

4

IDENTIFICATION

Product Code: Maindec-08-D3RA-D
Product Name: DECTREX 1
TC01 Random Exerciser
Maindec 851
Date: January 9, 1967
Maintainer: Diagnostic Group
Author: Keith F. Nelson

COPYRIGHT © 1967
DIGITAL EQUIPMENT CORPORATION

PDP-8
LIBRARY

1. ABSTRACT

DECTREX 1 is a DECTape Random Exerciser for the TC01 DECTape control and any configuration of one to eight TU55 DECTape transports. Drive selection, tape direction, number of blocks, sequence of operation and patterns generated are by random selection. The DECTape functions exercised are search, read data and write data in normal and continuous modes, read all in continuous mode, and move.

Also included are a short series of processor tests that are executed while waiting for interrupts and during data breaks while searching, reading, and writing from DECTape.

2. REQUIREMENTS

2.1 Equipment

PDP-8 (standard)

TC01 DECTape Control

One to eight TU55 DECTape Transports

One standard PDP-8 DECTape for each drive (2702₈ 129-word blocks)

2.2 Storage

2.2.1 Program Storage - The program occupies most of memory from address 0000 to 5000. In addition the JMS test uses addresses 6000 to 6200 for JMS storage.

2.2.2 Buffer Areas - The program utilizes three 129-word buffer areas as follows:

<u>Addresses</u>	<u>Contents</u>
6774 to 7174	Random data and output buffer
7175 to 7375	Read buffer 1
7376 to 7576	Read buffer 2

(Read buffers are used for WRITE DATA C MODE)

2.3 Preliminary Programs

All parts of the TC01 Basic Exerciser should run before attempting to run DECTREX 1.

3. LOADING PROCEDURE

3.1 Method

Procedures of normal binary loading from paper tape should be followed.

4. STARTING PROCEDURE

4.1 Control Switch Settings

When initially starting the program, SWITCH REGISTER bits 0 to 7 are used for drive selection. Each bit is a master bit for a drive. When the switch is a1 the drive is selected; when a0 the drive is not selected. Switch 0 is the master bit for drive 8, switch 1 for drive 1, etc. Any configuration of switches is considered valid except all 0s.

4.2 Starting Address

The starting address for DECTREX 1 is 0200.

4.3 Program and/or Operator Action

Load DECTREX 1 into memory.
 Dial the desired drive number(s) on each TU55 to be tested.
 Put each TU55 ON LINE, WRITE ENABLED with a standard PDP-8 DECTape installed.
 Set the SWITCH REGISTER to 0200.
 Press LOAD ADDRESS.
 Set the SWITCH REGISTER to select drives per paragraph 4.1.
 Press START.
 The processor halts at address 0207.
 Set all SWITCH REGISTER bits to 0 or as desired per paragraph 5.1.
 Press CONTINUE.

5. OPERATING PROCEDURE

5.1 Operational Switch Settings

SW0	UP	Delete error typeouts and halts.
SW1	UP	Delete error halts.
SW2	UP	Type first four data compare errors in each block.
SW2	DOWN	Type all data compare errors.

SW11	DOWN	Only hit end zone once for turnaround for blocks 0000 and 2701.
SW11	UP	Hit end zone twice before turnaround for blocks 0000 and 2701.

6. ERRORS

All DECtape hardware malfunctions detected by the program result in an error typeout and an error halt (see paragraph 5.1). The halt does not occur until all errors pertaining to the block and operation have been typed. (A read-data parity error and data compare error could occur in the same block. In this case halt would not occur until after the COMPARE ERROR typeout.)

The first three lines of every typeout indicate the DECtape drive, operation direction and mode, and the block being operated on or to be found as an end result of search.

6.1 Error Typeouts

6.1.1 Search Error Typeouts - The search error typeouts contain the following information:

- Drive number.
- Search direction and mode.
- Block wanted and direction.
- The block number put into memory by the TC01.
- The last block number found, if more than two blocks have been found.
- The number of blocks found since the last start-up or turnaround.
- DECtape status B.

Examine the typeout in the following order:

- a. Examine the direction of search and the direction of the block wanted. If they are different, the error occurred before turnaround. If the directions indicated are the same, the error occurred after turnaround.
- b. Examine the STAT B typeout. If it is 0001, indicating a normal interrupt (DECtape flag only), it can probably be ignored (6000 would indicate a mark track error). If STAT B is anything other than 0001, it caused the error typeout.
- c. Examine the number of block numbers read (BLOCKS READ). There are three conditions to note: 0000, indicating no block numbers read since start-up or turnaround; 0001, indicating one block number was read since start-up or turnaround; 0002 or greater, indicating that more than one block number was found.

d. Examine LAST BLOCK. This line is included only if two or more normal block numbers have been read since start-up or turnaround. This line is pertinent only if STAT B equals 0001.

e. Examine BLOCK FOUND. This line holds the contents of the memory location that block numbers are read into and is pertinent only if BLOCKS READ is not equal to 0000, or the STAT B typeout indicates that the DECTape flag is set (bit 11 a 1).

If STAT B is 0001 and BLOCKS READ is 0002 or greater, BLOCK FOUND compared against LAST BLOCK indicates two block numbers read that are not numerically consecutive.

If STAT B is not 0001 and BLOCKS READ is not 0000, BLOCK FOUND indicates the block where the status error occurred.

If STAT B is 0001, BLOCKS READ is 0001, and the direction of search is the same direction as the block wanted, the error was TURN AROUND.

The typeout C MODE indicates continuous mode. BLOCKS READ in a SEARCH C MODE typeout always indicates two blocks read. LAST BLOCK indicates the last block found in normal mode.

6.1.2 Write Data Error Typeouts - Write data typeouts contain the following information:

Drive number.
Direction and C mode if continuous mode block is being written.
DECTape status B.
Contents of the word count register (address 7754).

6.1.3 Read Data Typeouts -

6.1.3.1 Read Data Status Error Typeouts - These read data typeouts include the following information:

Drive number
Direction and mode
Block being read
DECTape status B if STAT B typed = 0001 see WC (word count register).
Contents of WC if it does not equal 0000.
(Note: If WC does not equal 0, no data compare is made.)

6.1.3.2 Read Data Compare Error Typeouts - These typeouts include the drive number, direction, mode, and block read. Each pair of data words is separated by a blank line of paper. The first octal number is the data generated or regenerated by the program. The second number is data read from tape, and the third line is the memory address of the data read.

6.1.4 Read All Error Typeouts -

6.1.4.1 Read All Status Errors - Read All Status Error typeouts include drive number, read all direction and mode, block number, and DECTape status B.

6.1.4.2 Read All CHECKSUM ERROR - This typeout indicates an error in parity generation (especially if not followed by a read-all compare error) and includes drive number, read-all direction and mode, block number, and the following:

a. REVERSE CHECKSUM as read from tape. If read-all direction is backward, the complement obverse of this line is the checksum going forward.

b. DATA CHECKSUM CALCULATED as generated by the 6-bit XOR of the data by the program. Since the data is written to be bidirectional, this line indicates the same sum for either direction.

c. CHECKSUM as read from tape is in the upper six bits of the word typed (bits 0 to 5). Again, if the read-all direction is backward, the complement obverse of these six bits would be the REV CKSUM if read forward.

d. LPB CALCULATED is the sum of the first three lines, and is generated by the same process as the TC01 parity generation circuitry. The LPB should equal 77_8 after the process is complete. The 0 bits in the lower six bits (bits 6 to 11) of this typeout are the error(s) that caused the typeout.

6.1.4.3 Read All Compare Error - These typeouts follow the same format as read-data compare errors. If a read-all compare error follows a checksum error, ignore the checksum error.

6.1.5 Program Interrupt Errors - The program also detects three classes of program interrupt errors.

a. Program interrupt and the DECTape IOT 771 did not skip. PI NO DECTAPE SKIP

b. No program interrupt for 45 seconds if a MOVE tape or for 5 seconds if any other DECTape function and DECTape IOT 771 did not skip at the end of that period.
NO PI NO DECTAPE SKIP

c. No program interrupt for 45 seconds if a MOVE tape or for 5 seconds if any other DECTape function and the DECTape IOT 771 did skip.
NO PI DECTAPE SKIP

Following one of these typeouts, the program forces another error typeout from the routine that called the wait for interrupt. The second typeout indicates the exact operation that the DECTape was doing when the PI error occurred.

6.1.6 Processor Errors - Any processor errors detected by the program result in an error halt only. Consult the program listing to determine the cause of the halt. The following table contains processor error halts and a description of the error:

<u>Address</u>	
3211	ISZ failed. Address 3342 should = 0.
3215	ISZ failed. Address 3341 should = 0001.
3226	ROTATE 1. Link should = 1.
3232	ROTATE 1. Data failure in AC address 3342 is data tested.
3242	ROTATE 2. Link should = 0.
3246	ROTATE 2. Data failure in AC. Address 3342 is data tested.
3272	TAD failure. Address 3347 plus address 3343 should = 3344.
3331	JMS failure. Address 3343 points to an address that does not equal its $\text{addr} + 1$ after executing a JMS.

6.1.7 Examples of Error Typeouts -

6.1.7.1 Example 1 Search Errors -

DRIVE B

SEARCH FWD

2677 BLOCK WANTED FWD

2700 BLOCK FOUND

0001 BLOCKS READ

0001 STAT B

(Block searched)

(Block found)

(One block number received)

(Normal interrupt)

This typeout indicates that the DECTape drive did not turn around and come up to speed in time.

DRIVE 8

SEARCH BKWD
0000 BLOCK WANTED FWD
1033 BLOCK FOUND
0000 BLOCKS READ
6000 STAT B

(Error was a mark-track error if STAT B is not normal interrupt and BLOCKS READ = 0000. Ignore second block number)

DRIVE 2

SEARCH BKWD
0077 BLOCK WANTED BKWD
0105 BLOCK FOUND
0000 BLOCKS READ
5000 STAT B

This timeout occurs if an end zone interrupt is received in error.

DRIVE 6

SEARCH FWD
2701 BLOCK WANTED FWD
0000 BLOCK FOUND
7700 LAST BLOCK
0002 BLOCKS READ
0001 STAT B

This timeout indicates that two block numbers were found in search and that they were not sequential.

6.1.7.2 Example 2 Read Data Status Error -

DRIVE 4

READ DATA FORWARD
0265 BLOCK
4200 STAT B

(Block being read)
(Parity error)
(If any data errors, a second timeout follows)

DRIVE 3

READ DATA BACKWARD
0011 BLOCK
4400 STAT B
7577 WC

(Block being read)
(Select error)
(WC was not = 0. The program makes no data comparison in this case.)

DRIVE 5

READ DATA FWD C MODE
0227 BLOCK
0001 STAT B
7577 WC

(Note: Normal STAT B)
(Error was WC did not go to 0000 before DTF was set)

6.1.7.3 Example 3 Write Data Status Error -

DRIVE 1

WRITE DATA FWD
0001 BLOCK
6000 STAT B
7743 WC

A mark-track error was received while doing a write data on block 1.

DRIVE 1

WRITE DATA BKWD C MODE
0040 BLOCK
0001 STAT B
7400 WC

(Note: Normal STAT B)
(WC indicates that WC overflow had not occurred when DTF was set).

6.1.7.4 Example 4 Read all Checksum Error -

DRIVE 6

READ ALL BACKWARD C MODE
0175 BLOCK
CHECKSUM ERROR

0033 REV CKSUM
0022 DATA CKSUM CALCULATED
6700 CKSUM
0076 CALCULATED CKSUM

(The complements of the REVCK, DATA and CKSUM did not XOR to 77)
(as read from tape)
(XOR of the data's complement)
(as read from tape)
(XOR of the complement of the above. Any 0 bits are in error. Calculated CKSUM should = 0077.)

NOTE: In this case, the REV CKSUM would be the CKSUM going forward. Realize also, that it would be complement obverse (44) going forward.

Since the DATA FORMAT written by DECTREX is bidirectional, the DATA CHECKSUM CALCULATED would be the same read in either direction.

Also, the CKSUM going forward would be the REV CKSUM; the complement obverse of 67 would be 01.

Note that the calculated checksum would be 67, if the block had been read forward.

6.1.7.5 Example 5 Read Compare Error - Read-data and read-all-data compare errors follow the same format.

DRIVE 4

READ DATA FORWARD
DATA ERROR
0265 BLOCK

(or READ ALL C MODE)

4632 COR
4432 INC
7237 ADRS INC

(Pattern word generated)
(Pattern word read)
(Memory address of the incorrect data)

2315 COR
2115 INC
7240 ADRS INC

6.2 Error Recovery

After an error typeout, the processor halts. Press CONTINUE to recover. For all errors, the program attempts to repeat the same or a similar operation. If a second error is encountered for the same operation, the typeout and HALT again occurs. However, pressing CONTINUE the second time causes the drive in error to rewind, and the random selections for that drive begin again from block 0. If a second error is not encountered, the program completes the operation and then continues its normal random selections.

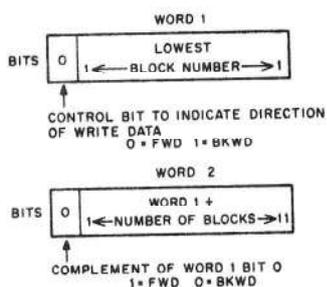
For the read errors (read data and read all) the direction of the second read operation is opposite the direction that the error was originally detected in. If a second error is detected, a typeout and halt again occur. Pressing CONTINUE causes the drive to be rewound to the end zone. If no error occurs during the second pass, the block is read a third time. The third read is in the same direction as the first read, or the direction the error was originally detected in. Again, if any errors are detected, a typeout and halt occur. In either case, the end result is that the drive is rewound and random selections start from block 0, so that the same read error sequence is not generated again.

7. RESTRICTIONS (None)

8. MISCELLANEOUS

8.1 Data Format

The data blocks written by DECTREX 1 are formatted to be bidirectional. Whether written forward or backward, they may be read in either direction on the TC01 with the resultant data looking the same. The first four words of the block contain pertinent information about the block. The first two words of each block are formatted as follows:



WORD 1 indicates the lowest numbered block in the series with bit 0 indicating the written direction of the series of blocks.

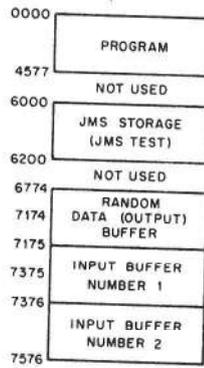
WORD 2 is 1 greater than the highest numbered block in the series bit 0 in word 2 is the complement of bit 0 in WORD 1.

Ignoring bits 0 of the two words, WORD 2 - WORD 1 = number of blocks in the series.

