

**DECTAPE
PROGRAMMING
MANUAL**

PDP-8

PDP-8 PROGRAMMING MANUAL

DECTAPE

Copyright 1965 by Digital Equipment Corporation

PREFACE

The PDP-8 comes to the user complete with an extensive selection of system programs and routines making the full data processing capability of the new computer immediately available to each user, eliminating many commonly experienced initial programming delays.

The programs described in these abstracts come from two sources, past programming effort on the PDP-5 computer, and present and continuing programming effort on the PDP-8. Thus the PDP-8 programming system takes advantage of the many man-years of program development and field testing by PDP-5 users.

Although in many cases PDP-8 programs originated as PDP-5 programs, all utility and functional program documentation is issued in a new, recursive format introduced with the PDP-8.

Programs written by users of either the PDP-5 or the PDP-8 and submitted to the users' library (DECUS - Digital Equipment Corporation Users' Society) are immediately available to PDP-8 users.

Consequently, users of either computer can take immediate advantage of the continuing program developments for the other.

CONTENTS

<u>Chapter</u>		<u>Page</u>
1	INTRODUCTION	1-1
2	DECTAPE SUBROUTINES	2-1
	Description	2-1
	Necessary Conventions	2-2
	Format	2-4
	Search Subroutine	2-4
	Read and Write Subroutines	2-5
	Errors	2-6
	Brief Description of DECTape Instructions	2-6
3	PDP-8 LIBRARY SYSTEM	3-1
	Description	3-1
	Definitions	3-1
	Named File	3-1
	Library System (or Skeleton System)	3-1
	Library	3-1
	System Exit	3-1
	Directory	3-1
	System Loader	3-1
	Usage	3-2
	Index	3-2
	Escape	3-2
	Update	3-3
	Delete	3-4
	Getsys	3-4
	Directory Size	3-5
	Error Halts	3-5
	Storage Map	3-6
<u>Appendix</u>		
1	DECTAPE TRANSPORT AND CONTROL	A1-1

CHAPTER 1

INTRODUCTION

PDP-8 DECTape software provides the programmer with the following three major operational materials:

1. Subroutines which the programmer may easily incorporate into a program for data storage, logging, data acquisition, data buffering (queueing), etc.
2. A library calling system for storing named programs on DECTape and a means of calling them with a minimal size loader.
3. Programs for preformatting tapes controlled by the content of the switch register to write the timing and mark channels, to write block formats, to exercise the tape and check for errors, and to provide ease of maintenance.

Subroutine development has resulted in a series of subroutines which will read or write any number of DECTape blocks, read any number of 129-word blocks as 128 words (or one memory page), or search for any block (used by read and write or to position the tape). These programs are assembled with the user's program and are called by a jump to subroutine instruction. The program interrupt is used to detect the setting of the DECTape (DT) flag thus allowing the main program to proceed while the DECTape operation is being completed. A program flag is set when the operation has been

completed. The program may thus effectively allow concurrent operation of several input-output devices along with the DECTape. These programs occupy two memory pages ($400_8 = 256_{10}$ words).

The Library System was developed with several design criteria in mind. First and perhaps foremost, the system should leave the state of the computer unchanged when it exits. Second, it should be capable of calling programs by name from the keyboard and allow for expansion of the program file stored on the tape. Finally, it should conform to existing system conventions, namely, that all of memory except for the last memory page ($7600_8 - 7777_8$), be available to the programmer. This convention was established so that the Binary Loader (paper tape), and/or future versions of this loader could reside in memory at all times.

The PDP-8 DECTape Library System is loaded by a 17_{10} instruction bootstrap routine that starts at 7600_8 . This loader calls a larger program into the last memory page, whose function is to preserve on the tape, the contents of memory from $6000_8 - 7577_8$, and then load the INDEX program and the directory into those same locations. Since the information in this area of memory has been preserved, it can be restored when operations have been completed. The minimum system tape contains the following programs:

INDEX	Typing this causes the names of all programs currently on file to be typed out.
UPDATE	Allows the user to add a new program to the files. UPDATE queries the operator about the program's name, its starting address, and its location in core memory.
GETSYS	Generates a skeleton Library Tape on a specified DECtape unit.
DELETE	Causes a named file to be deleted from the tape.

Starting with the skeleton library tape, the user can build up a complete file of his active programs and continuously update it. One of the uses of the library tape may be illustrated as follows:

A program is written in PDP-8 FORTRAN that is to be used repeatedly. The programmer may call the FORTRAN compiler from the library tape and with it compile the program, obtaining the object program. The FORTRAN Oper-

ating System may then be called from the library tape and used to load the object program. At this time the library program UPDATE is called, the operator defines a new program file (consisting of the FORTRAN Operating System and the object program), and adds it to the library tape. As a result, the entire operating program and the object program are now available on the DECtape library tape.

The last group of programs, a collection of short routines controlled by the contents of the switch register, is called DECTOG. It provides for the recording of timing and mark channels and permits block formats to be recorded for any block length. Patterns may be written in these blocks and then read and checked. Specified areas of tape may be "rocked" for specified periods of time. A given reel of tape may thus be thoroughly checked before it is used for data storage. These programs may also be used for maintenance and checkout purposes.

CHAPTER 2

DECTAPE SUBROUTINES

DESCRIPTION

DECTape subroutines allow the programmer to read, write, or search DECTape using prewritten and tested subroutines. The routines include three basic subroutines:

1. A search subroutine which finds a requested block on the tape and can exit with the tape moving in either direction or stopped. This subroutine can be used directly by the programmer or automatically by the read and write subroutines.
2. A read forward subroutine which searches for the block requested and reads data into core memory beginning at an indicated core location, from that block and from subsequent blocks.
3. A write forward subroutine which searches for the block requested and writes onto that block and subsequent blocks the contents of core memory beginning at an indicated core location.

The routines included will not read or write in the reverse direction. Instructions can be executed in the main program simultaneously with search, read, and write functions. The accumulator and link bit is preserved by the programmer and not by the DECTape subroutines.

The data break facility in DECTape data transfers allows information to be transferred in segments equal in size to the block capacity of the tape currently being used. This capacity is usually 129 (201 octal) 12-bit words per block (i.e., as in certified tapes), but can be of any length the user cares to create as long as the number of 12-bit words per block is a multiple of 3.

A mark track error detection circuit requires that an identifying mark be read every six lines on the tape. This 6-line mark track section requires that data be recorded in 12-line segments (12 being the lowest common multiple of 6-line marks and 4-line data words) which correspond to 3 12-bit words. This "multiple of 3" restriction results in an inability to construct a block whose capacity is the same as a core memory page. The closest approach is the creation of a block 129 words in length, exceeding the size of a core page by one word. The read and write subroutines included in this program have been designed to work with pages 129 words in length as if they were only 128 words in length, thus allowing the programmer to deal with tape blocks which are effectively equal in size to a core page. There are, therefore, two entries to the read subroutine and two entries to the write subroutine. One entry deals with tape blocks regardless of their size and the other entry deals with 129-word blocks as if they were 128-word blocks (see the individual subroutines for details of entry).

The subroutines are supplied as symbolic tapes occupying 400 octal words of core storage (2 successive pages). The tapes contain no specific origin and are not terminated by a dollar sign. The user positions the subroutine tape within his program and supplied origin information as he desires.

NECESSARY CONVENTIONS

The DECTape subroutines use the program interrupt facility as well as the data break facility and have a user-created interrupt service routine and dismiss routine.

The interrupt service routine is the section of a program to which control is transferred at the moment an interrupt occurs. It is responsible for preserving the states of the AC and link, determining the cause of the interrupt, and transferring control to a subprogram which will process the interrupting condition.

The dismiss routine is responsible for restoration of the states of the AC and link, reactivation of the program interrupt system, and transfer of control back to the interrupted program.

Due to the critical timing of certain DECTape functions, it is necessary that the user construct the interrupt service routine in such a way that a subroutine handling a low priority interrupt (user assigned, non-DECTape) may itself be interrupted.

