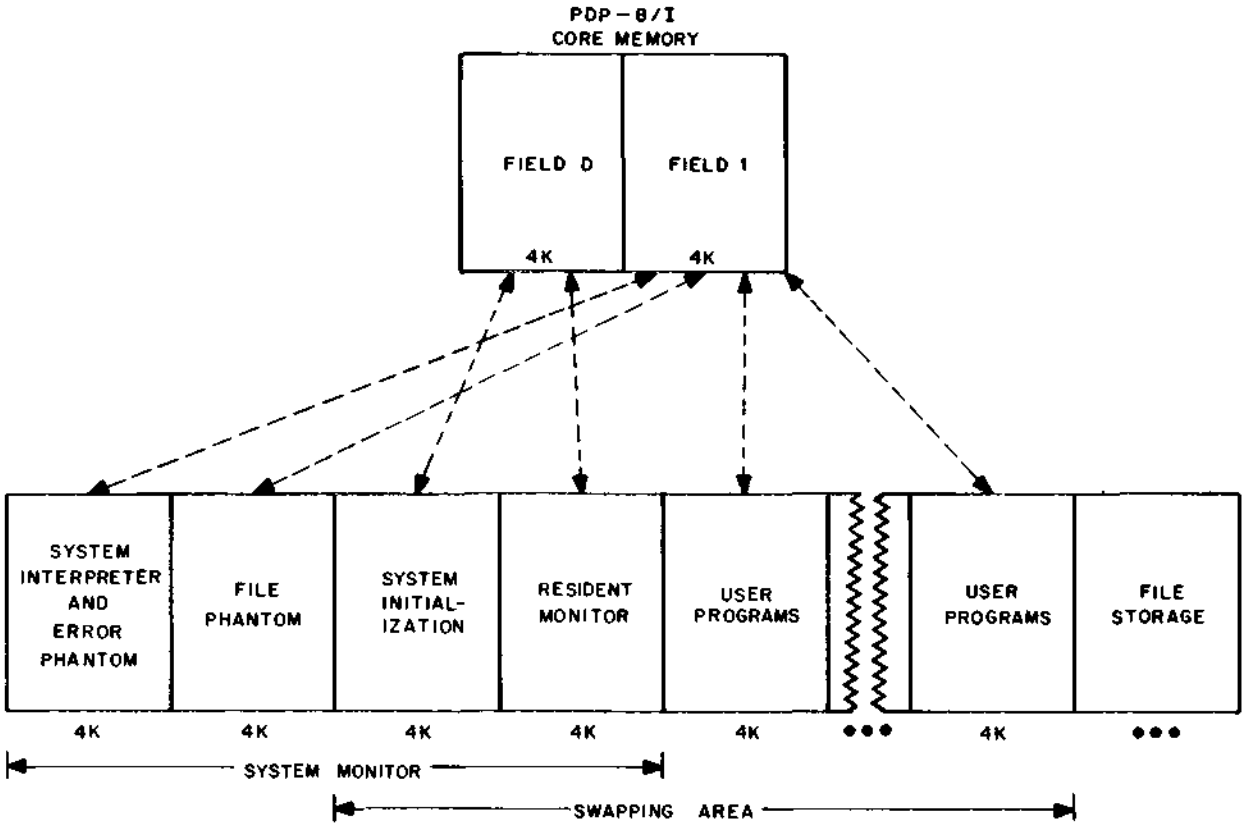

Figure E-1 TSS/8 Storage Map

E.2 FILE DIRECTORIES

There are two directories on the disk: the Master File Directory, referenced mainly by the system, and the User File Directory, referenced by the TSS/8 user. One of the functions of the MFD is to serve as a directory for the UFD. A UFD is a particular user's own file directory and will contain the names of programs he has created on the disk.

The UFD itself is a file like any other file except that its filename is a binary number in combination with a four-letter password. The leftmost seven bits of the binary combination are used

for the project number, and the remaining five bits are used for 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 UFD's of other users or the MFD; whether he is successful or not will depend upon the type of protection that has been specified for the area he is trying to reference.