Word 3 and word 4 of the block are random numbers used to generate the rest of the data. Word 5 of the block is word 3 rotated right one position. Word 6 is word 4 rotated right one position. This process is repeated until word 64 of the block is word 62 rotated right. The lower six bits of word 65 of the block contain the number of 1 bits in words 3 and 4. The upper six bits of word 65 (middle word of 129) are the complement obverse of the lower six bits. Word 66 is the complement obverse of word 64. Word 67 is the complement obverse of word 63, etc., until the last word of the block is the complement obverse of the first word of the block.

WORD 1	4036	Blocks 36, 37, 40, and
WORD 2	0042	41 were written backward
WORD 3	3252	with these two words
WORD 4	7734	as key to data pattern.
WORD 5	1525	WORD 3 RAR
WORD 6	3756	WORD 4 RAR
	etc.	WORD 5 RAR WORD 6 RAR
WORD 65	0617	Number of bits in words 3 and 4 in bits 6 to 11. Complement obverse of same in bits 0 to 5.
WORD 126	3400	Complement obverse of word 4
WORD 127	1204	Complement obverse of word 3
WORD 128	5377	Complement obverse of word 2
WORD 129	1473	Complement obverse of word 1

8.2 Core Map



9. PROGRAM DESCRIPTION

9.1 Discussion

DECTREX 1 is a random exerciser for the TC01 DECTape Control and any configuration of one to eight TU55 DECTape Drives. Drive, direction of operation, number of blocks, and data patterns are by random selection. First the program randomly selects a drive, then a number between 1 and 32 decimal for the number of blocks, and next the direction. There is one possibility in four that the direction will be backward. The number of blocks is added to or subtracted from the last block position of the drive selected. If the block generated has not been written, a write operation is initiated. If the last block table indicates that the block selected has already been written, a read operation is initiated. If READ is selected, the program then generates a random number between 0 and 7. If the number is 0, the block is read in read-all continuous mode; if 4, read-data normal mode; if any other number, read data continuous mode and two blocks are read. If WRITE is selected and the number of blocks is a multiple of 3 (3, 6, 9, etc.), the blocks are written in continuous mode. Otherwise, the blocks are written in normal mode.

The processor tests, that are run while waiting for interrupts do not guarantee that the processor operates correctly; but, other than EAE operations, should allow a reasonable amount of confidence in the processor.



```

WRITE1 1400
WRTDTY 0121
WRT1A 1455
WSTERR 1526
WTCNTR 3125
WTKON1 3123
WTKON2 3124
XORSAV 2565
XORSUM 2531

```

```

/TC01 RANDOM EXERCISER
/PAGE 0 CONSTANTS AND TEMP STOR.
RANBFR=6774
BUFFRS=RANBFR+201
IOT=6000

```

```

/GO TO PROCESS INTERRUPT

```

```

*1
0001 5402 JMP I .+1
0002 3034 IRECD

```

```

*40

```

```

0040 0000 RECORD, 0 /SEARCH ID
0041 0000 BLKFND, 0 /BLOCK FROM TC01
0042 0041 IDCON, .-1 /FOR SRCH CA
0043 7754 WCLOC, 7754 /TO STORE WC
0044 7755 CALOC, 7755 /TO STORE CA
0045 0000 CDRIVE, 0 /CURRENT DRIVE NUMBER
0046 0000 JNFUNC, 0 /SAME FOR TC01 COMMAND

0047 0000 MSBITS, 0 /DRIVE SELECT SWITCHES
0050 0000 COMBIT, 0
0051 0000 POSITN, 0 /TO GET DRIVE POSITION
0052 0000 DIRECT, 0 /TO GET DRIVE DIRECTION
0053 0000 -STBLK, 0 /TO GET LAST BLOCK WRITTEN
0054 0000 -STDRV, 0 /-LAST DRV SELECTED

```

/SUBROUTINE ADDRESSES

0055	3600	SRCHIT,	SEARCH
0056	3400	REWIND,	REPOSI
0057	3437	NEWDRV,	CHNGDR
0060	1000	SELRAN,	RANSEL
0061	0600	GENPAT,	PATGEN
0062	0735	RANGEN,	GENRAN
0063	0400	READ1,	RODATA
0064	0547	REGENP,	PREGEN
0065	1225	JATACO,	CODATA
0066	3000	WAITI,	WATINT
0067	4321	ERRSTP,	ERSTP
0070	1200	DATAMV,	MVDATA
0071	0040	RECRDK,	RECORD
0072	7777	COCNTR,	7777
0073	6774	RBUFST,	RANBFR
0074	6773	RBFLOC,	RANBFR-1
0075	7174	BF1LOC,	BUFFRS-1
0076	0000	JIRFLG,	0
0077	0000	VUMBLK,	0
0100	7174	RBUFND,	RANBFR+200
0101	4441	WRTDTY,	TYWDAT
0102	6775	RBFWD2,	RANBFR+1
0103	7376	BF2LOC,	BUFFRS+201
0104	0420	READ2,	REREAD
0105	0470	READ2A,	REREDA
0106	4272	TYPCON,	TYCONT
0107	4200	TYPTX,	TYTEXT

/-1 NO DATA ERRORS

/MASK AND COMPARE CONSTANTS

0110	7760	<7760,	7760
0111	0037	<0037,	37
0112	7770	<7770,	7770
0113	0200	<0200,	200
0114	0040	<0040,	40
0115	0614	<0614,	614
0116	0030	<0030,	30
0117	7577	<7577,	7577
0120	0007	<0007,	7
0121	4000	<4000,	4000
0122	0700	<0700,	700
0123	0070	<0070,	70
0124	7000	<7000,	7000
0125	0003	<0003,	3
0126	0214	<0214,	214
0127	7700	<7700,	7700
0130	4511	JRIVTY,	TYDRV
0131	1000	<1000,	1000
E=ZBI T=K1000			
0132	0010	<0010,	10
0133	0020	<0020,	20
0134	0077	<0077,	77
0135	0100	<0100,	100
0136	0240	<0240,	240
0137	0400	<0400,	400
0140	0604	<0604,	604
0141	0016	<16,	16
0142	0260	<260,	260
0143	3777	<3777,	3777
0144	4423	READTY,	TYRDAT
0145	4411	SRCHTY,	TYSRCH
0146	5075	<5075,	5075
0147	5076	<5076,	5076
0150	0050	<0050,	50
0151	0000	PASFLG,	0
0152	0002	<0002,	2
0153	2701	<2701,	2701
0154	6000	<6000,	6000

/CONSTANTS FOR SEARCH ROUTINE

0155	4000	SRCHER,	SRHERR
0156	0000	TAPONT,	0
0157	0000	BLKFLG,	0
0160	0000	PREBLK,	0

/JMS RETURN FOR PROCESSOR TEST

0161	3321	RETUJM,	JMRETU
0162	0000	FRSWAT,	0

/TC01 DECTAPE
/RANDOM DATA RANDOM SEQUENCE EXERCISER
/TESTS CONCURRENT OPERATION OF ANY CONFIGURATION OF 1 TO 8 DECTAPE
/DRIVES.
/MASTER HIT SELECTION BIT 0=DRV8 BIT1=DRV1 TO BIT7=DRV7

*200

```
0200 7604  RANDEX,      CLA OSR
0201 0110      AND K7760
0202 7440      SZA          /SELECT ANY AT ALL
0203 5206      JMP ,+3      /YES
0204 7402      HLT          /DIT NOT SELECT ANY DRIVES
0205 5200      JMP RANDEX
0206 3047      DCA MSBITS
0207 7402      HLT          /WAIT CLR SELECTION
0210 7040      CMA
0211 3152      DCA FRSWAT
0212 6774      IOT 774     /CLEAR STAT B
0213 4456      JMS I REWIND /PUT ALL DRIVES IN END ZONE

0214 3453      DCA I LSTBLK /CLR NUMBER OF BLOCKS
0215 4457      JMS I NEWDRV /CLR ALL
0216 5214      JMP , -2    /NO
0217 7201      IAC CLA
0220 3054      DCA LSTDRV
```

/MEAT OF THE PROGRAM
/RANDOM SELECTION OF OPERATIONS
/FIRST SELECT DRIVE

```
0221 4460  MOFPRO,      JMS I SELRAN  /RANDOMLY SELECT A DRIVE
0222 7240      CLA CMA
0223 3151      DCA PASFLG
0224 1045      TAD CDRIVE
0225 1054      TAD LSTDRV
0226 7650      SNA CLA
0227 5325      JMP SAMDRV
0230 1045      TAD CDRIVE
0231 7041      CMA IAC
0232 3054      DCA LSTDRV

0233 7040      CMA
0234 3350      DCA RS0FLG
          /SELECT 1 TO 32 BLOCKS FOR OPERATION
0235 4462  MOFPR1,      JMS I RANGEN  /GET RAN NUMBER
0236 0111      AND K0037
0237 7040      CMA          /MAKE -1 TO -32 DECI
0240 3077      DCA NUMBLK
```

```

/SELECT DIRECTION TO GO
0241 4462 JMS I RANGEN /GET RANDOM NUMBER
0242 0125 AND K0003
0243 7650 SNA CLA
0244 7040 CMA /SET BACKWARDS
0245 3076 DCA DIRFLG /DIRECTION FLAG
0246 1453 TAD I LSTBLK
0247 7650 SNA CLA /DRIVE BEEN WRITTEN ON
0250 5724 JMP I WRITES /NO, WRITE
0251 7100 CLL
0252 1076 TAD DIRFLG

```

```

/GENERATE BLOCK SELECT POS + OR-NUMBER
0253 7640 SZA CLA
0254 7120 STL
0255 1077 TAD NUMBLK
0256 7420 SNL
0257 7041 CMA IAC
0260 1451 TAD I POSITN
0261 3040 DCA RECORD
0262 1040 TAD RECORD
0263 7710 SPA CLA /-RECORD
0264 3040 DCA RECORD /MAKE =0

```

```

/HAS BLOCK SELECTED BEEN WRITTEN
/IF NOT WRITE OPERATION SELECTED

```

```

0265 1040 TAD RECORD
0266 7040 CMA
0267 1453 TAD I LSTBLK
0270 7710 SPA CLA /BLOCK BEEN WRITTEN
0271 5724 JMP I WRITES /NO
0272 4462 JMS I RANGEN
0273 0120 AND K0007
0274 7450 SNA
0275 5703 JMP I .+6
0276 0125 AND K0003
0277 7650 SNA CLA
0300 5304 JMP .+4
0301 5702 JMP I .+1
0302 2000 RDCMOD
0303 2400 RALLTS
0304 4463 RDTAB, JMS I READ1
0305 5504 JMP I READ2 /STATUS ERROR RETURN

```

```

/BLOCK HAS BEEN READ WITHOUT PAR ERR
0326 1113 TAD K0200
0327 6764 IOT 764 /STOP TAPE
0310 1040 TAD RECORD
0311 3451 DCA I POSITN /NEW POSITION
0312 1076 TAD DIRFLG
0313 3452 DCA I DIRECT /DIRECTION READ

/MOVE FIRST FOUR WORDS TO REGENERATE
0314 4464 JMS I REGENP /REGERATE DATA PATTERN
0315 4465 JMS I DATACO
0316 6774 RANBFR
0317 7175 BUFFRS
0320 0201 201
0321 2072 ISZ COCNTR
0322 5505 JMP I READ2A
0323 5221 JMP MOFPRO /SELECT NEW DRIVE
0324 1400 WRITES, WRITE1

/DRIVE SELECTED SAME AS LAST TIME TEST FOR SERIES
0325 4462 SAMDRV, JMS I RANGEN
0326 7510 SPA /READ THIS STRING
0327 5236 JMP MOFPR1+1 /FIND NEW BLOCK
0330 2350 ISZ RSQFLG
0331 5236 JMP MOFPR1+1
0332 0125 AND K0003
0333 7650 SNA CLA /FWD
0334 7040 CMA /NO GO BACKWARD
0335 3076 DCA DIRFLG
0336 1076 TAD DIRFLG
0337 7041 CMA IAC
0340 1073 TAD RBUFST /FWD GET
0341 3040 DCA RECORD /FIRST BLOCK
0342 1076 TAD DIRFLG /BKWD GETS
0343 1440 TAD I RECORD /LST BLOCK+1-1
0344 0143 AND K3777
0345 3040 DCA RECORD
0346 5747 JMP I .+1
0347 1600 RDSEQ
0350 0000 RSGFLG, 0

```

/READ DATA SUBROUTINE 1 BLOCK
/DIRFLG=7777 IS BACKWARDS=0 IS =WJ

*400

0400	5200	RDDATA,	JMP .	
0401	7200	CLA		
0402	4455	JMS I SRCHIT		
0403	1116	TAD K0030	/SEARCH TO READ DATA	
0404	6764	IOT 764		
0405	1117	TAD K7577	/129 WORDS	
0406	3443	DCA I WCLOC		
0407	1075	TAD BF1LOC	/INTO FIRST BUFFER	
0410	3444	DCA I CALOC		
0411	4466	JMS I WAITI		
0412	0001	1		
0413	5600	JMP I RDDATA	/STATJS ERR(READ EXIT)	
0414	1443	TAD I WCLOC		
0415	7650	SNA CLA		
0416	2200	ISZ RDDATA		
0417	5600	JMP I RDDATA	/NORMAL READ EXIT	

/TYPE STATUS ERROR ON READ 1 BLOCK
 /TEST FOR DATA ERRORS IF FULL READ

0420	7242	REREAD,	CLA CMA	
0421	3191	DCA PASFLG		
0422	4457	JMS I ERRSTR	/STOP TAPE	
0423	4544	JMS I READTY		
0424	1271	TAD RECRDK		
0425	4506	JMS I TYPCON	/TYPE BLOCK NUMBER	
0426	4507	JMS I TYPTX		
0427	2242	0042		
0430	5457	5457		
0431	4353	4353		
0432	7700	7700		
0433	5772	10T 772		
0434	3010	DCA 10		
0435	1132	TAD K0010		
0436	4506	JMS I TYPCON	/TYPE STATUS B	
0437	4507	JMS I TYPTX		
0440	2053	0063		
0441	6441	6441		
0442	6400	6400		
0443	4200	4200		
0444	7777	7777		
0445	7700	7700		
0446	1443	TAD I WCLOC		
0447	7650	SNA CLA	/READ 129 WORD	
0450	5260	JMP .+10	/YES	
0451	1043	TAD WCLOC		
0452	4506	JMS I TYPCON	/SHORT BUFFER	
0453	4507	JMS I TYPTX	/TYPEOUT	
0454	2057	0067		
0455	1643	1643		
0456	1677	1677		
0457	5270	JMP REREDA		
0460	4464	JMS I REGENP	/REGENERATE PATTERN	
0461	1444	TAD I CALOC		
0462	1343	TAD K7600		
0463	3266	DCA .+3		
0464	4455	JMS I DATACD	/COMPARE FOR TYPEOUTS	
0465	6774	RANBFR		
0466	7175	BUFFRS		
0467	2201	0201		