The high-priority interrupt conditions which the service routine should check are the DECTape

error flag and DT flag, respectively. The error flag is checked by the MMSC instruction. If this IOT instruction skips because of the error flag being set, a jump indirect to a register containing the address MERR should be executed. The DT flag is checked by the MMSF instruction. If this IOT instruction skips because of the DT flag being set, the instruction JMP I MCOM must be executed. A register must be created (on the current page) and tagged with MCOM.

Access by the DECTape subroutines to the dismiss routine is accomplished by the execution of DISM. The user must define DISM as an effective jump to the dismiss routine. If the dismiss routine is on page 0, this parameter assignment takes the form DISM = JMP Z ?, where ? is the address of the dismiss routine. If the dismiss routine is not on page 0, the parameter assignment takes the form DISM = JMP I Z R where R is the address of a page 0 address containing the address of the dismiss routine.

The mnemonic AUTO must be defined in the user's program as a page 0 auto-index register (i.e., AUTO = 10).

The programming example below illustrates the DECTape subroutine programming conventions in the construction of the user's interrupt routines. The tags INTR, INT, SCAT, SRET, SAC, LINK, SAU, LK, READ, and ERCM in this example were chosen at random. The use of register 5 as an intercom to the dismiss routine is optional; any page 0 nonauto-index register will do. If the interrupt service routine and dismiss routine

are located on page 0, the two references to them need not be indirect (i.e., DISM = JMP Z SCAT). This example assumes that the user

requires communication with the ASR 33 Keyboard via the program interrupt.

```

*2          JMP I . + 1
            INTR

*5          SCAT
DISM = JMP I Z 5
AUTO = 10

*1000       /INTERRUPT SERVICE ROUTINE
INTR,       KSF
            JMP INT
            DCA SAV          /SAVE THE AC
            RAL
            DCA LK          /SAVE THE LINK
            KCC             /CLEAR KEYBOARD FLAG
            TAD Z 1
            ION             /RESTART INTERRUPT SYSTEM
            DCA SRET        /SAVE RETURN
            JMP READ        /SERVICE KEYBOARD

INT,        DCA SAC          /SAVE AC
            RAL
            DCA LINK        /SAVE THE LINK
            MMSC            /SKIP ON DECTAPE ERROR FLAG
            SKP
            JMP I ERCM       /GO TO MERR IN DECTAPE SUBROUTINES
            MMSF            /SKIP ON DECTAPE DT FLAG
            SKP
            JMP I MCOM       /GO TO DECTAPE SUBROUTINES

SCAT,       CLA CLL          /INTERRUPT DISMISS ROUTINE
            TAD LINK
            RAR             /RESTORE LINK
            TAD SAC         /RESTORE AC
            ION             /RESTART INTERRUPT SYSTEM
            JMP I Z 1        /RETURN TO INTERRUPTED PROGRAM

SAC,        0
LINK,       0
SRET,       0
MCOM,       0              /MODIFIED BY DECTAPE SUBROUTINES
ERCM,       MERR          /ERROR ROUTINE ADDRESS IN DECTAPE SUBROUTINES

SAV,        0
LK,         0

```

READ,	KRB	/KEYBOARD INTERRUPT HANDLER

	CLA CLL	
	TAD LK	
	RAR	/RESTORE LINK
	TAD SAV	/RESTORE AC
	JMP I SRET	/RETURN TO INTERRUPTED PROGRAM

FORMAT

Search Subroutine

The search subroutine is used for positioning of block automatically. The user may independently access the search subroutine for his own purposes. The reading and writing subroutines will use the search subroutine to find the correct purposes.

Instructions

Comments

TAD B	Where B equals the number of the block being sought.
JMS I X	Where X is the address of a register containing LVSF, LVSR, or LVST. If LVSF is used, the tape will be moving in search forward mode when searching is completed (the instructions CLA and MMLM will stop the tape). If LVSR is used, the tape will be moving in search reverse mode when the search is completed. If LVST is used, the tape will be stopped when searching is completed.
E	Where E equals the address of the return location should any error occur. The interrupt will be off and the tape will be stopped. The AC will contain a code indicating the type of error.
C	Where C indicates the address of the location to return to after searching has been completed. If the main program is to be continued during searching, this should be coded as a DISM instruction. The interrupt will be off when control is transferred to this address.
ZZ00	Where ZZ indicates the number of the unit to be selected (1 - 10 ₈).
RETURN	The program will return to this line when searching starts. This allows multiprocessing. During additional program breaks, the search routine will return via the DISM instruction. The programmer can determine when searching has been completed by using any of the following sequences since DONE is reset to 0 after each block mark is read and to -1 (7777) when the correct block has been found.

1. ISZ I X	/C(X) = DONE
JMP ?	/NOT DONE
2. TAD I X	/C(X) = DONE
SZA	
JMP ?	/DONE
3. TAD I X	/C(X) = DONE
SNA	/OR SMA
JMP ?	/NOT DONE

(If no multiprocessing is necessary this line should be coded as a JMP to itself.)

Read and Write Subroutines

<u>Instructions</u>	<u>Comments</u>
JMS I ?	Where C (?) equals MRDS (generalized read), R128 (specialized read), MWRS (generalized write), or W128 (specialized write). MRDS and MWRS are subroutines which will read from or write onto blocks without respect to their size. R128 and W128 are subroutines which deal with 129-word blocks as if they were 128-word blocks.
S	Where S is the address of the first core location to be read into or written from.
E	Where E equals the address of the location to be returned to in the event of an error. The tape will be stopped and the interrupt will be turned off.
ZZ00	Where ZZ equals the number of the unit to be selected.
N	Where N equals the number of consecutive blocks to be read from or written into.
B	Where B equals the number of the initial block to be written into or read from.
RETURN	The program will return to this line when searching starts to allow multiprocessing. During additional program breaks and when reading or writing has been completed, the return will be via DISM instruction, thus this line should never be a jump to itself. The interrupt will be on after completion. Multiprogramming takes place during the read and write functions as well as during the search functions.

ERRORS

Should any errors occur after the subroutines are entered, a return will be made to the main program at the location indicated by the error return, with the following numbers in the accumulator.

<u>Contents of Accumulator</u>	<u>Meaning</u>
0001	Illegal format. Requested an illegal block.
0002	The block requested cannot be found.*
0003	The DECTape error flag was raised during searching for other than an end-of-tape condition.*
0004	The DECTape error flag was raised during reading.**
0005	The DECTape error flag was raised during writing.**

BRIEF DESCRIPTION OF DECTAPE INSTRUCTIONS

The following description is designed as an aid to understanding the DECTape routines only and should not be taken as a detailed description of the DECTape system.

<u>Mnemonic Instructions</u>	<u>Code</u>	<u>Comments</u>
MMLS	6751	DECTape select. Clears the DECTape flag. Selects the unit designated by a 2-5 of the AC and institutes a 70-msec delay. The DECTape flag is set at the end of this delay.
MMLM	6752	DECTape load motion. Clears the DECTape flag. Loads the motion register from AC bits 7 and 8 (bit 7 a 1 equals GO, bit 7 a 0 equals STOP, bit 8 a 1 equals REVERSE, bit 8 a 0 equals FORWARD). Institutes a 70-msec delay and sets the DECTape at the end of the delay.
MMLF	6754	Loads the function register from the contents of AC bits 9-11 and then clears the AC. Institutes no delay and undertakes no action in regard to the DECTape flag. The octal number in bits 9 through 11 is coded as follows:

*The number of the block mark being searched for can be found at BLKN. The block mark just read can be found at TEMN.