An illustration of the file directories is shown below.

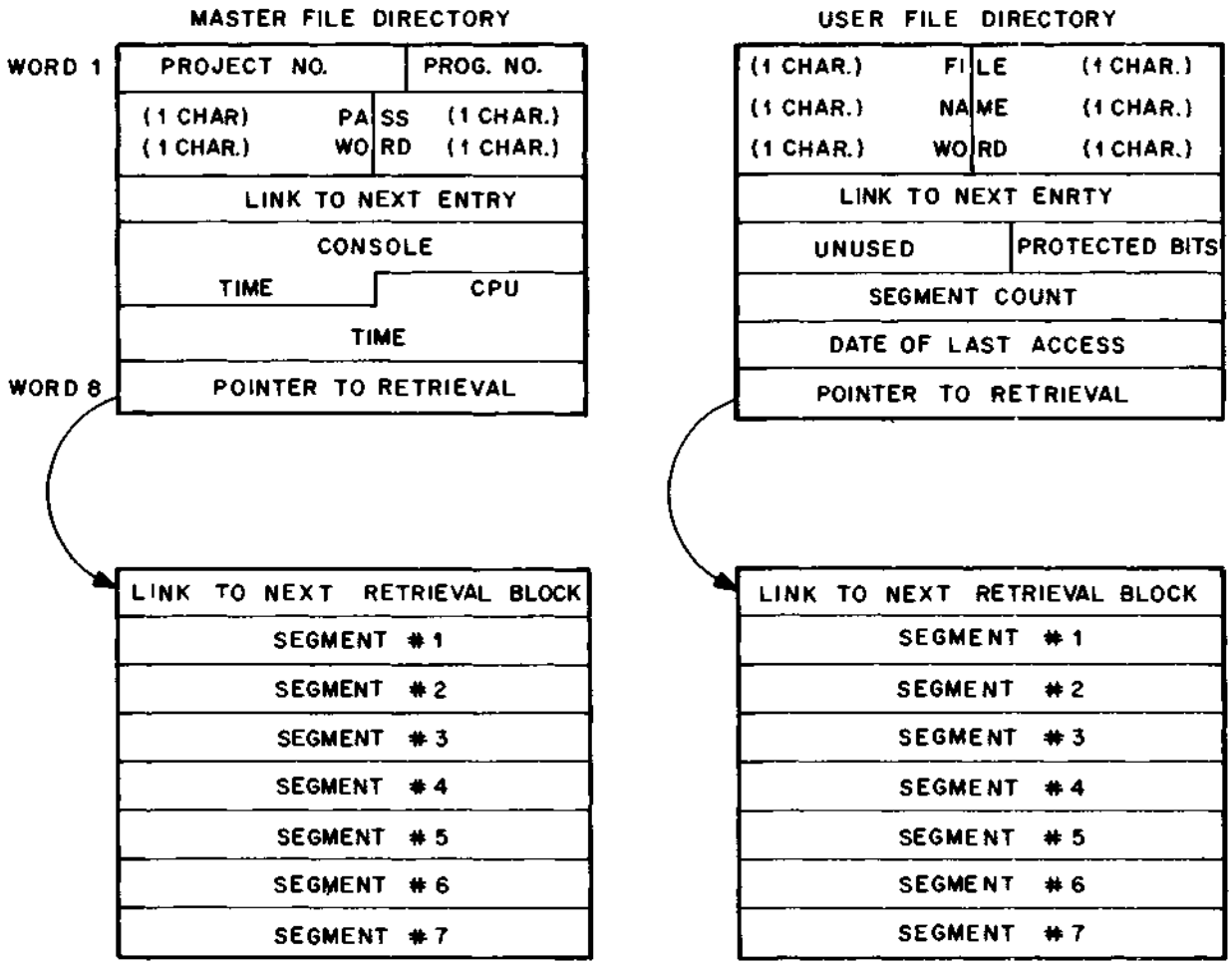

Figure E-2 File Directories