0470	7624	REREDA,	LAS	
0471	0154	AND	K6200	
0472	7650	SNA	CLA	
0473	7402	HLT		
0474	2151	ISZ	PASFLG	
0475	5325	JMP	REWDRV	
0476	1076	TAD	DIRFLG	
0477	7040	CMA		
0500	3076	DCA	DIRFLG	
0501	4463	JMS	I READ1	/READ OPPOSITE DIRECTION
0502	5222	JMP	REREAD+2	/ERROR, AGAIN, TRY OVER
0503	4464	JMS	I REGENP	/REGEN PATTERN
0504	4465	JMS	I DATACO	/COMPARE
0505	6774	RANBFR		
0506	7175	BUFFRS		
0507	0201	0201		
0510	2072	ISZ	COCNTR	
0511	5270	JMP	REREDA	
0512	1076	TAD	DIRFLG	
0513	7040	CMA		
0514	3076	DCA	DIRFLG	
0515	4463	JMS	I READ1	
0516	5222	JMP	REREAD+2	/ANOTHER STATUS ERROR
0517	4465	JMS	I DATACO	
0520	6774	RANBFR		
0521	7175	BUFFRS		
0522	0201	0201		
0523	2072	ISZ	COCNTR	
0524	5270	JMP	REREDA	
0525	1140	REWDRV,	TAD K0604	
0526	1046	TAD	UNFUNC	
0527	6766	IOT	766	
0530	4466	JMS	I WAITI	
0531	5000	5000		
0532	5344	JMP	K7600+1	
0533	3453	DCA	I LSTBLK	
0534	3451	DCA	I POSITN	
0535	7040	CMA		
0536	3452	DCA	I DIRECT	
0537	7001	IAC		
0540	3054	DCA	LSTDRV	
0541	5742	JMP	I .+1	
0542	0221	MOFPRO		
0543	7600	K7600,	7600	
0544	4746	JMS	I .+2	
0545	5325	JMP	REWDRV	
0546	3542	MOVER+2		

/MOVE FIRST 4 WORDS OF
 /BLOCK READ AND REGENERATE
 /COMPARE PATTERN

0547	5347	PREGEN,	JMP ,
0552	1274	TAD	RBFL0C
0551	3210	DCA	10
0552	1372	TAD	GETRED+2
0553	3217	DCA	17
0554	1444	TAD	I CALOC
0555	3370	DCA	GETRED
0556	1770	TAD	I GETRED
0557	4771	JMS	I GETRED+1
0560	3410	DCA	I 10
0561	7040	CMA	
0562	1370	TAD	GETRED
0563	3370	DCA	GETRED
0564	2017	ISZ	17
0565	5336	JMP	,-7
0566	4461	JMS	I GENPAT
0567	5747	JMP	I PREGEN
0570	2020	GETRED,	0
0571	0703	MCOMOB	
0572	7774	7774	

/GENERATE RANDOM DATA PATTERN
 /FIRST FOUR WORDS OF PATTERN ARE IN
 /RANDOM BUFFER WORDS 1 TO 4

```

*600
0600 5200 PATGEN,      JMP      .
0601 1353          TAD     MIN4
0602 3275          DCA     GENDEX      /TO COUNT FIRST 4 WORDS
0603 1073          TAD     RBUFST
0604 3276          DCA     GENDEX+1    /TO STORE FROM START
0605 1100          TAD     RBUFND
0606 3277          DCA     GENDEX+2    /TO STORE FROM END
0607 1301          TAD     RPATO+1    /TO GET
0610 3300          DCA     RPATO              /RANDJMS
0611 1676          TAD     I GENDEX+1
0612 3301          DCA     RPATO+1
0613 1301          TAD     RPATO+1
0614 4303          JMS     MCOMOB              /MAKE COMPLIMENT OBVERSE
0615 3677          DCA     I GENDEX+2
0616 2276          ISZ     GENDEX+1    /INC ADDRESS
0617 7240          CLA     CMA
0620 1277          TAD     GENDEX+2    /DECREMENT OTHER
0621 2275          ISZ     GENDEX      /DONE FIRST 4
0622 5206          JMP     PATGEN+6    /NO

0623 3277          DCA     GENDEX+2
0624 1354          TAD     MIN60              /TO COUNT 60 WORDS
0625 3275          DCA     GENDEX
0626 1300          GNPATR,      TAD     RPATO
0627 7110          CLL     RAR              /FORM NEXT
0630 7430          SZL              /WORD OF
0631 1121          TAD     K4000              /RANDJM PATTERN
0632 3302          DCA     RPATO+2

0633 1302          TAD     RPATO+2
0634 3676          DCA     I GENDEX+1
0635 1302          TAD     RPATO+2
0636 4303          JMS     MCOMOB
0637 3677          DCA     I GENDEX+2
0640 1301          TAD     RPATO+1    /MOVE WORDS
0641 3300          DCA     RPATO              /FOR NEXT PASS
0642 1302          TAD     RPATO+2
0643 3301          DCA     RPATO+1
0644 2276          ISZ     GENDEX+1    /INCREMENT LOWER
0645 7240          CLA     CMA
0646 1277          TAD     GENDEX+2    /DECREMENT UPPER ADDRESS
0647 3277          DCA     GENDEX+2
0650 2275          ISZ     GENDEX      /DONE ALL
0651 5226          JMP     GNPATR      /NO

```

0652	3302	DCA	RPATO+2	/CLR FOR COUNTING BITS
0653	1300	TAD	RPATO	
0654	7110	CLL	RAR	
0655	7430	SZL		/BIT=1
0656	2302	ISZ	RPATO+2	/YES, COUNT
0657	7440	SZA		/DONE, FIRST WORD
0660	5254	JMP	,-4	/NO
0661	1301	TAD	RPATO+1	/GET 2ND
0662	7110	CLL	RAR	
0663	7430	SZL		/BIT=1
0664	2302	ISZ	RPATO+2	/YES
0665	7440	SZA		/DONE 2ND
0666	5262	JMP	,-4	/NO
0667	1302	TAD	RPATO+2	
0670	4303	JMS	MCOMOB	/MAKE OBVERSE
0671	0127	AND	K7700	
0672	1302	TAD	RPATO+2	/MAKE WORD 65
0673	3676	DCA	I GENDEX+1	
0674	5600	JMP	I PATGEN	/EXIT
0675	0000	GENDEX,	0	/TO COUNT WORDS
0676	0000		0	/TO STORE LOWER
0677	0000		0	/TO STORE UPPER
0700	0000	RPATO,	0	/PATTERN STORAGE
0701	0000		0	
0702	0000		0	/TO COUNT BITS

/MAKE COMPLIMENT OBVERSE OF AC

```

0703 5303 MCOMOB,    JMP .
0704 3355      DCA COMSTR    /SAVE ORIGINAL
0705 1355      TAD COMSTR
0706 0122      AND K0702
0707 7110      CLL RAR
0710 7012      RTR
0711 3356      DCA COMSTR+1
0712 1355      TAD COMSTR

0713 0123      AND K070
0714 7006      RTL
0715 7004      RAL
0716 1356      TAD COMSTR+1
0717 3356      DCA COMSTR+1
0720 1355      TAD COMSTR
0721 0120      AND K0007
0722 7012      RTR
0723 7012      RTR
0724 1356      TAD COMSTR+1
0725 3356      DCA COMSTR+1
0726 1355      TAD COMSTR
0727 0124      AND K7000
0730 7006      RTL
0731 7006      RTL
0732 1356      TAD COMSTR+1
0733 7040      CMA
0734 5703      JMP I MCOMOB

```

/RANDOM NUMBER GENERATOR

```

0735 5335 GENRAN,    JMP .
0736 7200      CLA
0737 1352      TAD RANVAR
0740 7104      CLL RAL
0741 7430      SZL
0742 7001      IAC
0743 3352      DCA RANVAR
0744 1352      TAD RANVAR
0745 1351      TAD RANN0
0746 3351      DCA RANN0
0747 1351      TAD RANN0
0750 5735      JMP I GENRAN    /EXIT AC=RAN0JM

0751 2634 RANN0,      2634
0752 4263 RANVAR,     4263
0753 7774 MIN4,      7774
0754 7724 MIN60,     7724
0755 0000 COMSTR,    0

```

/RANDOMLY SELECT A DRIVE
 /STAYS IN THIS ROUTINE UNTIL A DRIVE
 /IS FOUND AND DIRECTION AND POSITION
 /POINTERS ARE GENERATED

```

*1000
1000 5200 RANSEL,      JMP .
1001 4246      JMS SELGEN          /GET RANDOM NUMBER
1002 0120      AND K0007          /SELECT DRIVE
1003 3045      DCA CDRIVE
1004 1045      TAD CDRIVE
1005 7110      CLL RAR
1006 7012      RTR
1007 7010      RAR

1010 3046      DCA UNFUNC        /POSITION DRIVE NUMBER
1011 1045      TAD CDRIVE
1012 7040      CMA
1013 3010      DCA 10           /MAKE NEG FOR COUNT
1014 1121      TAD K4000
1015 3011      DCA 11
1016 2010      ISZ 10          /BIT IN DRIVE POSITION
1017 7410      SKP

1020 5225      JMP EXIST        /BIT IS IN POSITION
1021 1011      TAD 11
1022 7110      CLL RAR        /MOVE UNIT BIT
1023 3011      DCA 11
1024 5216      JMP , -6
1025 1011      EXIST,          TAD 11      /GET UNIT BIT
1026 0047      AND MSBITS      /MASK WITH DRIVES SELECT
1027 7650      SNA CLA        /DOES DRIVE EXIST

1030 5201      JMP RANSEL+1    /NO, TRY AGAIN
1031 1045      TAD CDRIVE
1032 1243      TAD POSTBL
1033 3051      DCA POSITN      /FORM POSITION POINTER
1034 1045      TAD CDRIVE
1035 1244      TAD DIRTBL
1036 3052      DCA DIRECT      /FORM DIRECTION POINTER
1037 1045      TAD CDRIVE
1040 1245      TAD LSTTBL      /GENERATE LAST BLOCK
1041 3053      DCA LSTBLK      /WRITTEN POINTER
1042 5600      JMP I RANSEL

```

1043	3502	POSTBL,	PNTRS+1	/TO GET DRIVE POSITION
1044	3513	JIRTBL,	PNTRS+12	/DIRECTION
1045	3524	LSTTBL,	PNTRS+23	/LAST BLOCK WRITTEN
1046	5246	SELGEN,	JMP .	
1047	1261	TAD	SELVAR	
1050	7104	CLL	RAL	
1051	7430	SZL		
1052	7001	IAC		
1053	3261	DCA	SELVAR	
1054	1261	TAD	SELVAR	
1055	1262	TAD	SELVAR+1	
1056	3262	DCA	SELVAR+1	
1057	1262	TAD	SELVAR+1	
1060	5646	JMP	I SELGEN	
1061	4263	SELVAR,	4263	
1062	2634		2634	

/MOVE DATA SUBROUTINE
 /FROM ADDRESS=JMS+1 TO IS JMS+2
 /NUMBER OF WORDS IS JMS+3

	*1200			
1200	5200	MVDATA,	JMP .	
1201	7200	CLA		
1202	1600	TAD	I MVDATA	
1203	2200	ISZ	MVDATA	
1204	3222	DCA	MOVDEX	
1205	1600	TAD	I MVDATA	
1206	2200	ISZ	MVDATA	
1207	3223	DCA	MOVDEX+1	
1210	1600	TAD	I MVDATA	
1211	2200	ISZ	MVDATA	
1212	3224	DCA	MOVDEX+2	
1213	1622	TAD	I MOVDEX	
1214	3623	DCA	I MOVDEX+1	
1215	2222	ISZ	MOVDEX	
1216	2223	ISZ	MOVDEX+1	
1217	2224	ISZ	MOVDEX+2	
1220	5213	JMP	,-5	
1221	5600	JMP	I MVDATA	
1222	0000	MOVDEX,	0	
1223	0000		0	
1224	0000		0	

/COMPARE DATA SUBROUTINE
 /JMS+1=ADDRESS JMS+2=ADDRESS JMS+3=WC

```

1225 5225 CODATA,    JMP .
1226 7200      CLA
1227 1625      TAD I CODATA    /GET ADDRESS 1 CORRECT
1230 2225      ISZ CODATA
1231 3255      DCA CODEX
1232 1625      TAD I CODATA    /GET ADDRESS 2 UNKNOWN
1233 3256      DCA CODEX+1
1234 2225      ISZ CODATA

1235 1625      TAD I CODATA    /GET LENGTH
1236 7041      CMA IAC
1237 3257      DCA CODEX+2
1240 2225      ISZ CODATA
1241 7040      CMA
1242 3072      DCA COCNTR    /SET NO ERROR FLAG
1243 1655      COLOOP,      TAD I CODEX
1244 7041      CMA IAC
1245 1656      TAD I CODEX+1

1246 7640      SZA CLA        /WORDS =
1247 5261      JMP COERRO    /NO, TYPE OJT
1250 2255      ISZ CODEX      /INCREMENT ADDRESS
1251 2256      ISZ CODEX+1
1252 2257      ISZ CODEX+2    /DONE ALL
1253 5243      JMP COLOOP    /NO
1254 5625      JMP I CODATA   /EXIT

1255 0000      CODEX,      0          /KNOWN DATA ADDRESS
1256 0000      0          /UNKNOWN DATA ADDRESS
1257 0000      0          /LENGTH
1260 1256      .#2

```

/DATA ERROR PRINTOUT

```

1261 2072      COERRO,      ISZ COCNTR    /FIRST DATA ERROR
1262 5313      JMP COERR1    /NOT FIRST
1263 4467      JMS I ERRSTP
1264 6761      IOT 761
1265 0132      AND K0010
1266 7650      SNA CLA
1267 5273      JMP .+4
1270 4672      JMS I .+2
1271 5274      JMP .+3
1272 4432      TYRALL
1273 4544      JMS I READTY    /FIRST ERROR
1274 4507      JMS I TYPTX    /TYPE HEADER