**The number of the block being read when the error occurred can be found at TEMN.

0 = Move
 1 = Search
 2 = Read data
 3 = Read all bits
 4 = Write data
 5 = Write all bits
 6 = Write mark and timing

MMMF	6756	DECTape load function and motion. A microprogrammed instruction encompassing MMLM and MMLF. Sets function and motion registers from AC bits 7-11, i.e., 22 = Read data forward 32 = Read data reverse 24 = Write data forward 34 = Write data reverse 21 = Search forward 31 = Search reverse etc.
MMMM	6757	DECTape load select, motion, and function. A microprogram of instructions MMLF, MMLM, and MMLS.
MMCC	6762	DECTape clear memory address counter. Clears the memory address counter.
MMLC	6764	DECTape load memory address counter. Contents of the accumulator are transferred into the memory address counter and then the AC is cleared.
MMML	6766	Clears and loads the memory address counter. A microprogram of MMCC and MMLC.
MMSF	6761	Skip on DECTape flag. If the DECTape flag has been set to a 1, the contents of the PC are incremented by one and the next instruction is skipped. Does not clear flag. If flag is set: <u>In search mode:</u> A block mark number has just been read and transferred to core address contained in MAC. Contents of MAC remain unchanged. <u>In read data mode:</u> A block of data has just been transferred to core memory via data break. The control is still in read data mode and the tape is moving. MAC contains the address of next sequential core register following the last one read into. The next block on tape will be read into core beginning at current address in MAC if no action is taken.

In write data mode: A block of data has just been written on tape from core memory via data break. The control is still in write data mode and the tape is moving. MAC contains the address of the next sequential core address following the last one written from. The next block on tape will be written onto from core beginning at current address in MAC if no action is taken.

MMSC

6771

Skip on error flag. If the error flag has been set to a 1, the contents of the PC are incremented and the next instruction is skipped.

Encountering an end zone will set the error flag. In this case, expect to see an MT flag set 70-msec after the end zone is encountered. This is a signal to institute turn-around action (if desired) and be assured of opening the "eyes" of the control in time to read the block mark number of the last block encountered prior to the end zone (useful procedure when "bouncing" off the end zone to read or write the end block).

MMCF

6772

Clears the DECTape DT and error flags.

MMRS

6774

The condition of the status levels is transferred into bits 0 through 7 of the accumulator. The accumulator bit assignments are:

- 0 = DT flag
- 1 = Error flag
- 2 = END (selected tape at end point)
- 3 = Timing error
- 4 = Reverse tape direction
- 5 = GO
- 6 = Parity or mark track error
- 7 = Select error

```

/PDP-8 DECTAPE SUBROUTINES - SEARCH
/"DISM" MUST BE DEFINED AS AN EFFECTIVE JUMP TO THE DISMIS ROUTINE
/ VIA PAGE ZERO
/"AUTO" MUST BE DEFINED AS SOME AUTO INDEX REGISTER
/"MCOM" MUST BE THE ADDRESS TAG OF THE REGISTER INDIRECTLY REFERENCED
/ BY A JMP INSTR. WHEN THE INTERRUPT ROUTINE FINDS THE DT FLAG SET
/ WHEN THE INTERRUPT ROUTINE FINDS THE ERROR FLAG SET, AN
/ EFFECTIVE JUMP TO "MERR" MUST BE EXECUTED
/EXAMPLE:
/      MMSC
/      SKP
/      JMP I ERCO      /RESPONSE TO DECTAPE ERROR FLAG
/      MMSF
/      SKP
/      JMP I MCOM      /RESPONSE TO DECTAPE MT FLAG
/      .....
/      .....
/      .....
/MCOM, 0
/ERCO,  MERR
/SCAT,  TAD
/      RAR      /RESTORE LINK
/      TAD      /RESTORE C(AC)
/      ION
/      JMP I Z 1  /RETURN TO MAIN PROG
/DISM=JMP I Z .
/      SCAT
/AUTO=10
/SEARCH ROUTINE FORMAT:
/      TAD B      /WHERE B=BLOCK NUMBER
/      JMS I X      /WHERE X IS A REGISTER CONTAINING LVSF,
/                  /LVSF OR LVST
/      E          /WHERE E=ERROR RETURN ADDRESS
/      C          /WHERE C=COMPLETION RETURN ADDRESS
/      ZZ00       /WHERE ZZ=UNIT NUMBER (OCTAL)
/      MULTIPROGRAM RETURN
/SEARCH AND LEAVE TAPE MOVING IN SEARCH REVERSE MODE
LVSR,  0
      DCA BLKN
      TAD LVSR
      DCA LVSF
      JMP LVSF+3
/SEARCH AND LEAVE TAPE STOPPED IN SEARCH FORWARD MODE
LVST,  0
      DCA BLKN
      TAD LVST
      DCA LVSF
      CMA
      DCA LVST
      JMP LVSF+2

```

```

/SEARCH AND LEAVE TAPE MOVING IN SEARCH FORWARD MODE
LVSF, 0
      DCA BLKN
      IAC
      DCA BACO
      CMA
      TAD BACO
      DCA FOCO
      IOF
      PCF
      TCF
      RRB
      DCF
      PLCF
      KCC
      MMCF
      TAD CN9      /DATA
      DCA I MIN3  /MCOM
      TAD CN1      /TEMN
      MMML
      DCA DONE
      TAD CN2      /31
      DCA DIRC
      TAD BLKN
      SNA
      JMP FERR+1 /FORMAT ERROR
      CIA
      DCA BLKN
      TAD CN3      /7770
      DCA DRC      /CHANGE OF DIRECTION COUNTER
      TAD I LVSF  /PICK UP ERROR RETURN ADDRESS
      DCA LVSR
      ISZ LVSF
      TAD I LVSF  /PICK UP COMPLETION RETURN ADDRESS
      DCA BACK
      ISZ LVSF
      TAD I LVSF  /PICK UP UNIT NUMBER
      DCA UNIT
      ISZ LVSF
      TAD LVSF    /PICK UP MULTI PROGRAM RETURN ADDRESS
      DCA Z 1     /SET UP FOR INITIAL DISMIS

```

```

TURN,    ISZ DRC
          SKP
          JMP FERR /BLOCK CANNOT BE FOUND
          TAD CN6 /-21
          TAD DIRC
          SNA CLA
          JMP MREV
          TAD CN7 /21
          DCA DIRC
          TAD SPAC /SPA CLA
          DCA SCH2
          TAD BLKN
          TAD FOCO
          DCA DISB
          TAD UNIT
          TAD DIRC
          MMMM
          JMS I MINO /WAIT
          DISM
DATA,    MCMF
          TAD TEMN
          TAD DISB
          SZA /BLOCK FOUND
          JMP SCH2
          TAD TEMN
          TAD BLKN
          SZA CLA /OBJECT BLOCK FOUND
          JMP TURN
          CMA
          DCA DONE
          ISZ LVST
          JMP I BACK /LEAVE TAPE MOVING
          MMLM /STOP TAPE
          JMS I MINO /WAIT
          JMP I BACK
SCH2,    SPA CLA /OR SMA CLA (WHEN IN REVERSE DIRECTION)
          DISM
          JMP TURN /REVERSE THE TAPE DIRECTION
MREV,    TAD CN2 /31
          DCA DIRC
          TAD DIRC
          MMMF
          JMS I MINO /WAIT
          TAD SMAC /SMA CLA
          DCA SCH2
          TAD BLKN
          TAD BACO
          DCA DISB
          DISM
ERGO,    JMP I LVSR /GO TO USER'S ERROR ROUTINE
/FORMAT AND NOT FOUND ERROR
FERR,    IAC
          IAC
          JMP I MIN2 /OOPS

```

/ CONSTANTS

CN1, TEMN
CN2, 31
CN3, 7770
CN6, 7757 /-21
CN7, 21

CN9, DATA

SMAC, SMA CLA

SPAC, SPA CLA

/ INTERCOMMUNICATIONS REGISTERS

MIN0, WAIT

MIN2, OOPS

MIN3, MCOM

/ VARIABLES

BLKN, 0

BACO, 0

FOCO, 0

DIRC, 0

DRC, 0

BACK, 0

UNIT, 0

DISB, 0

DONE, 0

TEMN, 0

*L VSR+200

/PDP-8 DECTAPE SUBROUTINES - READ AND WRITE

/FORMAT:

```

/      JMS I Y      /WHERE Y=REGISTER CONTAINING MRDS, R128, MWRS
/      S            /WHERE S=STARTING CORE ADDRESS
/      E            /WHERE E=ERROR RETURN ADDRESS
/      ZZ00         /WHERE ZZ=UNIT NUMBER (OCTAL)
/      N            /WHERE N=NUMBER OF CONSECUTIVE BLOCKS
/      B            /WHERE B=INITIAL BLOCK NUMBER
/      MULTI-PROGRAM RETURN