```

1275	7777	7777	
1276	4441	4441	/TYPE
1277	6441	6441	/CDATA ERROR)
1300	0045	0045	
1301	6262	6262	
1302	5762	5762	
1303	7700	7700	
1304	1071	TAD RECRDK	/TYPE THE OCTAL
1305	4506	JMS I TYPCON	/BLOCK NUMBER
1306	4507	JMS I TYPTX	
1307	0042	0042	
1310	5457	5457	/TYPE (BLOCK)
1311	4353	4353	
1312	7700	7700	
1313	7604	COERR1, CLA OSR	
1314	0131	AND K1000	
1315	7650	SNA CLA /ONLY TYPE4	
1316	5323	JMP ,+5 /TYPE ALL ERRORS	
1317	1072	TAD COCNTR	
1320	1354	TAD K7774	
1321	7700	SMA CLA /DONE 4 DATA TYPEOUTS	
1322	5250	JMP COLOOP+5	/YES, DELETE REST
1323	4507	JMS I TYPTX	
1324	7777	7777	/BLANK LINE BETWEEN
1325	7700	7700	/EACH PAIR
1326	1255	TAD CODEX	
1327	4506	JMS I TYPCON	/TYPE CORRECT IN OCTAL
1330	4507	JMS I TYPTX	
1331	0043	0043	/TYPE (COR)
1332	5762	5762	
1333	7700	7700	
1334	1256	TAD CODEX+1	
1335	4506	JMS I TYPCON	/TYPE INCOR OCTAL
1336	4507	JMS I TYPTX	/TYPE (INC)
1337	0051	0051	
1340	5643	5643	
1341	7700	7700	
1342	1260	TAD CODEX+3	
1343	4506	JMS I TYPCON	
1344	4507	JMS I TYPTX	
1345	0041	41	
1346	4444	4444	
1347	6263	6263	
1350	0051	51	
1351	5643	5643	
1352	7700	7700	
1353	5250	JMP COLOOP+5	
1354	7774	<7774, 7774	

/INITIATE WRITE OPERATIONS
/GENERATE PATTERN WORDS
/AND BLOCK NUMBERS

```
*1400
1400 1274 *WRITE1, TAD RBFLOC
1401 3010 DCA 10
1402 1453 TAD I LSTBLK
1403 3410 DCA I 10 /FIRST BLOCK WRITTEN
1404 1077 TAD NUMBLK
1405 7041 CMA IAC
1406 1453 TAD I LSTBLK /LST BLOCK+1
1407 3040 DCA RECORD
1410 1040 TAD RECORD
1411 1146 TAD K5075
1412 7700 SMA CLA
1413 5317 JMP REWCK
1414 1040 TAD RECORD

1415 3410 DCA I 10
1416 4462 JMS I RANGEN
1417 3410 DCA I 10 /FIRST RANDOM WORD
1420 4462 JMS I RANGEN
1421 3410 DCA I 10 /2ND RANDOM WORD
1422 4462 JMS I RANGEN
1423 0125 AND K0003
1424 7650 SNA CLA
1425 7040 CMA
1426 3076 DCA DIRFLG
1427 1076 TAD DIRFLG
1430 1073 TAD RBUFST
1431 3010 DCA 10

1432 1010 TAD 10
1433 3011 DCA 11 /MAKE FIRST WORD
1434 1121 TAD K4000 /INDICATE BACKWARD
1435 1410 TAD I 10 /OR 2ND WORD
1436 3411 DCA I 11 INDICATE FORWARD
1437 4461 JMS I GENPAT /GENERATE 129 WORD PATTERN
1440 1077 TAD NUMBLK
1441 3370 DCA SAVNUM
1442 1077 *SORT1R, TAD NUMBLK
1443 7040 CMA

1444 3040 DCA RECORD
1445 1076 TAD DIRFLG
1446 7640 SZA CLA /BACKWARD IS
1447 1040 TAD RECORD /LAST BLOCK+NUM BLOCKS
1450 1453 TAD I LSTBLK
1451 3040 DCA RECORD /TO FIND FIRST
/BLOCK TO BE WRITTEN
```

```

/TEST FOR WRITE DATA C MODE
/IF NUMBER OF BLOCKS=INC OF 3
1452 5653      JMP I .+1
1453 2200      WDCMOD
1454 4455      JMS I SRCHIT

1455 1150      WRT1A,      TAD K0050
1456 6764      IOT 764      /SRCH TO WRITE DATA
1457 1117      TAD K7577
1460 3443      DCA I WCLOC
1461 1074      TAD RBFLOC
1462 3444      DCA I CALOC
1463 4466      JMS I WAITI

1464 0001      1
1465 5326      JMP WSTERR
1466 1443      TAD I WCLOC
1467 7640      SZA CLA
1470 5326      JMP WSTERR
1471 1076      TAD DIRFLG      /0 OR -1
1472 7100      CLL
1473 1040      TAD RECORD
1474 7510      SPA
1475 7320      CLA STL
1476 3040      DCA RECORD
1477 7420      SNL      /RECORD-1
1500 2040      ISZ RECORD      /NO +1

1501 2077      ISZ NUMBLK      /DONE ALL BLOCKS
1502 5256      JMP WRT1A+1      /WRITE DATA 1 MORE
1503 1040      *DINC,      TAD RECORD      /NEW POSITION
1504 3451      DCA I POSITN
1505 1076      TAD DIRFLG

1506 3452      DCA I DIRECT      /DIRECTION FLAG
1507 1502      TAD I RBFWD2
1510 0143      AND K3777
1511 3453      DCA I LSTBLK
1512 1113      TAD K0200
1513 6764      IOT 764      /STOP TAPE
1514 5715      JMP I .+1
1515 0221      MOFPRO      /RANDOM SELECT AGAIN

1516 0525      REWDRV
1517 4462      REWCK,      JMS I RANGEN      /GET RANDOM NUMBER
1520 7710      SPA CLA      /+ OR -
1521 5716      JMP I REWCK-1
1522 3040      DCA RECORD      /+READ BLOCK 0
1523 3076      DCA DIRFLG      /FORWARD
1524 5725      JMP I .+1
1525 0304      RDTAB

```

1526	4467	WSTERR,	JMS I ERRSTP
1527	4501	JMS I	WRTDTY
1530	1071	TAD	RECRDK
1531	4506	JMS I	TYPCON
1532	4527	JMS I	TYPTX
1533	7242		42
1534	5457		5457
1535	4353		4353
1536	7700		7700
1537	1370	TAD	SAVNUM
1540	3077	DCA	NUMBLK
1541	6772	IOT	772
1542	3372	DCA	SAVNUM+2
1543	1371	TAD	SAVNUM+1
1544	4506	JMS I	TYPCON
1545	4507	JMS I	TYPTX
1546	0063		0063
1547	6441		6441
1550	6400		6400
1551	4277		4277
1552	1043	TAD	WCLOC
1553	4506	JMS I	TYPCON
1554	4507	JMS I	TYPTX
1555	0067		67
1556	1643		1643
1557	1677		1677
1560	7604	LAS	
1561	0154	AND	K6000
1562	7650	SNA	CLA
1563	7402	HLT	
1564	2151	ISZ	PASFLG
1565	5767	JMP I	+.2
1566	5242	JMP	CORT1R
1567	0525	REWDRV	
1570	0000	SAVNUM,	0
1571	1572	.+1	

/2ND ERROR
/REWIND DRIVE

/READ ENTIRE SEQUENCE OF
/BLOCKS AS ORIGINALLY WRITTEN

```

*1600
1600 1473 RDSEQ, TAD I RBUFST /GET LOWEST BLOCK
1601 0143 AND K3777
1602 3040 DCA RECORD
1603 1502 TAD I RBFWD2 /GET LST BLOCK +1
1604 0143 AND K3777
1605 3077 DCA NUMBLK /MAKE NUMBER OF BLOCKS
1606 1040 TAD RECORD
1607 7040 CMA
1610 1077 TAD NUMBLK
1611 3077 DCA NUMBLK
1612 4462 JMS I RANGEN
1613 7100 CLL
1614 7710 SPA CLA /READ FWD
1615 7040 CMA /NO BKWD
1616 7440 SZA
1617 7120 STL /L=1 IF BKWD
1620 3076 DCA DIRFLG
1621 1040 TAD RECORD
1622 7430 SZL
1623 1077 TAD NUMBLK
1624 3040 DCA RECORD
1625 1077 TAD NUMBLK /MAKE NUMBER OF BLKS-
1626 7040 CMA

1627 3077 DCA NUMBLK /MAKE-
1630 4462 JMS I READ1 /READ FIRST BLOCK
1631 5504 JMP I READ2 /STATUS ERROR ON READ
1632 2077

```

		RDBUF1,	ISZ NUMBLK	/READ ALL
1633	5236	JMP ,+3	/NO,	
1634	1113	TAD K0200		/IF ALL BLOCKS READ
1635	5242	JMP ,+5		/STOP TAPE
1636	1103	TAD BF2LOC		
1637	3444	DCA I CALOC		/ADDRS IS BUFR 2
1640	1117	TAD K7577		/129 WORDS
1641	3443	DCA I WCLOC		
1642	6764	IOT 764		/RESET ENABLES OR STOP TAPE
1643	4465	JMS I DATACO		/COMPARE
1644	6774	RANBFR /BLOCK		GENERATED
1645	7175	BUFFRS		/AGAINST BLOCK READ
1646	0201	0201		
1647	2072	ISZ COCNTR		/ANY DATA ERRORS
1650	5505	JMP I READ2A		/YES, READ OTHER DIRECTION
1651	1077	TAD NUMBLK		
1652	7650	SNA CLA		/DONE COMPLETE SERIES
1653	5325	JMP NDOFRD		/YES
1654	1076	TAD DIRFLG		
1655	7450	SNA		
1656	7001	IAC		
1657	1040	TAD RECORD		/LAST BLOCK +OR-1
1660	3040	DCA RECORD		
1661	4466	JMS I WAITI		
1662	0001	1		
1663	5504	JMP I READ2		/STATUS ERROR, REREAD
1664	1443	TAD I WCLOC		
1665	7640	SZA CLA		
1666	5504	JMP I READ2		
1667	2077	ISZ NUMBLK		/DONE ALL READS
1670	5273	JMP ,+3		
1671	1113	TAD K0200		/DONE ALL
1672	5277	JMP ,+5		/STOP TAPE
1673	1075	TAD BF1LOC		
1674	3444	DCA I CALOC		/ADDRS IS FIRST BUFR
1675	1117	TAD K7577		/129 WORDS
1676	3443	DCA I WCLOC		
1677	6764	IOT 764		/RESET ENABLES OR STOP TAPE
1700	4465	JMS I DATACO		/COMPARE
1701	6774	RANBFR		/GENERATED
1702	7377	BUFFRS+202		/AGAINST READ
1703	0201	0201		
1704	2072	ISZ COCNTR		/ANY COMPARE ERRORS
1705	5505	JMP I READ2A		/YES, READ OTHER DIRECTION
1706	1077	TAD NUMBLK		
1707	7650	SNA CLA		/READ AND COMPARED ALL
1710	5325	JMP NDOFRD		/YES
1711	1076	TAD DIRFLG		

```

1712 7450 SNA /LAST BLOCK + DR-
1713 7001 IAC /1
1714 1040 TAD RECORD
1715 3040 DCA RECORD
1716 4466 JMS I WAITI /WAIT TO FINISH READ
1717 0001 1
1720 5504 JMP I READ2 /STATUS ERROR, REREAD
1721 1443 TAD I WCLOC
1722 7640 SZA CLA
1723 5504 JMP I READ2
1724 5232 JMP RDBUF1 /COMPARE FIRST BUFFER

```

```

1725 1040 VDOFRD, TAD RECORD
1726 3451 DCA I POSITN /NEW POSITION
1727 1076 TAD DIRFLG
1730 3452 DCA I DIRECT /INDICATE DIRECTION
1731 5732 JMP I .+1
1732 0221 MOFPRO

```

PAUSE

```

/TC01 DECTREX 1 - TAPE 2
/READ DATA CONTINUOUS MODE
/TWO BLOCKS AND COMPARE DATA READ

```

*2000

```

2000 1076 RDCMOD, TAD DIRFLG
2001 7640 SZA CLA /FORWARDS
2002 5212 JMP ,+10 /NO
2003 1040 TAD RECORD
2004 7040 CMA
2005 1453 TAD I LSTBLK /BLOCK+1 WRITTEN
2006 7440 SZA
2007 5215 JMP ,+6
2010 5611 JMP I .+1 /BLOCK NOT WRITTEN
2011 0304 RDTAB /READ ONLY 1
2012 1040 TAD RECORD
2013 7650 SNA CLA /BLOCK 0 BACKWARDS
2014 5611 JMP I .-3 /READ ONLY 1

2015 4455 JMS I SRCHIT /FIND FIRST BLOCK
2016 1313 TAD K0130 /READ DATA
2017 6764 IOT 764 /CONTINUOUS MODE
2020 1117 TAD K7577
2021 1117 TAD K7577
2022 3443 DCA I WCLOC
2023 1075 TAD BF1LOC
2024 3444 DCA I CALOC
2025 4466 JMS I WAITI
2026 0001 1
2027 5275 JMP RDCERR /SEE WHICH BLOCK IN ERROR
2030 1443 TAD I WCLOC
2031 7640 SZA CLA
2032 5504 JMP I READ2
2033 1113 TAD K0200
2034 6764 IOT 764 /STOP TAPE

```

2040	1040	TAD RECORD	
2041	3040	DCA RECORD	
2042	4464	JMS I REGENP	/VERIFY LAST BLOCK
2043	4465	JMS I DATAO	/FIRST
2044	6774	RANBFR	
2045	7376	BUFFRS+201	
2046	0201	0201	
2047	2072	ISZ COCNTR	/ANY DATA ERRORS
2050	5505	JMP I READ2A	/YES REREAD
2051	7040		

2052	1103	CMA	
2053	3444	TAD BF2LOC	
		DCA I CALOC	
2054	4464	JMS I REGENP	
2055	1076	TAD DIRFLG	
2056	7450	SNA	
2057	7001	IAC	
2060	7041	CMA IAC	
2061	1040	TAD RECORD	
2062	3040	DCA RECORD	/VERIFY FIRST
2063	4465	JMS I DATAO	/BLOCK READ
2064	6774	RANBFR	
2065	7175	BUFFRS	
2066	0201	0201	
2067	2072	ISZ COCNTR	/ANY DATA ERRORS
2070	5505	JMP I READ2A	/YES, REREAD
2071	1040	TAD RECORD	
2072	3451	DCA I POSITN	
2073	5674	JMP I .+1	
2074	0221	MOFPRO	
2075	1443	RDCERR, TAD I WCLOC	
2076	7450	SNA	/ERROR IN 2ND BLOCK
2077	5305	JMP .+6	/YES COUNT BLOCK
2100	7041	CMA IAC	
2101	1117	TAD K7577	/WC-129
2102	3443	DCA I WCLOC	
2103	5704	JMP I .+1	/TYPE STATUS ERROR
2104	0420	REREAD	
2105	1076	TAD DIRFLG	
2106	7450	SNA	/BACKWARDS-1
2107	7001	IAC	/FORWARD+1
2110	1040	TAD RECORD	
2111	3040	DCA RECORD	
2112	5704	JMP I .-6	/TYPE STATUS ERROR
2113	0130	<0130,	130