```

/ERROR HANDLING ROUTINE

```

MERR,  MMRS
      AND CN13      /1000
      SZA CLA
      JMP MENZ
      TAD I MIN5 /MCOM
      CIA
      TAD CN20      /DROR
      SNA CLA
      TAD ER CN
      TAD CN22      /3
OOPS,  DCA HOOP
      MMRS
      DCA MRSA      /SAVE STATUS
      TAD HOOP
      MMLM          /STOP TAPE
      JMS WAIT
      JMP I MIN7 /ERGO
MENZ,  JMS WAIT
      JMP I MIN6 /TURN

```

/WRITE SUBROUTINE (TO WRITE ON 129 WORD BLOCKS)

```

W128,  0
      CLA
      TAD W128
      DCA MRDS
      TAD CN23      /MRD2
      JMP MWRS+5

```

/GENERAL WRITE SUBROUTINE

```

MWRS,  0
      CLA
      TAD MWRS
      DCA MRDS
      TAD CN18      /PASS
      DCA CHK1+2
      TAD CN10      /TAD PASS+3 (6754)
      DCA PASS+2
      JMP MRD1

```

/READ SUBROUTINE (TO READ 128 WORDS FROM A 129 WORD BLOCK)

```

R 128, 0
      CLA
      TAD R128
      DCA MRDS
      TAD CN23      /MRD2
      JMP MRDS+3

```

/READ SUBROUTINE (TO READ ANY LENGTH BLOCK

```

MRDS,      0
           CLA
           TAD CN18      /PASS
           DCA CHK1+2
           TAD CN12      /TAD DROR+3 (6772)
           DCA PASS+2
           SKP
MRD1,      IAC
           IAC
           DCA ERCN
           TAD I MRDS
           DCA HERE      /SAVE STARTING CORE ADDRESS
           TAD MRDS
           JMP CHK        /GO FIND BLOCK
MRD2,      TAD HERE
           TAD CN15      /200
           DCA R128
           TAD I R128    /SAVE 129TH CORE WORD
           DCA MWRS
PASS,      TAD HERE
           MMLL
           TAD           /DROR+3 (2, READ) OR PASS+3 (4, WRITE)
           MMLF
           TAD CN20      /DROR
           DCA I MIN5    /MCOM
           DCA I MIN4    /DONE
           DISM
DROR,      MMSC          /RETURNS HERE AFTER READING OR WRITING DONE
           SKP CLA
           JMP MERR
           MMCF
           TAD CHK1+2
           CIA
           TAD CN23      /MRD2
           SZ A CLA
           JMP EXNT-2
           TAD MWRS
           DCA I R128    /RESTORE 129TH CORE WORD
           TAD R128
           MMLL
           TAD R128
           TAD CN15      /200
           DCA R128
           TAD I R128
           DCA MWRS
           ISZ HOOP
           DISM
EXNT,      MMLM
           JMS WAIT
           CMA
           JMP DROR-2

```

```

/Routine TO EXTRACT ARGUMENTS AND INSTITUTE BLOCK SEARCH
CHK,   DCA Z AUTO /SET UP AUTO INDEX REG
        TAD I Z AUTO           /PICK UP ERROR ROUTINE ADDRESS
        DCA CHK1+1
        TAD I Z AUTO           /PICK UP UNIT SELECTION
        DCA CHK1+3
        TAD I Z AUTO           /PICK UP NUMBER OF BLOCKS
        CIA
        DCA HOOP
        TAD I Z AUTO           /PICK UP INITIAL BLOCK NUMBER
CHK1,   JMS I MIN8 /LVSF
        0
        0
        0
        JMP I Z AUTO           /MULTI-PROGRAM WITH MAIN PROGRAM
/ ROUTINE TO CLEAR MT FLAG AFTER LOAD COMMANDS
WAIT,   0
        MMSF
        JMP .-1
        MMCF
        JMP I WAIT

/VARIABLES
ERCN,   0
HERE,   0
HOOP,   0
MRSA,   0
/CONSTANTS
CN10,   TAD PASS+3
CN12,   TAD DROR+3
CN13,   1000
CN15,   200
CN18,   PASS
CN20,   DROR
CN22,   3
CN23,   MRD2
/INTERCOMMUNICATION REGISTERS
MIN4,   DONE
MIN5,   MCOM
MIN6,   TURN
MIN7,   ERGO
MIN8,   LVSF

*MERR+200

```

CHAPTER 3

PDP-8 LIBRARY SYSTEM

DESCRIPTION

The Library System was developed with several design criteria in mind. First and perhaps foremost, the system should leave the state of the computer unchanged when it exits. Second, it should be capable of calling programs by name from the keyboard and allowing for expansion of the program file stored on the tape. Finally, it should conform to existing system conventions; namely, that all of memory except for the last memory page ($7600_8 - 7777_8$) be available to the programmer. This convention was established so that the Binary Loader (paper tape) and/or future versions of this loader could reside in memory at all times.

With these ideas in mind, the PDP-8 DECtape Library System was developed. It is loaded by a 17_{10} instruction bootstrap routine that starts at 7600_8 . This loader calls a larger program into the last memory page whose function is to preserve on tape the contents of memory from $6000_8 - 7577_8$, and then load the INDEX program and the directory into those same locations. Since the information in this area of memory has been preserved, it can be restored when operations have been completed.

DEFINITIONS

Named File

This is a binary entity defined by the user. It consists of a name, a starting address, and one or more memory pages which need not be contiguous.

Library System (or Skeleton System)

A collection of five programs stored on DECtape. These programs provide for the loading of named files into core memory, for the definition of new named files, for the deletion of named files, and for the creation of a new Library System. The Library System is loaded by the System Loader.

Library

The sum of user-defined named files and the Library System.

System Exit

When the Library System has loaded a named file into core memory, it restores all of memory that it occupied and it restores the System Loader and the Binary Loader to their normal locations. This is called System Exit.

Directory

This is part of the Library System. It contains the names of files on the library tape and all information that is required by the Library System to load a named file, to delete a named file, or to add new named files to the library.

System Loader

This is a short "bootstrap" program that loads the Library System. The Library System may then load a named file or perform some other operation. The System Loader is initially loaded with the LOAD ADDRESS key, the DEPOSIT key, and the switch register. It is automatically restored upon system exit. The System Loader is as follows:

*7600

7600/	1223	TAD REVC
7601/	6757	MMLM MMLF MMLS
7602/	4216	JMS WAIT
7603/	4216	JMS WAIT
7604/	1224	TAD FRCN
7605/	6756	MMLM MMLF
7606/	4216	JMS WAIT
7607/	1225	TAD ADRN
7610/	6766	MMML
7611/	4216	JMS WAIT

/THE SYSTEM GENERATES
/CODING HERE ON RETURN

		WAIT, 0
7617/	6761	MMSF
7620/	5217	JMP .-1
7621/	6772	MMCF
7622/	5616	JMP I WAIT
7623/	1030	REVC, 1030
7624/	0022	FRCN, 0022
7625/	7600	ADRN, 7600

USAGE

1. The library tape is placed on Unit 8 with the WRITE switch UP.
2. The System Loader (assumed to be in core memory) is started at address 7600₈.
3. The Library System automatically loads itself into core memory. It is now waiting for the operator to type a name (up to six characters) on the ASR 33 Teletype. This name must be terminated by a carriage return. The Library System searches the directory for this name. If it is found, the file of that name is loaded into core memory, the system exits, and control transfers to the starting address of the file. If the name typed by the operator is not found or if the operator has not correctly terminated

the name, a question mark will be typed and the line is ignored.

The Library System tape always contains five permanent names which comprise the actual system. These names are INDEX, ESCAPE, UPDATE, DELETE, and GETSYS. Their functions are as follows:

Index

INDEX causes the names of all files on the library tape to be typed out on the console teleprinter.

Escape

ESCAPE causes the Library System to exit: that is, it restores all of core memory that it has used and restores the System Loader and the BIN Loader to their normal locations. The computer then halts at location 7700₈, the HLT in the BIN Loader.

Update

UPDATE is the name of a permanent system program. It allows the user to add files to the library tape. When it is called, it types questions for the operator to answer. It assumes that the file to be added to the library tape was in core memory before the system was loaded. The PDP-8 Library System stores files in blocks of 0200_8 12-bit words, i.e., one memory page. UPDATE requires information concerning the locations of these blocks in memory, the starting address of the file, and the name to be given to the file. It will type:

Name of Program:

The operator types a name that contains, at most, six characters terminated by a carriage return. A rubout causes anything typed to be ignored, and UPDATE retypes the line. All characters are legal except @, ↑, tab, form feed, and line feed. When this line has been answered by the operator, UPDATE will type:

SA (Octal) :

The operator types an octal address delimited by a carriage return. This is the location to which control transfers when this file is loaded. If the file being updated does not have a proper starting address (for example, the floating-point package), the operator may specify that the starting address is 7600_8 , the starting address of the System Loader, or he may specify that the starting address is 7700_8 , the HLT in the BIN Loader. A rubout typed here will cause the UP-

DATE program to ignore anything previously typed on the line and UPDATE will retype the question.

Page Locations

The Library System loads files by memory pages and UPDATE is now requesting this information. Information is provided to UPDATE in either of two forms: `<xxxx>` which means the single page on which the octal address `xxxx` falls or `<xxxx,xxxx>` which means the page on which the first address through (inclusively) the page on which the second address falls. UPDATE accepts information of this type until a semicolon (;) is received. Spaces, tabs, carriage returns, and line feeds are ignored outside of location elements. For example, if a program had the following limits, 1-2354, 4600-7577, 2400-2577, UPDATE would be told: `<0, 2200> <2400> <4600, 7577>`.

NOTE: The numbers must be in sequence and any numbers lying within the same page are considered equivalent. It could have been presented as: `<177, 2377> <2577> <4777, 7400>`; etc.

If the operator has made a typing error, UPDATE will retype the entire question. A rubout causes the entire line to be ignored.

When the new file has been added to the library, control transfers to the file-loading program with the directory in memory.

If UPDATE finds that a file has already been defined with this name, it will type an error mes-

sage and return to the file-loading program from which UPDATE may be recalled.

Delete

DELETE is the name of a permanent system program. It removes any user-defined file from the library tape. It will not delete a system program. When DELETE is called, it types: NAME OF FILE TO BE DELETED: The operator types a name (up to six characters) terminated by a carriage return. A rubout causes anything typed to be ignored and the entire line is re-typed. When the name of the file to be deleted has been entered, DELETE tries to find it in the directory. If it is not found or if it is a system program, an error message is typed. In either case, control returns to the file-loading program with the directory in memory. If the name is found, DELETE will go through a series of operations to remove it from the directory and the library tape. Thus, all space is recovered. When the deletion has been completed, control transfers to the file-loading program with the directory in memory.

Getsys

GETSYS is the last of the permanent system programs. It creates, on a specified tape unit, a new Library System tape consisting of the loaders and the system programs. When GETSYS is called, it will type: SKELETON TAPE WILL BE CREATED ON UNIT#.

It now expects a single digit from 1 to 7 terminated by a carriage return. On that DECtape

unit, it expects the WRITE switch to be in the UP position, and that the tape has been prerecorded to the following specifications: 201_8 12-bit words per block 2700_8 blocks including block numbers 0000. (This is done with DEC-TOG.) When the new Library System has been created, control transfers to the file-loading program with the directory in memory.

A sample of system usage is as follows:

1. Load FORTRAN Compiler with Binary Loader.
2. Set $C(SR)=7600_8$.
(System Loader in core, Library System on unit 8, WRITE switch UP)
3. Press LOAD ADDRESS.
4. Press START.
5. (Everything underlined is typed by the Library System.)

INDEX) USER TYPES THIS, SYSTEM
WILL TYPE:

ESCAPE

UPDATE

DELETE

GETSYS

ODTLO THIS IS ODT 1200-1760

EDIT33 TAPE EDITOR

PALII PAL ASSEMBLER

UPDATE) USER TYPES THIS

PROGRAM NAME : FRTRAN)

SA (OCTAL) : 200_8)

PAGE LOCATIONS: <0, 7400>

```

INDEX )
ESCAPE
UPDATE
DELETE
GETSYS
ODTLO
EDIT33
PALII
FRTRAN
DELETE )
NAME OF FILE TO BE DELETED: ODL
NAME OF FILE TO BE DELETED: ODTLO )
INDEX )
ESCAPE
UPDATE
DELETE
GETSYS
EDIT33
PALII
FRTRAN
ESCAPE )

```

RUBOUT

Computer halts at location 7747₈.

DIRECTORY SIZE

The directory contains 348₁₀ usable locations, with each entry requiring a minimum of seven

locations. Thus, there is space for almost 50₁₀ names in the directory. UPDATE will determine whether or not the directory is full and if so, it will type a message. There are 2640₈ usable blocks of DECtape, which is more than adequate considering the limitations on the directory size. The amount of time required to load a file from tape into core memory is a function of the file location of the tape. If the file is near the beginning, loading time will be about 8 seconds. UPDATE time requires 30 to 45 seconds. DELETE time varies too much to make an estimate possible; it may take as long as several minutes. GETSYS requires approximately 30 seconds.

ERROR HALTS

The system has one DECtape error halt at location 7670₈. The accumulator will contain the contents of the status register (see PDP-8 Users' Handbook, page 90). No recovery is possible at this point. Any attempt to restart may result in the destruction of the library tape.

STORAGE MAP

The Library System is organized as follows:

<u>Block No.</u>	<u>Function</u>	
0	Bootstrap #2	7600-7777
1	Save contents of	6000-6177
2	Save contents of	6200-6377
3	Save contents of	6400-6577
4	Save contents of	6600-6677
5	Save contents of	7000-7177
6	Save contents of	7200-7377
7	Save contents of	7400-7577
10	Return—System Loader and Bin Loader	
11	Directory	6200-6377
12	Directory	6400-6577
13	Directory	6600-6777
14	File—Loading Program	7000-7177
15	File—Loading Program	7200-7377
16	File—Loading Program	7400-7577
17	UPDATE	6000-6177
20	UPDATE	6200-6377
21	UPDATE	6400-6577
22	UPDATE	6600-6777
23	UPDATE	7000-7177
24	UPDATE	7200-7377
25	UPDATE	7400-7577
26	DELETE	6600-6777
27	DELETE	7000-7177
30	DELETE	7200-7377
31	DELETE	7400-7577
32	GETSYS	6000-6177
33	GETSYS	6200-6377
34	GETSYS	6400-6577
35	GETSYS	6600-6777
36	GETSYS	7000-7177
37	GETSYS	7200-7377
40	GETSYS	7400-7577

APPENDIX 1

DECTAPE TRANSPORT AND CONTROL

SPECIAL FEATURES

DECTape represents a significant new concept in magnetic tape systems. Its simplicity of mechanical design, coupled with an ingenious tape format and unusual method of recording, bring to industry a fast, convenient, reliable, low cost in-out data storage facility and updating device. Special features include:

Fixed-Position Addressing - Permits selective updating of tape information as in magnetic disk or drum storage devices. Units as small as a single computer word may be stored or recorded on tape without disturbing adjacent information. Data blocks are numbered and completely addressable. Inter-record gaps are eliminated thereby increasing tape storage capacity.

Automatic Word Transfers - Use the PDP-8 data break facility to allow concurrent information processing and data acquisition during block transfers.

Simple Transport Mechanism - Reduces maintenance requirements, improves reliability. DECTape's simple drive system requires no capstans, no pressure pad, and no mechanical buffering; therefore tape and head wear is minimal.

Pocket-Size Reels - Handy to carry, easy to load. Each 3-1/2 inch reel holds up to 3 million bits, the equivalent of 4000 feet of paper tape, assuming 6-bit words are used.

Bidirectional Operation - Saves time, provides easy access to stored information; reading, writing, and searching may be conducted in either direction.

Redundant, Phase Recording - Insures transfer reliability, reduces problem of skew in bidirectional operation. Each tape track is redundantly paired with a second, nonadjacent track. Use of phase (rather than amplitude) recording greatly reduces drop-outs due to variations in amplitude.