/WRITE DATA CONTINUOUS MODE
/IF NUMBER OF BLOCKS IS AN INCREMENT OF 3

*2200

2200	3277	WDCMOD,	DCA TRECTR	
2201	1077	TAD	NUMBLK	
2202	1125	TAD	K0203	
2203	7450	SNA		/AN INC OF 3
2204	5212	JMP	SWCMOD	/YES WRITE C MODE
2205	7500	SMA		/GONE PAST 0
2206	5611	JMP	I .+3	/NOT INC OF 3
2207	2277	ISZ	TRECTR	
2210	5202	JMP	WDCMOD+2	
2211	1454	WRT1A-1		
2212	1277	SWCMOD,	TAD TRECTR	/MAKE NUMBER
2213	7040	CMA		/OF GROUPS OF 3
2214	3277	DCA	TRECTR	/2'S COMPLIMENT
2215	4470	JMS	I DATAMV	/MAKE PATTERN
2216	6774	RANBFR		/3 BUFFERS LONG
2217	7175	BUFFRS		
2220	7376	7376		
2221	4455	JMS	I SRCHIT	/FIND FIRST BLOCK
2222	1301	TAD	K0150	/WRITE DATA CONTINUOUSLY
2223	6764	SWCMDL,	IOT 764	
2224	1300	TAD	K7175	/-129 3 TIMES
2225	3443	DCA	I WCLOC	
2226	1074	TAD	RBFLOC	
2227	3444	DCA	I CALOC	
2230	4466	JMS	I WAITI	
2231	0001	1		
2232	5253	JMP	WDCERR	/NOT NORMAL INTERRUPT
2233	1443	TAD	I WCLOC	
2234	7640	SZA	CLA	
2235	5253	JMP	WDCERR	/WC NOT ZERO
2236	1076	TAD	DIRFLG	
2237	7100	CLL		
2240	7640	SZA	CLA	
2241	7120	STL		
2242	1125	TAD	K0003	
2243	7430	SZL		/BACKWARDS IS -3
2244	7041	CMA	IAC	
2245	1040	TAD	RECORD	
2246	3040	DCA	RECORD	
2247	2277	ISZ	TRECTR	/DONE ALL
2250	5223	JMP	SWCMDL	/NO, DO NEXT 3 BLOCKS
2251	5652	JMP	I .+1	
2252	1503	WDINC		

2253	3277	*DCERR,	DCA TRECTR	/FIND WHICH
2254	1443	TAD I WCLOC		/BLOCK WAS IN
2255	7041	CMA IAC		/ERROR
2256	1117	TAD K7577		
2257	7510	SPA		
2260	5253	JMP ,+3		
2261	2277	ISZ TRECTR		
2262	5256	JMP ,-4		
2263	7200	CLA		
2264	1076	TAD DIRFLG		
2265	7100	CLL		
2266	7640	SZA CLA		/FORWARD
2267	7120	STL		/BACKWARD MAKE-1 OR -2
2270	1277	TAD TRECTR		
2271	7430	SZL		
2272	7041	CMA IAC		
2273	1040	TAD RECORD		
2274	3040	DCA RECORD		/BLOCK IN ERRJR
2275	5676	JMP I ,+1		/TYPEOUT WRITE ERROR
2276	1526	WSTERR		
2277	0000	TRECTR,	0	
2300	7175	<7175,	7175	
2301	0150	<0150,	150	

/READ ALL ROUTINE
/REV CKSUM DATA AND CKSUM ARE READ
/CKSUMS ARE GENERATED AND TESTED

*2400

2400	1076	RALLTS,	TAD DIRFLG	
2401	7450	SNA	/BACKWARDS	
2402	7001	IAC	/FORWARDS	
2403	7041	CMA IAC	/+1 FOR BACK-1 FOR FWD	
2404	1040	TAD RECORD		
2405	3040	DCA RECORD		
2406	1040	TAD RECORD		
2407	7510	SPA		
2410	5214	JMP ,+4		
2411	1147	TAD K5076		
2412	7710	SPA CLA		
2413	5217	JMP ,+4		
2414	7200	CLA		
2415	5616	JMP I ,+1	/CAVT DO 0 OR 2701	
2416	0235	MOFPR1		

/FIND THE BLOCK AND CHANGE TO READ ALL
/CLEAR READ ALL ERR FLG

2417	3377	DCA RAEFLG	/CLEAR READ ALL ERR. FLG	
2420	4455	JMS I SRCHIT		
2421	6764	IOT 764		
2422	1076	TAD DIRFLG		
2423	7450	SNA		
2424	7001	IAC		
2425	1040	TAD RECORD		
2426	3040	DCA RECORD		
2427	6773	IOT 773		
2430	5227	JMP , -1		
2431	7710	SPA CLA		
2432	5371	JMP RASERR		
2433	1041	TAD BLKFND		
2434	7041	CMA IAC		
2435	1040	TAD RECORD		
2436	7640	SZA CLA		
2437	5371	JMP RASERR		

```
2440 1374 /CHANGE TO READ ALL CONTINUOUS  
      TAD K0120  
2441 6764      IOT 764  
2442 1375      TAD K7571  
2443 3443      DCA I WCLOC  
2444 1075      TAD BF1LOC  
2445 3444      DCA I CALOC  
2446 4466      JMS I WAITI  
2447 0001      1  
2450 5770      JMP I RARERR /READ ALL STATUS ERROR  
2451 1443      TAD I WCLOC  
2452 7640      SZA CLA  
2453 5770      JMP I RARERR  
2454 1113      TAD K0200  
2455 6764      IOT 764 /STOP TAPE
```

/GENERATE CKSUMS AND TEST SUM=00

```
2456 1376 TAD RADLOC
2457 3010 DCA 10
2460 1117 TAD K7577
2461 3011 DCA 11
2462 3364 DCA CKSUMR /CLR ACCUMULATED CKSUM
2463 1410 TAD I 10
2464 0134 AND K0077
2465 3363 DCA REVCHK /SAVE REVERSE CKSUM
2466 1410 TAD I 10
2467 4331 JMS XORSUM /GENERATE DATA SUM
2470 2011 ISZ 11
2471 5266 JMP , -3
2472 1364 TAD CKSUMR
2473 7040 CMA
2474 0134 AND K0077
2475 3361 DCA DATASM /SAVE DATA SUM
2476 1410 TAD I 10
2477 0127 AND K7700
2500 3362 DCA FWDCHK /SAVE FORWARD CHECKSUM
2501 1361 TAD DATASM
2502 3364 DCA CKSUMR
2503 1362 TAD FWDCHK
2504 4331 JMS XORSUM /DATA SUM + FORWARD CKECK
2505 1363 TAD REVCHK
2506 4331 JMS XORSUM /MAKE FINAL SJM
2507 1364 TAD CKSUMR
2510 7640 SZA CLA /SUM OK
2511 5760 JMP I CKSERR /NO, ERROR
2512 1040 RADCHK, TAD RECORD
2513 3451 DCA I POSITN
2514 7040 CMA
2515 1444 TAD I CALOC
2516 3444 DCA I CALOC
2517 4464 JMS I REGENP
2520 4465 JMS I DATACO
2521 6774 RANBFR
2522 7202 BUFFRS+5
2523 0201 0201
2524 2377 ISZ RAEFLG
2525 2072 ISZ COCNTR
2526 5505 JMP I READ2A
2527 5730 JMP I .+1
2530 0217 MOFPRO-2
```

```

/FORM 6 BIT XOR OF AC AND CKSUMR
/SAVE RESULTS IN CKSUMR
2531 5331 XORSUM, JMP
2532 7040 CMA
2533 3365 DCA XORS AV
2534 7240 CMA
2535 3366 DCA XORS AV+1 /TO COUNT PASSES
2536 1365 TAD XORS AV
2537 7040 CMA
2540 0364 AND CKSUMR
2541 3367 DCA XORS AV+2 /MAKE PARTIAL SUM
2542 1364 TAD CKSUMR
2543 7040 CMA
2544 0365 AND XORS AV /OTHER HALF SJM
2545 1367 TAD XORS AV+2 /COMBINE
2546 0134 AND K0077 /CLR TO LWR 6
2547 3364 DCA CKSUMR
2550 2366 ISZ XORS AV+1 /DONE UPPER 6?
2551 5731 JMP I XORSUM /DONE ALL EXIT
2552 1365 TAD XORS AV
2553 7012 RTR /MOVE OVER 6
2554 7012 RTR
2555 7012 RTR
2556 3365 DCA XORS AV
2557 5336 JMP XORSUM+5 /XOR UPPER

2560 2600 CKSERR, ERRCKS
2561 0000 JATASM, 0
2562 0000 FWDCHK, 0
2563 0000 REVCHK, 0
2564 0000 CKSUMR, 0
2565 0000 XORS AV, 0
2566 0000 0
2567 0000 0
2570 2707 RARERR, ERRRAR
2571 3151 RASERR, DCA PASFLG
2572 5773 JMP I .+1
2573 4000 SRHERR
2574 0120 <0120, 0120
2575 7571 <7571, 7571
2576 7200 RADLOC, BUFFRS+3
2577 0000 RAEFLG, 0

```

/CHECK SUM ERROR TYPEOUT

*2600

2600	4731	ERRCKS,	JMS I CSECON	/TYRALL
2601	1071	TAD	RECRDK	
2602	4506	JMS I	TYPCON	
2603	4507	JMS I	TYPTX	
2604	0042		42	
2605	5457		5457	
2606	4353		4353	
2607	7777		7777	
2610	4353		4353	
2611	6365		6365	
2612	5500		5500	
2613	4562		4562	
2614	6257		6257	
2615	6277		6277	
2616	1302	TAD	CSECON+1	/REVCHK
2617	4506	JMS I	TYPCON	
2620	4507	JMS I	TYPTX	
2621	0062		62	
2622	4566		4566	
2623	7700		7700	
2624	4257	JMS	CKSTYP	/TYPE CHK SJM
2625	1303	TAD	CSECON+2	/DATASM
2626	4506	JMS I	TYPCON	
2627	4704	JMS I	CSECON+3	/TYDATA
2630	4257	JMS	CKSTYP	/CHECKSUM
2631	4270	JMS	CALCTY	
2632	1305	TAD	CSECON+4	/FWD CHECK
2633	4506	JMS I	TYPCON	
2634	4257	JMS	CKSTYP	/CHK SUM
2635	1706	TAD I	CSECON+5	
2636	7040		CMA	
2637	0134	AND	K0077	
2640	3706	DCA I	CSECON+5	
2641	1306	TAD	CSECON+5	/CKSUMR
2642	4506	JMS I	TYPCON	
2643	4270	JMS	CALCTY	
2644	4507	JMS I	TYPTX	
2645	0054		54	/L
2646	1660		1660	/,P
2647	1642		1642	/,B
2650	1677		1677	/,
2651	7040		CMA	
2652	3655	DCA I	.*+3	
2653	5654	JMP I	.*+1	
2654	2512		RADCHK	
2655	2577		RAEFLG	
2656	0525		REWDRV	

2657	5257	CKSTYP,	JMP ,
2660	4507	JMS I	TYPTX
2661	0043	43	
2662	5045	5045	
2663	4353	4353	
2664	0063	63	
2665	6555	6555	
2666	7700	7700	
2667	5657	JMP I	CKSTYP
2670	5270	CALCTY,	JMP ,
2671	4507	JMS I	TYPTX
2672	0043	43	
2673	4154	4154	
2674	4365	4365	
2675	5441	5441	
2676	6445	6445	
2677	4477	4477	
2700	5670	JMP I	CALCTY
2701	4432	CSECON,	TYRALL
2702	2563	REVCHK	
2703	2561	DATASM	
2704	4474	TYDATA	
2705	2562	FWDCHK	
2706	2564	CKSUMR	

/READ ALL STATUS ERROR TYPEOUT

2707	4467	ERRRAR, JMS I ERRSTP
2710	4701	JMS I CSECON
2711	1071	TAD RECROK
2712	4506	JMS I TYPCON
2713	4507	JMS I TYPTX
2714	0042	42
2715	5457	5457
2716	4353	4353
2717	7700	7700
2720	6772	LOT 772
2721	3345	DCA ,+24
2722	1344	TAD ,+22
2723	4506	JMS I TYPCON
2724	4507	JMS I TYPTX
2725	0063	63
2726	6441	6441
2727	6400	6400
2730	4277	4277
2731	1043	TAD WCLOC
2732	4506	JMS I TYPCON
2733	4507	JMS I TYPTX
2734	0067	67
2735	1643	1643
2736	1677	1677
2737	7004	LAS
2740	0154	AND K6000
2741	7650	SNA CLA
2742	7402	HLT
2743	5656	JMP I CKSTYP=1
2744	2745	,+1
2745	0000	0

/TC01 RANDOM EXERCISER
/TYPEOUTS AND SOME CONTROL ROUTINES

*3000

/WAIT FOR INTERRUPT ROUTINE
/SYNCHRONIZE PROCESSOR TESTS

3000	5200	*ATINT,	JMP ,	
3001	1323	TAD	WTKON1	/GET 5 SECOND WAIT K
3002	3325	DCA	WTCNTR	
3003	6761	IOT	761	
3004	0123	AND	K0070	
3005	7640	SZA	CLA	/MOVE FUNCTION IN STAT
3006	5211	JMP	,+3	/NO 5 SECONDS ENOUGH
3007	1324	TAD	WTKON2	/CHANGE WAIT COUNT TO
3010	3325	DCA	WTCNTR	/45 SECOND FOR MOVE
3011	2162	ISZ	FRSWAT	/FIRST PROGRAM WAIT
3012	5217	JMP	RESETD	/NO RESTORE AC AND LINK
3013	4331	JMS	SETIDX	
3014	6001	ION		
3015	5616	JMP	I ,+1	/START IS TEST
3016	3200	ISZTST		
3017	3162	RESETD,	DCA FRSWAT	/INDICATE NOT FIRST WAIT
3020	2342	ISZ	PICNTR	
3021	7410	SKP		
3022	4331	JMS	SETIDX	
3023	1000	TAD	0	
3024	3743	DCA	I P1DEX1	
3025	2343	ISZ	P1DEX1	
3026	1326	TAD	LKSAVE	
3027	7010	RAR		/RESTORE LINK
3030	7200	CLA		
3031	1327	TAD	ACSAVE	/RESTORE ACCUMULATOR
3032	6001	ION		
3033	5400	JMP	I 0	/RETURN FROM LAST INTERRUPT