Prerecorded Timing and Mark Tracks - Simplify programming. Relieve the programmer of the responsibility of furnishing timing or counting instructions and permit block and word addressability.

Pretested Subroutines - Are available for information storage and retrieval, maintenance, and diagnostic tests.

DECTAPE FORMAT

DECTape utilizes a 10-track recording head to read and write five duplexed channels. Three of these track pairs are available for data; the two remaining pairs are used for timing and mark information. Duplication of each track by non-adjacent read/write heads wired in series eliminates most drop-outs due to noise and dust and minimizes the effect of skew (see Figure 1).

Key to the system is DECTape's mark channel which is used to raise flags, create data breaks, detect block mark numbers and block ends, and protect control portions of the tape. In addition, the mark channel provides DECTape with automatic bidirectional compatibility, variable length blocks, and end of tape sensing (Figure 2).

Information is stored on tape in block form (Figure 3). Block length is flexible and determined by information on the mark channel. A complete reel of tape, 849,036 lines, can be divided into any number of blocks up to 4096. Usually, a uniform block length is established over the entire length of a reel of tape by a program which writes mark and timing information at specific locations. However, the ability to write variable-length blocks is useful for certain data formats, for example, where small blocks containing index or tag information need to be alternated with large blocks of data. Each block contains two types of words which are assembled by the Type 552 Control. These are data and control words (Figure 4). Since DECtape has no inter-record gaps, the data portions of adjacent blocks are separated by control words. Control words occupy six lines* and are used to record address and checking information. They provide compatibility between DECtape

written on any of DEC's 12-, 18-, or 36-bit computers. Data words contain stored information and occupy four lines on tape (12 bits). To maintain compatibility with the mark channel format, data words are recorded in 12-line segments (12 being the lowest common multiple of 6-line marks and 4-line data words) which correspond to three 12-bit data words.

Block numbers normally occur in sequence from 1 to N. There is one block numbered 0 and one block N + 1. Programs are entered with a statement of the first block number to be used and the total number of blocks to be read or written. The maximum number of blocks is determined by the following equation in which N_B = decimal number of blocks, and N_W = number of words per block. (N_W must be divisible by 3).

$$N_B = \frac{212112}{N_W + 15} - 2$$

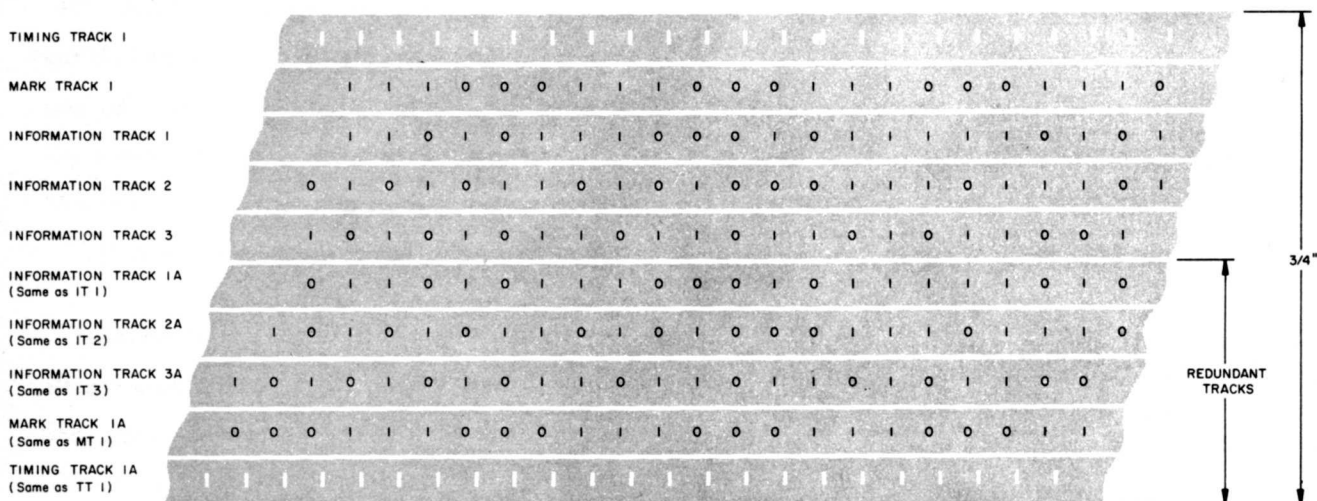


Figure A1-1 Track Allocation Showing Redundantly Paired Tracks

*As used on the PDP-8, only the last four lines of each control word are used.

FUNCTIONAL UNITS

The basic functional elements of the 522 Control and the 555 Transport are shown in the block diagram as they interface with the PDP-8. The main functional elements of the system follow:

Data Buffer (DB)

This 12-bit register serves as a storage buffer for data to be transferred between DECTape and the computer memory buffer register. During a read operation, information sensed from the tape is transferred into the DB from the read/write buffer and is transferred to the computer during a data break cycle. During a write operation, the DB receives information from the computer and transfers it to the read/write buffer for disassembly and recording on tape. In this manner, the DB synchronizes data transfers by allowing transfers between itself and the read/write buffer as a function of the tape timing.

Read/Write Buffer (R/WB)

This 12-bit register is composed of three 4-bit shift registers. During reading, one bit from each information channel is read into a separate segment of the R/WB and shifted right or left as a function of the direction of tape movement. When four tape positions have been read, the content of the R/WB is set into the DB as an assembled 12-bit computer word. During writing, the content of each segment of the R/WB is shifted serially to the write register (one bit from each of the three segments of the R/WB is transferred into the write register at a time to provide the data to be written at one line) for recording on tape.

Write Register

A 3-bit register which is alternately loaded from the R/WB and complemented to write the phase-coded information on tape.

Select Register

This 4-bit register is loaded under program control to specify the tape drive selected for operation from the control unit. A single Type 522 DECTape Control can select the drives of four Type 555 Dual DECTape Transports (eight tape drives).

Motion Register

This 2-bit register contains a go/stop flip-flop and a forward/reverse flip-flop which control the motion of the selected tape drive. The register is set under program control.

Longitudinal Parity Buffer (LPB)

This 6-bit register performs a parity check of the information in the three information channels. The check essentially reads the number of binary zeros in each half of a 12-bit data word and forms a parity bit to be recorded in the checksum control word at the end of the data block. This is effected by setting the information read from two consecutive tape positions into the LPB and then complementing a bit of the LPB if the corresponding bit of the R/WB contains a 0. After reading a block of data, the LPB holds a number which indicates the parity of bits 0 and 6, 1 and 7, etc. A 1 in the

LPB at this time indicates odd parity and a 0 indicates even parity. This information is compared with the data stored in the checksum control word during reading and is used to generate the checksum during writing. If the data read is not equal to the checksum, the parity or mark channel error flip-flop is set to 1.

Memory Address Counter (MAC)

This 12-bit register specifies an address in computer core memory to be used for each word transfer. During program initialization, the starting address of a transfer is set into MAC from the computer accumulator. During the transfer, the address contained in MAC is transferred into the computer memory address register for each data word. The contents of MAC is incremented by one at the conclusion of each word transfer so that transfers occur between successive addresses of computer core memory and tape, regardless of tape direction.

Window (W)

This 9-bit register serves as a control signal generator for the DECTape system. The mark channel data is stored in the W and control signals are generated as a function of the mode of operation in progress and the contents of the W. For example, in the search mode when the W detects a block mark, control signals are generated to raise the DECTape (DT) flag to indicate the presence of a block number in the DB and signals the start of data block to the computer.

Device Selector (DS)

The device selector is a gating circuit which produces the IOT pulses necessary to initiate operation of the DECTape system and strobe information into the computer.

DECTape Flag (DT)

This flip-flop serves as an indicator of DECTape system operation to the computer and is connected to the computer program interrupt facility. The function of the DT flag is determined by the control mode in operation at the time, as follows:

1. In the search mode, the DT flag rises each time a block mark (block number) is read to indicate the beginning of a new block and to allow programmed determination of the block number which just passed the read/write head.
2. In the read data or write modes, the DT flag rises at the end of each block to indicate the end of a data block. Under these conditions, the computer program can sense for this flag to determine when the transfer is complete.
3. In the read all bits or write all bits modes, the DT flag rises to indicate completion of each 12-bit word transfer. Since block marks are not observed in these modes, this flag can be used by the computer program to count the number of words transferred as a means of determining tape location.