/AFTER INTERRUPT RETURNS TO HERE

3034	3327	IRECD,	DCA ACSAVE	/SAVE ACCUMULATOR
3035	7004	RAL		
3036	3326	DCA LKSAVE		/AND LINK FOR NEXT WAIT
3037	6771	IOT 771		
3040	5255	JMP NDTSKP		/NO DECTAPE FLAG SET
3041	6772	IOT 772		
3042	3330	DCA SBRECD		/SAVE STATUS B
3043	1600	TAD I WATINT		
3044	7040	CMA		/GET STAT B COMPARE CON
3045	0330	AND SBRECD		
3046	7650	SNA CLA		/ANY ILLEGAL FLAGS
3047	2200	ISZ WATINT		/NO IS SKIP EXIT TWICE
3050	2200	ISZ WATINT		/ONLY 1 ISZ IF NOT I EXPECTED
3051	1200	TAD WATINT		
3052	3744	DCA I PIDEX2		
3053	2344	ISZ PIDEX2		
3054	5600	JMP I WATINT		
3055	4467	NDTSKP,	JMS I ERRSTP	/STOP TAPE
3056	4507	JMS I TYPTX		
3057	7777	7777		
3060	6016	6016		
3061	5116	5116		
3062	0077	0077		
3063	5304	JMP PTSTNS		
3064	3065	PTSTND,	.+1	
3065	2325	ISZ WTCNTR		/WAITED LONG ENOUGH
3066	5664	JMP I PTSTND		/NO DO NEXT TEST
3067	6002	IOF		
3070	6771	IOT 771		/ANY FLAGS SET
3071	7040	CMA		/NO
3072	3330	DCA SBRECD		
3073	4467	JMS I ERRSTP		/STOP TAPE
3074	4507	JMS I TYPTX		
3075	7777	7777		/NO PI
3076	5657	5657		
3077	0060	60		
3100	1651	1651		
3101	1677	1677		
3102	2330	ISZ SBRECD		
3103	5307	JMP .+4		/DECTAPE DID SKIP

3104	4507	PTSTNS,	JMS I TYPTX	
3105	0056	56		
3106	5777	5777		
3107	4507	JMS I TYPTX		
3110	0044	44		
3111	4543	4543		
3112	6441	6441		
3113	6045	6045		
3114	0003	03		
3115	5351	5351		
3116	6077	6077		
3117	2200	ISZ WATINT		
3120	7040	CMA		
3121	3102	DCA FRSWAT		
3122	5600	JMP I WATINT		
3123	7655	WTKON1,	7655	/WAIT COUNT 5 SECONDS
3124	6422	WTKON2,	6422	/WAIT COUNT 45 SECONDS
3125	0000	WTCNTR,	0	
3126	0000	-KSAVE,	0	
3127	0000	ACSAVE,	0	
3130	0000	SBRECD,	0	
3131	5331	SETIDX,	JMP .	
3132	1341	TAD K7772Y		
3133	3342	DCA PICNTR		
3134	1345	TAD PITBL		
3135	3343	DCA PINDEX1		
3136	1354	TAD PIRTL		
3137	3344	DCA PINDEX2		
3140	5731	JMP I SETIDX		
3141	7772	<7772Y,	7772	
3142	0000	PICNTR,	0	
3143	0000	PINDEX1,	0	
3144	0000	PINDEX2,	0	
3145	3146	PITBL,	.*1	
		*PITBL+7		
3154	3155	PIRTL,	.*1	

/PROCESS OR TEST FOR DECTREX1
/TESTS ARE RUN WHILE WAITING FOR INT

*3200

/ISZ TEST ABOUT 61 MILLISECONDS

3200	7040	ISZTST, CMA
3201	3345	DCA TEMP5
3202	3342	DCA TEMP2
3203	3341	DCA TEMP1
3204	2341	ISZ TEMP1
3205	2342	ISZ TEMP2
3206	5204	JMP , -2
3207	1342	TAD TEMP2
3210	7440	SZA
3211	7402	HLT
3212	7240	CLA CMA
3213	1341	TAD TEMP1
3214	7440	SZA
3215	7402	HLT
3216	2345	ISZ TEMP5
3217	7410	SKP
3220	5203	JMP ISZTST+3
3221	4750	JMS I NDPTST

/ROTATE 1 TEST ABOUT 67 MILLISECONDS

3222	1342	ROT1TS, TAD TEMP2
3223	7130	STL RAR
3224	7004	RAL
3225	7420	SNL
3226	7402	HLT
3227	7041	CMA IAC
3230	1342	TAD TEMP2
3231	7440	SZA
3232	7402	HLT
3233	2342	ISZ TEMP2
3234	5222	JMP ROT1TS
3235	4750	JMS I NDPTST

/ROTATE 2 TEST ALSO ABOUT 67 MILLISECONDS

3236	1342	ROT2TS,	TAD TEMP2
3237	7106	CLL	RIL
3240	7012	RTR	
3241	7430	SZL	
3242	7402	HLT	
3243	7041	CMA	IAC
3244	1342	TAD	TEMP2
3245	7440	SZA	
3246	7402	HLT	
3247	2342	ISZ	TEMP2
3250	5236	JMP	ROT2TS
3251	4750	JMS	I NDPTST

/TAD TEST ADD EVERY COM TO RAN VO
/ABOUT 86 MILLISECONDS

3252	3343	TADTST,	DCA TEMP3
3253	1346	TAD	PRAN1
3254	7104	CLL	RAL
3255	7430	SZL	
3256	7001	IAC	
3257	3346	DCA	PRAN1
3260	1347	TAD	PRAN2
3261	1346	TAD	PRAN1
3262	3347	DCA	PRAN2
3263	1347	TAD	PRAN2
3264	3344	DCA	TEMP4
3265	1347	TAD	PRAN2
3266	1343	TAD	TEMP3
3267	7041	CMA	IAC
3270	1344	TAD	TEMP4
3271	7440	SZA	
3272	7402	HLT	
3273	2344	ISZ	TEMP4
3274	7000	NOP	
3275	2343	ISZ	TEMP3
3276	5265	JMP	.-11
3277	4750	JMS	I NDPTST

/JMS TEST MAKE 13 PASSES OF 128 CONSECUTIVE JMS ,
 /AND COMPARE RESULTS FOR ABOUT 63 MILLISECONDS

3300	1354	JMSTST,	TAD K7763X	
3301	3341	DCA	TEMP1	
3302	1353	TAD	K7600X	
3303	3342	DCA	TEMP2	
3304	1351	TAD	JMSLOC	
3305	3343	DCA	TEMP3	
3306	1352	TAD	JMSKON	
3307	3344	DCA	TEMP4	
3310	1344	TAD	TEMP4	/STORE 128 JMS .
3311	3743	DCA I	TEMP3	/STARTING AT ADDRESS
3312	2344	ISZ	TEMP4	/6000
3313	2343	ISZ	TEMP3	
3314	2342	ISZ	TEMP2	
3315	5310	JMP	,-5	
3316	1355	TAD	RETJMP	/STORE JMP I RETUJM
3317	3743	DCA I	TEMP3	/TO RETURN FROM JMS
3320	5751	JMP I	JMSLOC	/EXECUTE 128 JMS
3321	1353	JMRETU,	TAD K7600X	/RETURN FROM EXECUTE
3322	3342	DCA	TEMP2	
3323	1351	TAD	JMSLOC	
3324	3343	DCA	TEMP3	/COMPARE ADDRESSES
3325	1343	TAD	TEMP3	/FOR .+1
3326	7040	CMA		
3327	1743	TAD I	TEMP3	
3330	7440	SZA		
3331	7402	HLT		
3332	2343	ISZ	TEMP3	/INC COMP AND FETCH
3333	2342	ISZ	TEMP2	/DONE 128 YET
3334	5325	JMP	JMRETU+4	
3335	2341	ISZ	TEMP1	
3336	5302	JMP	JMSTST+2	
3337	4750	JMS I	NDPTST	
3340	5200	JMP	ISZTST	
3341	0000	TEMP1,	0	
3342	0000	TEMP2,	0	
3343	0000	TEMP3,	0	
3344	0000	TEMP4,	0	
3345	0000	TEMP5,	0	
3346	4263	PRAN1,	4263	
3347	2634	PRAN2,	2634	
3350	3064	NDPTST,	PTSTND	
3351	6000	JMSLOC,	6000	
3352	4200	JMSKON,	4200	/JMS + 200
3353	7600	K7600X,	7600	/FOR COUNTING JMS
3354	7763	K7763X,	7763	
3355	5561	RETJMP,	JMP I RETUJM	/(RETJJM) ARE JMRETU

PAUSE

/TC01 DECTEX 1 - TAPE 3
 /REWIND ALL DRIVES SELECTED
 /TO END ZONE AT START OF TAPE

*3400

```

3400 5200 REPOSI,      JMP ,
3401 4217     JMS RSFDRV      /RESET POINTRS TO FIRST DRIVE
3402 1140     TAD K0604        /MOVE BACKWARDS
3403 1046     TAD UNFUNC      /*POSITIONED UNIT NO
3404 6766     IOT 766        /SET STATUS A
3405 4466     JMS I WAITI
3406 5000     5000          /INDICATE EXPECT END
3407 5340     JMP MOVER      /NO! STATUS EXPECTED

3410 7240     CLA CMA
3411 3451     DCA I POSITN    /INDICATE END ZONE
3412 7240     CLA CMA
3413 3452     DCA I DIRECT    /INDICATE BACKWARDS
3414 4237     JMS CHNGDR     /SET UP NEXT DRIVE
3415 5202     JMP REPOSI+2    /REWIND NEXT DRIVE
3416 5600     JMP I REPOSI    /GOT ALL DRIVES, EXIT

/RESET CURRENT DRIVE POINTERS TO
/FIRST DRIVE SELECTED
3417 5217     RSFDRV,      JMP ,
3420 7200     CLA
3421 3045     DCA CDRIVE     /SET INITIALLY TO 0
3422 1121     TAD K4000
3423 3050     DCA COMBIT
3424 1047     TAD MSBITS

3425 0050     AND COMBIT
3426 7640     SZA CLA        /THIS DRIVE SELECTED
3427 5235     JMP RSFDR1     /YES, SET POINTER
3430 1050     TAD COMBIT
3431 7110     CLL RAR
3432 3050     DCA COMBIT     /MOVE COMPARE BIT
3433 2045     ISZ CDRIVE     /INCREMENT DRIVE NUM.
3434 5224     JMP RSFDRV+5

/HAVE FOUND FIRST DRIVE SELECTED
3435 4261     RSFDR1,      JMS GNPTRS    /GENERATE CONTROL POINTERS
3436 5617     JMP I RSFDRV    /EXIT

```

```

/SELECT NEXT DRIVE OR
/RESET TO FIRST DRIVE AND SKIP
3437 5237 CHNGDR, JMP .
3440 7200 CLA
3441 1050 TAD COMBIT /GET DRIVE COMPARE BIT
3442 7110 CLL RAR /MOVE IT TO NEXT
3443 0300 AND NBIT8
3444 7440 SZA /LAST DRIVE NJM 7
3445 5251 JMP ,+4 /NO
3446 4217 JMS RSFDRV /RESET TO FIRST

3447 2237 ISZ CHNGDR /INCR. EXIT, SKIP
3450 5637 JMP I CHNGDR /EXIT
3451 3050 DCA COMBIT
3452 2045 ISZ CDRIVE
3453 1050 TAD COMBIT
3454 0047 AND MSBITS
3455 7650 SNA CLA /THIS DRIVE SELECTED
3456 5240 JMP CHNGDR+1 /NO

3457 4261 JMS GNPTRS /GENERATE DRIVE POINTERS
3460 5637 JMP I CHNGDR

/GENERATE LAST RECMD,
/DIRECTION AND UNIT NUMBER POINTERS
/FOR DECTAPE FUNCTIONS

3461 5261 GNPTRS, JMP .
3462 1045 TAD CDRIVE /DRIVE NUMBER
3463 7112 CLL RTR
3464 7012 RTR /POSITION TO BITS 0,1,2
3465 3046 DCA UNFUNC
3466 1045 TAD CDRIVE /DRIVE NUMBER
3467 1301 TAD PNTRS /+ POS. PNTR ADDRS,
3470 3051 DCA POSITN /FOR INDIRECTS

3471 1045 TAD CDRIVE
3472 1312 TAD PNTRS+11 /+ DIRECTION PNTR
3473 3052 DCA DIRECT /FOR INDIRECTS
3474 1045 TAD CDRIVE
3475 1323 TAD PNTRS+22
3476 3053 DCA LSTBLK
3477 5661 JMP I GNPTRS
3500 7767 NBIT8, 7767

```

3501	3502	PNTRS,	.+1	/TO GET LAST RECORD NUMBER
3502	0000	0		/FOR DRIVE 8
3503	0000	0		/1
3504	0000	0		/2
3505	0000	0		/3
3506	0000	0		/4
3507	0000	0		/5
3510	0000	0		/6
3511	0000	0		/7
3512	3513	.+1		/TO GET LAST DIRECTION
3513	0000	0		/DIRECTION - JNIT 8
3514	0000	0		/1
3515	0000	0		/2
3516	0000	0		/3
3517	0000	0		/4
3520	0000	0		/5
3521	0000	0		/6
3522	0000	0		/7
3523	3524	.+1		
3524	0000	0		/TO GET LAST BLOCK WRITTEN
3525	0000	0		/DRIVE 8
3526	0000	0		/1
3527	0000	0		/2
3530	0000	0		/3
3531	0000	0		/4
3532	0000	0		/5
3533	0000	0		/6
3534	0000	0		/7
3535	3536	.+1		
3536	0000	0		
3537	4400	TYMOVE		
3540	4342	MOVER,	JMS ,+2	
3541	5202	JMP REPOSI+2		
3542	5342	JMP ,		
3543	4467	JMS I ERRSTP		
3544	4737	JMS I MOVER-1	/TYMOVE	
3545	6772	IOT 772		
3546	3336	DCA MOVER-2		
3547	1335	TAD MOVER-3		
3550	4506	JMS I TYPCON		
3551	4507	JMS I TYPTX		
3552	0063	63		
3553	6441	6441		
3554	6400	6400		
3555	4277	4277		
3556	7604	LAS		
3557	0154	AND K6000		
3560	7650	SNA CLA		
3561	7402	HLT		
3562	5742	JMP I MOVER+2		

/SEARCH ROUTINE FIND
/BLOCK IN (RECORD) IN DIRECTION
/INDICATED BY DIRFLG==0 FWD
/=7777 BKWD

*3600

3600	5200	SEARCH,	JMP .	
3601	7300	CLL	CLA	
3602	1076	TAD	DIRFLG	
3603	7640	SZA	CLA	/FORWARD
3604	7120	STL		/NO, BACKWARD
3605	1125	TAD	K0003	/BACKWARD TA
3606	7420	SNL		/IS BLOCK+3
3607	7041	CMA	IAC	/FWD IS BLOCK-3
3610	1040	TAD	RECORD	
3611	7041	CMA	IAC	
3612	3156	DCA	TAPONT	/INDICATE TA POINT
3613	1076	TAD	DIRFLG	
3614	7650	SNA	CLA	/FORWARD IS
3615	1137	TAD	K0400	/START BKWD
3616	1126	TAD	K0214	
3617	1046	TAD	UNFUNC	
3620	6766	IOT	766	
3621	1042	TAD	IDCON	
3622	3444	DCA	I CALOC	
3623	7040	CMA		
3624	3157	DCA	BLKFLG	
3625	4466	JMS	I WAITI	
3626	0001		1	
3627	5326	JMP	SREZTS	/SEE IF EZ FOUND
3630	4263	JMS	SRCNT	
3631	5235	JMP	SRTARN	/BLOCK=BLOCKFND
3632	5235	JMP	SRTARN	/GONE PAST BLK
3633	6764	IOT	764	/HAVEN'T REACHED BLOCK
3634	5225	JMP	,-7	/FOR TURN AROUND YET

```

3635 6761 SRTARN, IOT 761
3636 7240 CMA
3637 0113 AND K0200 /IN CASE MOTION=0
3640 1137 TAD K0400
3641 6764 IOT 764 /DO TURN AROUND
3642 1040 TAD RECORD
3643 7041 CMA IAC
3644 3156 DCA TAPONT
3645 7040 CMA
3646 3157 DCA BLKFLG
3647 4466 JMS I WAITI
3650 0021 1
3651 5555 JMP I SRCHER
3652 4263 JMS SRCONT /CHK RELATION OF BLOCK
3653 5620 JMP I SEARCH /FOUND BLOCK, EXIT
3654 5555 JMP I SRCHER /GONE PAST, ERROR
3655 7040 CMA
3656 1137 TAD BLKFLG
3657 7650 SNA CLA /READ 2 BLOCK NUMS
3660 5355 JMP SRCMOD /YES CHNG TO C MODE
3661 6764 IOT 764 /HAVEN'T REACHED YET
3662 5247 JMP , -13 /WAIT FOR NEXT

```

```

/FIND RELATION OF BLOCK FOUND
/TO BLOCK SOUGHT AND TEST
/BLOCK NUMBERS TO BE CONSECUTIVE
SRCONT, JMP .

```

```

3663 5263
3664 2157 ISZ BLKFLG /FIRST BLOCK NUMBER
3665 7410 SKP /NOT FIRST
3666 5302 JMP SRFBLK /FIRST, JUST SAVE IT
3667 6761 IOT 761
3670 0137 AND K0400
3671 7640 SZA CLA /BKWRDS IS
3672 7040 CMA /-1

3673 7450 SNA /FWD IS
3674 7001 IAC /+1
3675 1160 TAD PREBLK /LAST BLOCK +JR-1
3676 7041 CMA IAC
3677 1041 TAD BLKFND
3700 7640 SZA CLA /BLOCKS CONSECUTIVE
3701 5555 JMP I SRCHER /NO, ERROR

```

3702	1041	SRFBLK,	TAD BLKFND	
3703	3160	DCA	PREBLK	
3704	1041	TAD	BLKFND	
3705	1156	TAD	TAPONT	
3706	7450	SNA		/BLOCKS=
3707	5663	JMP I	SRCONT	/YES, FOUND IT
3710	2263	ISZ	SRCONT	
3711	7120	CLL		
3712	7710	SPA	CLA	/BLKFND GREATER
3713	7120	STL		/NO, LESS
3714	6761	IOT	761	
3715	0137	AND	K0400	
3716	7430	SZL		
3717	5323	JMP	,+4	
3720	7640	SZA	CLA	/IF FWD HAVE GONE PAST
3721	2263	ISZ	SRCONT	/BKWD
3722	5663	JMP I	SRCONT	
3723	7650	SNA	CLA	
3724	2263	ISZ	SRCONT	
3725	5663	JMP I	SRCONT	
3726	6772	SREZTS,	IOT 772	
3727	0131	AND	EZBIT	
3730	7650	SNA	CLA	/END ZONE INTERRUPT
3731	5555	JMP I	SRCHER	/NO, SOME OTHER ERROR
3732	1156	TAD	TAPONT	
3733	7500	SMA		/BLOCK 0 OR 1
3734	5340	JMP	,+4	/YES HIT EZ AGAIN
3735	1153	TAD	K2701	
3736	7700	SMA	CLA	/BLOCK 2701 OR 2700
3737	5352	JMP	,+13	/NO, TURN AROUND
3740	7604	LAS		
3741	7010	RAR		
3742	7620	SNL	CLA	/NEW FORMAT TAPE
3743	5235	JMP	SRTARN	/YES TURN AROUND
3744	1113	TAD	K0200	/SET MOTION AGAIN
3745	6764	IOT	764	
3746	4466	JMS I	WAITI	/WAIT FOR EZ AGAIN
3747	5000	5000		
3750	5555	JMP I	SRCHER	
3751	5235	JMP	SRTARN	
3752	2157	ISZ	BLKFLG	/NOT FIRST INTERRUPT
3753	5555	JMP I	SRCHER	/EZ IS ERROR
3754	5235	JMP	SRTARN	/TURN AROUND

3755	1041	SRCMOD,	TAD BLKFND	/FIND DIFFERENCE
3756	7041	CMA IAC	/IV NUMBER OF BLOCKS	
3757	1040	TAD RECORD		
3760	7500	SMA		
3761	7041	CMA IAC	/MAKE -	
3762	3443	DCA I WCLDC	/FOR WORD COUNT	
3763	1135	TAD K0100		
3764	6764	IOT 764	/SET CONTIN MODE	
3765	4466	JMS I WAITI		
3766	0001	1		
3767	5555	JMP I SRCHER	/NOT NORMAL INTERRUPT	
3770	1041	TAD BLKFND		
3771	7041	CMA IAC		
3772	1040	TAD RECORD		
3773	7640	SZA CLA	/RIGHT BLOCK	
3774	5555	JMP I SRCHER	/NO. ERROR	
3775	1135	TAD K0100	/CLEAR CONTIN MODE	
3776	6764	IOT 764		
3777	5600	JMP I SEARCH		

/SEARCH ERROR TYPEOUT

*4000

4000	4467	SRHERR.	JMS I ERRSTP
4001	4545		JMS I SRCHTY
4002	1071		TAD RECRDK
4003	4506		JMS I TYPCON
4004	4507		JMS I TYPTX
4005	0042		42
4006	5457		5457
4007	4353		4353
4010	0067		67
4011	4156		4156
4012	6445		6445
4013	4400		4400
4014	7700		7700
4015	1076		TAD DIRFLG
4016	7650	SNA	CLA
4017	5222		JMP ,+3
4020	4714		JMS I BACKTY
4021	7410		SKP
4022	4715		JMS I FORDTY
4023	1042		TAD IDCON
4024	4506		JMS I TYPCON
4025	4507		JMS I TYPTX
4026	0042		42
4027	5457		5457
4030	4353		4353
4031	0046		46
4032	5765		5765
4033	5644		5644
4034	7700		7700
4035	7040		CMA
4036	1710		TAD I SEKONS
4037	7710		SPA CLA
4040	5252		JMP ,+12

/BLKF.G

4041	1311	TAD SEKONS+1	/PREBLK
4042	4506	JMS I TYPCON	
4043	4507	JMS I TYPTX	
4044	0054	54	
4045	4163	4163	
4046	6400	6400	
4047	4254	4254	
4050	5743	5743	
4051	5377	5377	
4052	2710	ISZ I SEKONS	
4053	7000	NOP	
4054	1310	TAD SEKONS	
4055	4506	JMS I TYPCON	
4056	4507	JMS I TYPTX	
4057	0042	42	
4060	5457	5457	
4061	4353	4353	
4062	6300	6300	
4063	6245	6245	
4064	4144	4144	
4065	7700	7700	
4066	6772	IOT 772	
4067	3313	DCA SEKONS+3	
4070	1312	TAD SEKONS+2	
4071	4506	JMS I TYPCON	
4072	4507	JMS I TYPTX	
4073	0063	63	
4074	6441	6441	
4075	6400	6400	
4076	4277	4277	
4077	7604	LAS	
4100	0154	AND K6000	
4101	7650	SNA CLA	
4102	7402	HLT	
4103	2151	ISZ PASFLG	/2ND ERROR
4104	5707	JMP I .+3	/REWIND DRIVE
4105	5706	JMP I .+1	
4106	3601	SEARCH+1	
4107	0525	REWDRV	
4110	0157	SEKONS, BLKFLG	
4111	0160	PREBLK	
4112	4113	.+1	
4113	0000	0	
4114	4547	BACKTY, TYBKW	
4115	4557	FORDTY, TYFWD	

/TYPE TEXT ROUTINE

```

*4200
4200 5200 TYTEXT,      JMP ,
4201 7200          CLA
4202 1600          TAD I TYTEXT      /GET NEXT 2 CHARACTERS
4203 7040          CMA                /MAKE -
4204 3320          DCA TXSTOR
4205 2200          ISZ TYTEXT
4206 1320          TAD TXSTOR
4207 7440          SZA                /CARRIAGE RETURN - LINE FEED
4210 5213          JMP ,+3           /NO

4211 4250          JMS CRLFLF        /CR LF
4212 5201          JMP TYTEXT+1     /GET NEXT
4213 0127          AND K7700        /CLEAR TO UPR CHAR
4214 7450          SNA                /END OF MESSAGE
4215 5600          JMP I TYTEXT     /YES
4216 7012          RTR                /MOVE
4217 7012          RTR                /OVER
4220 7012          RTR                /6 PLACES
4221 4230          JMS TYCHAR        /OUTPUT
4222 1320          TAD TXSTOR
4223 0134          AND K0077
4224 7450          SNA                /END OF MESSAGE
4225 5600          JMP I TYTEXT     /YES EXIT

4226 4230          JMS TYCHAR        /OUTPUT
4227 5201          JMP TYTEXT+1     /GET NEXT 2

4230 5230 TYCHAR,      JMP ,
4231 3247          DCA CRLFLF-1
4232 7604          LAS
4233 7710          SPA CLA
4234 5630          JMP I TYCHAR
4235 1247          TAD CRLFLF-1
4236 7040          CMA                /MAKE + AGAIN
4237 0134          AND K0077        /CLEAR TO LOWER 6
4240 1136          TAD K0240        /MAKE ASCII
4241 6046          TLS                /OUTPUT
4242 6041          TSF                /WAIT FLAG
4243 5242          JMP , -1
4244 7200          CLA
4245 6042          TCF                /CLEAR FLAG
4246 5630          JMP I TYCHAR     /DO NEXT

```

MAINDEC 08-D3RA-LA
Page 62

```
4247 0000      0
4250 5250  CRLF,      JMP ,
4251 7604      LAS
4252 7710      SPA CLA      /DELETE TYPEOJTS
4253 5650      JMP I CRLF
4254 1270      TAD K0215      /CARRIAGE RETURN
4255 6046      TLS
4256 6041      TSF
4257 5256      JMP , -1
4260 7200      CLA
4261 1271      TAD K0212      /LINE FEED
4262 6046      TLS
4263 6041      TSF
4264 5263      JMP , -1
4265 6042      TCF
4266 7200      CLA
4267 5650      JMP I CRLF
4270 0215      <0215,      215
4271 0212      <0212,      212
```

Page 00
/TYPE CONTENTS OF ADDRESS IN AC

4272 5272 TYCONT, JMP .
4273 3320 DCA TXSTOR /SAVE ADDRESS
4274 1720 TAD I TXSTOR /GET CONTENTS
4275 3320 DCA TXSTOR
4276 4250 JMS CRLF LF /CARRIAGE RETURN - LINE FEED
4277 4304 JMS TYCOVR /TYPE UPPER OCTAL
4300 4304 JMS TYCOVR
4301 4304 JMS TYCOVR
4302 4304 JMS TYCOVR
4303 5672 JMP I TYCONT

4304 5304 TYCOVR, JMP .
4305 1320 TAD TXSTOR
4306 7006 RTL
4307 7004 RAL
4310 3320 DCA TXSTOR
4311 1320 TAD TXSTOR
4312 7004 RAL
4313 0120 AND K0007
4314 1133 TAD K0020
4315 7040 CMA /MAKE - FOR
4316 4230 JMS TYCHAR /OUTPUT
4317 5704 JMP I TYCOVR
4320 0000 TXSTOR, 0

/STOP TAPE ON ERROR, LEAVE FLAGS SET
4321 5321 ERSTP, JMP .
4322 6761 IOT 761
4323 0113 AND K0200
4324 1125 TAD K0003
4325 6764 IOT 764
4326 5721 JMP I ERSTP

*4400

```
          /TYPE MOVE AND DIRECTION
4400 5000 TYMOVE,    JMP
4401 4311          JMS TYDRV
4402 4507          JMS I TYPTX
4403 7777          7777
4404 5557          5557
4405 6645          6645
4406 0077          0077
4407 4340          JMS TYDIR
4410 5600          JMP I TYMOVE
```

```
          /TYPE SEARCH DIRECTION AND MODE
4411 5211 TYSRCH,   JMP
4412 4311          JMS TYDRV
4413 4507          JMS I TYPTX
4414 6345          6345
4415 4162          4162
4416 4350          4350
4417 0077          0077
4420 4340          JMS TYDIR
4421 4365          JMS TYMODE
4422 5611          JMP I TYSRCH
```

```
          /TYPE READ DATA DIRECTION AND MODE
4423 5223 TYRDAT,   JMP
4424 4311          JMS TYDRV
4425 4257          JMS TYREAD
4426 4274          JMS TYDATA
4427 4340          JMS TYDIR
4430 4365          JMS TYMODE
4431 5623          JMP I TYRDAT
```

```
          /TYPE READ ALL DIRECTION AND MODE
4432 5232 TYRALL,   JMP
4433 4311          JMS TYDRV
4434 4257          JMS TYREAD
4435 4303          JMS TYALL
4436 4340          JMS TYDIR
4437 4365          JMS TYMODE
4440 5632          JMP I TYRALL
```