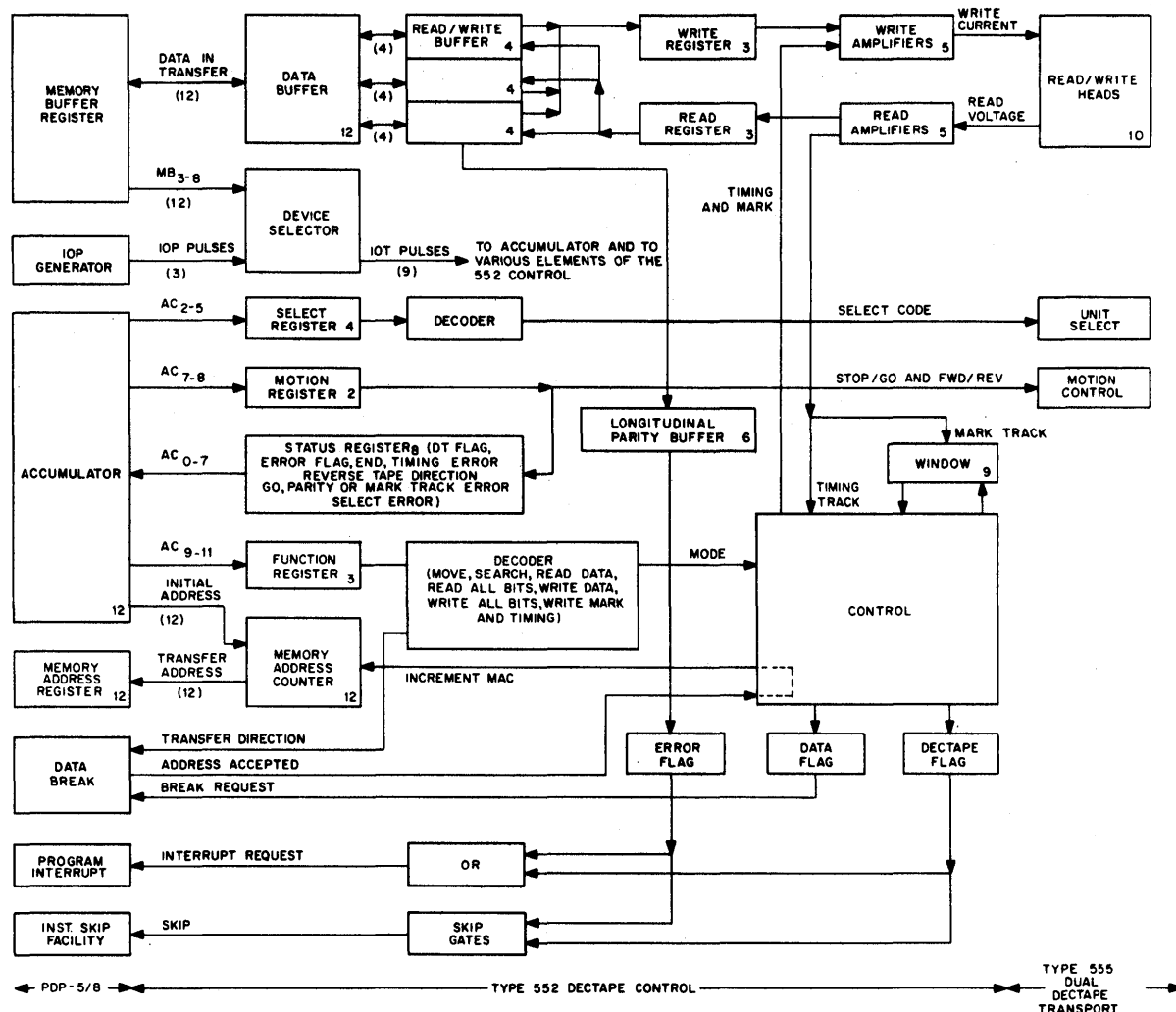


Figure A1-5 DECtape System Block Diagram and PDP-8 Interface

Error Flag

This flag is raised by four error conditions. When the flag rises, it initiates a program interrupt to allow the computer interrupt subroutine to determine the condition of the 552 Control by means of a read status command. The four error conditions indicated are:

legitimate signal used to indicate the end of a normal operation (such as rewind) if it is anticipated by the program. If the transport is not selected when the tape enters the end zone, this signal is not given, tape motion is not stopped automatically, and the tape can run off the end of the reel.

End

The tape of the selected transport is in the end zone and tape motion is stopped automatically. Under these conditions, end is an error if it is not expected by the program in process or is a

Timing Error

The program was not able to keep pace with the tape transfer rate or a new motion or select command was issued before the previous command was completely executed.

Parity or Mark Track Error	Indicates that during the course of the previous block transfer a data parity error was detected or one or more bits have been picked up or dropped out from either the timing track or the mark track.
Select Error	Signifies that a tape transport unit select error has occurred such that more than one transport in the system have been assigned the same select code or that no transport has been assigned the programmed select code.

Therefore, a select error indicates an error by the operator, a timing error is a program error, and a parity or mark track error indicates an equipment malfunction. Under certain conditions, the end may also be an indication of equipment malfunction.

Data Flag

This flag is raised each time the DECTape system is ready to transfer a 12-bit word with the computer. When raised, the flag produces a computer data break.

DECTAPE DUAL TRANSPORT 555

The Type 555 Dual DECTape Transport consists of two logically independent tape drives capable of handling 3.5-inch reels of 0.75-inch magnetic tape. Bits are recorded at a density of 350 ± 55 bits per track inch at a speed of over 80 inches per second on the 260-foot length reel. Each line on the tape is read or written in approxi-

mately $33\frac{1}{3}$ microseconds. Simultaneous writing occurs in the two redundant mark and timing channels.

DECTAPE CONTROL 552

The Type 552 DECTape Control operates up to four Type 555 Dual DECTape Transports (8 drives). Binary information is transferred between the tape and the computer in 12-bit computer words approximately every $133\frac{1}{3}$ microseconds. In writing, the control disassembles 12-bit computer words so that they are written at four successive lines on tape. Transfers between the computer and the control always occur in parallel for a 12-bit word. Data transfers use the data break (high-speed channel) facility of the computer. As the start and end of each block are detected by the mark track detection circuits, the control raises a DECTape (DT) flag which causes a computer program interrupt. The program interrupt is used by the computer program to determine the block number. When it determines that the forthcoming block is the one selected for a data transfer, it selects the read or write control mode. Each time a word is assembled or DECTape is ready to receive a word from the computer, the control raises a data flag. This flag is connected to the computer data break facility to signify a break request. Therefore, when each 12-bit computer word is assembled, the data flag causes a data break and initiates a transfer. By using the mark channel decoding circuits and data break facility in this manner, computation in the main computer program can continue during tape operations.

TABLE A1-1 DECTAPE INSTRUCTION LIST

Mnemonic	Octal	Operation	Mnemonic	Octal	Operation
MMLS	6751	Load unit select register from the contents of AC 2-5 and clear DECTape (DT) flag. DT flag is automatically set approximately 70 msec after this IOT.	MMLC	6764	Load MAC from the contents of AC 0-11 and then clear the AC (1's transfer).
			MMML	6766	Clear and load MAC from the contents of AC 0-11 and then clear the AC.
MMLM	6752	Load motion register from the contents of AC 7-8 and clear DT flag. DT flag is automatically set approximately 70 msec after this IOT.	MMSC	6771	Skip if error flag is a 1.
			MMCF	6772	Clear error flag and DT flag.
			MMRS	6774	Read status bits into the contents of AC 0-7. The bit assignments are:
MMLF	6754	Load function register from the contents of AC 9-11, then clear the AC. The octal code of these three bits establishes the following DECTape control modes: 0 = Move 4 = Write data 1 = Search 5 = Write all bits 2 = Read data 6 = Write mark and timing bits 3 = Read all bits			AC0 = DT flag AC1 = Error flag AC2 = End (Selected tape at end point) AC3 = Timing error AC4 = Reverse tape direction AC5 = Go AC6 = Parity or mark channel error AC7 = Select error
MMSF	6761	Skip if DT flag is a 1.			
MMCC	6762	Clear memory address counter (MAC).			