/TYPE WRITE DATA DIRECTION AND MODE

4441 5241 TYWDAT, JMP .
4442 4311 JMS TYDRV
4443 4265 JMS TYWRIT
4444 4274 JMS TYDATA
4445 4340 JMS TYDIR
4446 4365 JMS TYMODE
4447 5641 JMP I TYWDAT

/TYPE WRITE ALL DIRECTION AND MODE

4450 5250 TYWALL, JMP .
4451 4311 JMS TYDRV
4452 4265 JMS TYWRIT
4453 4303 JMS TYALL
4454 4340 JMS TYDIR
4455 4365 JMS TYMODE
4456 5650 JMP I TYWALL

/TYPE READ

4457 5257 TYREAD, JMP .
4460 4507 JMS I TYPTX
4461 6245 6245
4462 4144 4144
4463 0077 0077
4464 5657 JMP I TYREAD

/TYPE WRITE

4465 5265 TYWRIT, JMP .
4466 4507 JMS I TYPTX
4467 6762 6762
4470 5164 5164
4471 4500 4500
4472 7700 7700
4473 5665 JMP I TYWRIT

/TYPE DATA

4474 5274 TYDATA, JMP .
4475 4507 JMS I TYPTX
4476 0000 0
4477 4441 4441
4500 6441 6441
4501 0077 0077
4502 5674 JMP I TYDATA

```

/TYPE ALL
4503 5303 TYALL, JMP
4504 4507 JMS I TYPTX
4505 4154 4154
4506 5400 5400
4507 7700 7700
4510 5703 JMP I TYALL

```

```

/TYPE DRIVE AND NUMBER
4511 5311 TYDRV, JMP
4512 7604 LAS
4513 7710 SPA CLA
4514 5711 JMP I TYDRV
4515 4507 JMS I TYPTX
4516 7777 7777
4517 7777 7777
4520 4462 4462
4521 5166 5166
4522 4500 4500
4523 0077 0077
4524 1045 TAD CORIVE
4525 7450 SNA
4526 1132 TAD K0010
4527 1142 TAD K260
4530 6046 TLS
4531 6041 TSF
4532 5331 JMP , -1
4533 6042 TCF
4534 4507 JMS I TYPTX
4535 7777 7777
4536 7700 7700
4537 5711 JMP I TYDRV

```

```

/TYPE FORWARDS OR BACKWARD
4540 5340 TYDIR, JMP
4541 6761 IOT 761
4542 0137 AND K0400
4543 7650 SNA CLA
4544 5355 JMP TYFWD-2
4545 4347 JMS TYBKW
4546 5740 JMP I TYDIR
4547 5347 TYBKW, JMP
4550 4507 JMS I TYPTX
4551 4253 4253
4552 6744 6744
4553 0077 0077
4554 5747 JMP I TYBKW

```

```

4555 4357 JMS TYFWD
4556 5740 JMP I TYDIR
4557 5357 TYFWD, JMP
4560 4507 JMS I TYPTX
4561 4667 4667
4562 4400 4400
4563 7700 7700

```

/TYPE CONTINUOUS IF NOT NORMAL MOJE

4565	5365	TYMODE, JMP
4566	6761	IOT 761
4567	0135	AND K0100
4570	7650	SNA CLA
4571	5765	JMP I TYMODE
4572	4507	JMS I TYPTX
4573	4300	4300
4574	5557	5557
4575	4445	4445
4576	0077	0077
4577	5765	JMP I TYMODE

ACSAVE	3127
BACKTY	4114
BF1LOC	0075
BF2LOC	0103
BLKFLG	0157
BLKFND	0041
BUFFRS	7175
CALCTY	2670
CALOC	0044
CDRIVE	0045
CHNGDR	3437
CKSERR	2560
CKSTYP	2657
CKSUMR	2564
COCNTR	0072
CODATA	1225
CODEX	1255
COERRO	1261
COERR1	1313
COLOOP	1243
COMBIT	0050
COMSTR	0755
CORT1R	1442
CRLFLF	4250
CSECON	2701
DATAO	0065
DATAMV	0070
DATASM	2561
DIRECT	0052
DIRFLG	0076
DIRTBL	1044
DRIVTY	0130
ERRCKS	2600
ERRRAR	2707
ERRSTP	0067
ERSTP	4321
EXIST	1025
EZBIT	0131
FORDTY	4115
FRSWAT	0102
FWDCHK	2562
GENDEX	0675
GENPAT	0061
GENRAN	0735

MAINDEC-08-D3RA-LA

ACSAVE 3127
 BACKTY 4114
 BF1LOC 0075
 BF2LOC 0103
 BLKFLG 0157
 BLKFND 0041
 BUFFRS 7175
 CALCTY 2670
 CALOC 0044
 CDRIVE 0045
 CHNGDR 3437
 CKSERR 2560
 CKSTYP 2657
 CKSUMR 2564
 COCNTR 0072
 CODATA 1225
 CODEX 1255
 COERRO 1261
 COERR1 1313
 COLOOP 1243
 COMBIT 0050
 COMSTR 0755
 CORT1R 1442
 CRLFLF 4250
 CSECON 2701
 DATACO 0065
 DATAMV 0070
 DATASM 2561
 DIRECT 0052
 DIRFLG 0076
 DIRTBL 1044
 DRIVTY 0130
 ERRCKS 2600
 ERRRAR 2707
 ERRSTP 0067
 ERSTP 4321
 EXIST 1025
 EZBIT 0131
 FORDTY 4115
 FRSWAL 0162
 FWDCHK 2562
 GENDEX 0675
 GENPAT 0061
 GENRAN 0735
 GETRED 0570
 GNPATR 0626
 GNPTRS 3461
 IDCON 0042
 IOT 6000
 IRECD 3034
 ISZTST 3200
 JMRETU 3321
 JMSKON 3352
 JMSLOC 3351
 JMSTST 3300
 K0002 0152
 K0003 0125
 K0007 0120

K0010 0132
 K0020 0133
 K0030 0116
 K0037 0111
 K0040 0114
 K0050 0150
 K0070 0123
 K0077 0134
 K0100 0135
 K0120 2574
 K0130 2113
 K0150 2301
 K0200 0113
 K0212 4271
 K0214 0126
 K0215 4270
 K0240 0136
 K0400 0137
 K0604 0140
 K0614 0115
 K0700 0122
 K1000 0131
 K16 0141
 K260 0142
 K2701 0153
 K3777 0143
 K4000 0121
 K5075 0146
 K5076 0147
 K6000 0154
 K7000 0124
 K7175 2300
 K7571 2575
 K7577 0117
 K7600 0543
 K7600X 3353
 K7700 0127
 K7760 0110
 K7763X 3354
 K7770 0112
 K7772Y 3141
 K7774 1354
 LKSAVE 3126
 LSTBLK 0053
 LSTDRV 0054
 LSTTBL 1045
 MCOMOB 0703
 MIN4 0753
 MIN60 0754
 MOFPRO 0221
 MOFPR1 0235
 MOVDEX 1222
 MOVER 3540
 MSBITS 0047
 MVDATA 1200
 NBIT8 3500
 NDOFRD 1725
 NDPST 3350
 NDTSKP 3055
 NEWDRV 0057

NUMBLK 0077
 PASFLG 0151
 PATGEN 0600
 PICNTR 3142
 PIDEX1 3143
 PIDE_X2 3144
 PIRTBL 3154
 PITBL 3145
 PNTRS 3501
 POSITN 0051
 POSTBL 1043
 PRAN1 3346
 PRAN2 3347
 PREBLK 0160
 PREGEN 0547
 PTSTND 3064
 PTSTNS 3104
 RADCHK 2512
 RADLOC 2576
 RAEFLG 2577
 RALLTS 2400
 RANBFR 6774
 RANDEX 0200
 RANGEN 0062
 RANN0 0751
 RANSEL 1000
 RANVAR 0752
 RARERR 2570
 RASERR 2571
 RBFLOC 0074
 RBFWD2 0102
 RBUFND 0100
 RBUFST 0073
 RDBUF1 1632
 RDCERR 2075
 RDCMOD 2000
 RDDATA 0400
 RDSEQ 1600
 RDTAB 0304
 READTY 0144
 READ1 0063
 READ2 0104
 READ2A 0105
 RECORD 0040
 RECRDK 0071
 REGENP 0064
 REPOSI 3400
 REREAD 0420
 REREDA 0470
 RESETD 3017
 RETJMP 3355
 RETUJM 0161
 REVCHK 2563
 REWCK 1517
 REWDRV 0525
 REWIND 0056
 ROT1TS 3222
 ROT2TS 3236
 RPATO 0700
 RSFDRV 3417

MAINDEC-08-D3RA-LA

RSFDR1 3435
 RSQFLG 0350
 SAMDRV 0325
 SAVNUM 1570
 SBRECD 3130
 SEARCH 3600
 SEKONS 4110
 SELGEN 1046
 SELRAN 0060
 SELVAR 1061
 SETIDX 3131
 SRCHER 0155
 SRCHIT 0055
 SRCHTY 0145
 SRCMOD 3755
 SRCONT 3663
 SREZTS 3726
 SRFBLK 3702
 SRHERR 4000
 SRTARN 3635
 SWCMDL 2223
 SWCMOD 2212
 TADTST 3252
 TAPONT 0156
 TEMP1 3341
 TEMP2 3342
 TEMP3 3343
 TEMP4 3344
 TEMP5 3345
 TRECTR 2277
 TXSTOR 4320
 TYALL 4503
 TYBKW 4547
 TYCHAR 4230
 TYCONT 4272
 TYCOVR 4304
 TYDATA 4474
 TYDIR 4540
 TYDRV 4511
 TYFWD 4557
 TYMODE 4565
 TYMOVE 4400
 TYPCON 0106
 TYPTEX 0107
 TYRALL 4432
 TYRDAT 4423
 TYREAD 4457
 TYSRCH 4411
 TYTEXT 4200
 TYWALL 4450
 TYWDAT 4441
 TYWRIT 4465
 UNFUNC 0046
 WAITI 0066
 WATINT 3000
 WCLOC 0043
 WDCERR 2253
 WDCMOD 2200
 WDINC 1503
 WRITES 0324

GNPATR 0626
 GNPTRS 3461
 IDCON 0042
 IOT 6000
 IRECD 3034
 ISZTST 3200
 JMRETU 3321
 JMSKON 3352
 JMSLOC 3351
 JMSTST 3300
 K0002 0152
 K0003 0125
 K0007 0120
 K0010 0132
 K0020 0133
 K0030 0116
 K0037 0111
 K0040 0114
 K0050 0150
 K0070 0123
 K0077 0134
 K0100 0135
 K0120 2574
 K0130 2113
 K0150 2301
 K0200 0113
 K0212 4271
 K0214 0126
 K0215 4270
 K0240 0136
 K0400 0137
 K0604 0140
 K0614 0115
 K0700 0122
 K1000 0131
 K16 0141
 K260 0142
 K2701 0153
 K3777 0143
 K4000 0121
 K5075 0146
 K5076 0147
 K6000 0154
 K7000 0124
 K7175 2300
 K7571 2575
 K7577 0117
 K7600 0543
 K7600X 3353
 K7700 0127
 K7760 0110
 K7763X 3354
 K7770 0112
 K7772Y 3141
 K7774 1354
 LKSAVE 3126
 LSTBLK 0053
 LSTDRV 0054
 LSTTBL 1045
 MCOMOB 0703

MIN4 0753
 MIN60 0754
 MOFPRO 0221
 MOFPR1 0235
 MOVDEX 1222
 MOVER 3540
 MSBITS 0047
 MVDATA 1200
 NBIT8 3500
 NDOFRD 1725
 NDPTST 3350
 NDTSKP 3055
 NEWDRV 0057
 NUMBLK 0077
 PASFLG 0151
 PATGEN 0600
 PICNTR 3142
 PIDEK1 3143
 PIDEK2 3144
 PIRTBL 3154
 PITBL 3145
 PNTRS 3501
 POSITN 0051
 POSTBL 1043
 PRAN1 3346
 PRAN2 3347
 PREBLK 0160
 PREGEN 0547
 PTSTND 3064
 PTSTNS 3104
 RADCHK 2512
 RADLOC 2576
 RAEFLG 2577
 RALLTS 2400
 RANBFR 6774
 RANDEX 0200
 RANGEN 0062
 RANN0 0751
 RANSEL 1000
 RANVAR 0752
 RARERR 2570
 RASERR 2571
 RBFLOC 0074
 RBFWD2 0102
 RBUFND 0100
 RBUFST 0073
 RDBUF1 1632
 RDCERR 2075
 RDCMOD 2000
 RDDATA 0400
 RDSEQ 1600
 RDTAB 0304
 READTY 0144
 READ1 0063
 READ2 0104
 READ2A 0105
 RECORD 0040
 RECROK 0071
 REGENP 0064
 REPOSI 3400

MAINDEC-08-D3RA-LA

REREAD	0420	TEMP5	3345
REREDA	0470	TRECTR	2277
RESETD	3017	TXSTOR	4320
RETJMP	3355	TYALL	4503
RETUJM	0161	TYBKW	4547
REVCHK	2563	TYCHAR	4230
REWCK	1517	TYCONT	4272
REWDRV	0525	TYCOVR	4304
REWIND	0056	TYDATA	4474
ROT1TS	3222	TYDIR	4540
ROT2TS	3236	TYDRV	4511
RPATO	0700	TYFWD	4557
RSFDRV	3417	TYMODE	4565
RSFDR1	3435	TYMOVE	4400
RSQFLG	0350	TYPCON	0106
SAMDRV	0325	TYPTX	0107
SAVNUM	1570	TYRALL	4432
SBRECD	3130	TYRDAT	4423
SEARCH	3600	TYREAD	4457
SEKONS	4110	TYSRCH	4411
SELGEN	1046	TYTEXT	4200
SELRAN	0060	TYWALL	4450
SELVAR	1061	TYWDAT	4441
SETIDX	3131	TYWRIT	4465
SRCHER	0155	UNFUNC	0046
SRCHIT	0055	WAITI	0066
SRCHTY	0145	WATINT	3000
SRCMOD	3755	WCLOC	0043
SRCNT	3663	WDCERR	2253
SREZTS	3726	WDCMOD	2200
SRFBLK	3702	WDINC	1503
SRHERR	4000	WRITES	0324
SRTARN	3635	WRITE1	1400
SWCMDL	2223	WRTDTY	0101
SWCMOD	2212	WRT1A	1455
TADTST	3252	WSTERR	1526
TAPONT	0156	WTCNTR	3125
TEMP1	3341	WTKON1	3123
TEMP2	3342	WTKON2	3124
TEMP3	3343	XORSAV	2565
TEMP4	3344	XORSUM	2531

11. FLOW CHARTS