CONTROL MODES

The seven modes of operation loaded into the function register during the MMLF command are used as follows:

Move

Initiates movement of the selected transport tape in either direction. Mark channel errors are inhibited in this mode.

Search

As the tape is moved in either direction, sensing

a block mark causes both the data flag and the DECTape (DT) flag to rise. The data flag causes a computer data break to deposit the block number in core memory at the address held in MAC. The DT flag initiates a program interrupt to cause the program to jump to a subroutine which is responsible for checking the block numbers by using either the block number stored during this operation or by counting the number of times the DT flag rises.

Read Data

A block of data is read in either direction, the data flag rises to cause a data break each time a 12-bit word is to be transferred, and the DT flag is raised to initiate a program interrupt at the end of the data block. The program is responsible for controlling tape motion at the end of a block transfer and must stop motion or change the contents of the function register when the DT flag rises. The transport continues reading until taken out of the read data mode.

Read All Bits

In this mode of operation, the three information channels are continuously read and transferred to the computer. This mode is similar to the read data mode except that the DT flag rises each time the data flag rises. The read all bits mode is used to read an unusual tape format which is not compatible with the read data mode. The DT flag does not cause an interrupt when in this mode.

Write Data

A block of data is written on tape in either direction, the data flag is raised to effect each transfer, and the DT flag is raised at the end of the block as in the read data mode.

Write All Bits

This special mode of operation is used to write information at all positions, disregarding blocks (such as in writing block numbers). The mode is similar to the read all bits mode for writing. The DT flag does not cause an interrupt in this mode.

Write Mark and Timing

This mode is used to write on the timing and mark channels to establish or change block length.

PROGRAMMED OPERATION

Prerecording of a reel of DECTape, prior to its use for data storage, is accomplished in two passes. During the first pass, the timing and mark channels are placed on the tape. During the second pass, forward and reverse block mark numbers, the standard data pattern, and the automatic parity checks are written. These functions are performed by the DECTOG program. Prerecording utilizes the write timing and mark channel control mode and a manual switch in the control which permits writing on the timing and mark channels, activates a clock which produces the timing channel recording pattern, and enables flags for program control. Unless both this control mode and switch are used simultaneously, it is physically impossible to write on the mark or timing channels. A red indicator lights on all transports associated with the control when the manual switch is in the ON position. Under these conditions only, the write register and write amplifier used to write on information channel 1 (bits 0, 3, 6, and 9) is used to write on the mark channel.

Two PDP-8 IOT microinstructions initiate operation of the DECTape system: the first, MMMM, loads the select register, motion register, and function register by means of instruction 6757 (combining MMLS, MMLM, and MMLF) and the second command, MMML (6766, combining MMCC and MMLC), loads the MAC with the core

memory address to be used to store the block number during searching. After initiating operation of the DECTape system, the program should always check for errors immediately by means of the MMSC instruction. This instruction should also be used at the conclusion of each transfer. A program should always start the DECTape system in the search mode to locate the block number selected for a transfer, then when the block number has been located, the transfer is accomplished by loading the function register with the read data or write data mode.

In searching, each block number is read by the transport and is transferred to the control. The control raises the DT flag upon receipt of each block number and stores the number in the computer core memory at the address contained in MAC. The computer program then samples the DT flag and either counts the number of blocks passed or reads the block number from core memory and compares it with the number it is seeking. The results of the data obtained in this way are used to further control the search operation. Upon determining that the forthcoming block is the one selected for a data transfer, the program loads the function register with either the read data or write data mode. Entering another mode discontinues the search mode. The starting address to be used for the first core memory address of the transfer is then set into the MAC by the computer.

When the start of the data position of the block is detected, the data flag is raised to initiate a data break each time the DECTape system is

ready to transfer a 12-bit word. Therefore, the main computer program continues running but is interrupted approximately every $133\frac{1}{3}$ microseconds during a data break for the transfer of a word. Transfers occur between DECTape and successive core memory locations, commencing at the address previously set into MAC. The number of words transferred is determined by the size of the selected tape block. At the conclusion of the block transfer, the DT flag is raised and a program interrupt occurs. The interrupt subroutine checks the DECTape error flag to determine the validity of the transfer and either initiates a search for the next information to be transferred or returns to the main program.

During all normal writing transfers, a checksum (the 6-bit exclusive OR of the words in the data block) is computed automatically by the control and is automatically recorded as one of the control words immediately following the data portion of the block. This same checksum is used during reading to determine that the data playback and recognition takes place without error.

Any one of the eight tape drives may be selected for use by the program. After using a particular drive, the program can stop the drive currently being used and select a new drive, or can select another drive while permitting the original selection to continue running. This is a particularly useful feature when rapid searching is desired, since several transports may be used simultaneously. Caution must be exercised however, for although the earlier drive continues to run, no tape end detection or other sensing

takes place. Automatic end sensing that stops tape motion occurs in all modes, but only in the selected tape drive.

Whenever either the motion or select code is changed, the program must wait until the DT flag is set to 1 before giving another motion or selection command. In other words, to prevent a timing error, all operations of the currently selected drive must be completed before issuing a new select code.

SPECIFICATIONS

Functional Specifications

Capacity

1. 260 feet of 0.75 inch, 1.0 Mylar tape per 3.5 inch reel.
2. 350 ± 55 lines per inch.
3. 849,036 usable lines per tape.
4. 60 lines per control zone.
5. 4096 is the maximum number of addressable blocks per reel.
6. The number of words in a block must be divisible by 3.
7.
$$N_B = \frac{212,112}{N_W + 15} - 2$$

N_B = decimal number of blocks
 N_W = number of words per block
8. Capacity for 190,000 12-bit words in blocks of 129 words.

Transfer Rate

1. One tape line is ready or written every $33\frac{1}{3}$ μ sec.
2. A 12-bit word is read and assembled or disassembled and written in $133\frac{1}{3}$ μ sec.
3. In reverse direction, the transfer rate varies by 20% as the effective reel diameter changes.
4. Transfers require 4.5% of PDP-5 cycles after the initial 200-msec start time.
5. Transfers require 1.2% of PDP-8 cycles after the initial 200-msec start time.

Addressing

1. Mark and timing channels allow searching for a particular block.
2. Start time is <300 msec, stop time is <150 msec, turn-around time is <300 msec.
3. Start and stop distances are approximately 8 inches.
4. When a command to reverse direction is issued at a given tape location, the system is up to speed when that same location passes the head after turn around.

Physical Specifications

555 Transport

1. Rack-Mounted Size – 10-1/2 inches high; 19 inches wide.
2. Table Model Size – 11-11/16 inches high; 20 inches wide; 15-1/4 inches deep.
3. Rack-Mounted Weight – 65 lbs.
4. Table Model Weight – 80 lbs.
5. Power Requirements – 115v, 60 cycles at 1.5 amp idle, and 3.2 amp maximum.

552 Control

1. Size – 21 inches high and 19 inches wide for a control which operates four Type 555 Dual Transports. (Requires three module mounting panels plus three standard 50-terminal interface connectors.)

2. Power Requirements – 115v, 60 cycles, 4 amp. The Type 728 Power Supply is included with the Type 552 Control.

Cabinet

A maximum of one Type 552 Control and three Type 555 Transports can be installed in a standard DEC computer cabinet. Specifications for this configuration including appropriate power control and power supplies are:

1. Size – 69-1/8 inches high; 22-1/4 inches wide; 27-1/16 inches deep. Minimum clearances for door openings are 8-3/4 inches at the front; 14-7/8 inches at the back.
2. Weight – 555 lbs.
3. Power Requirements – 115v, 60 cycle source capable of delivering 20 amp.

digital
EQUIPMENT
CORPORATION
MAYNARD, MASSACHUSETTS