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

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 (2702_8 129-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:

Addresses	Contents
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 b'ock 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:

Address	
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 itself + 1 after executing a JMS.

6.1.7 Examples of Error Typeouts -

6.1.7.1 Example 1 Search Errors -

```
DRIVE 8

SEARCH FWD
2677 BLOCK WANTED FWD         (Block searched)
2700 BLOCK FOUND              (Block found)
0001 BLOCKS READ              (One block number received)
0001 STAT B                   (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 typeout 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 typeout 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 typeout 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	A mark-track error was received while doing a write data on block 1.
0001 BLOCK	
6000 STAT B	
7743 WC	
DRIVE 1	
WRITE DATA BKWD C MODE	
0040 BLOCK	
0001 STAT B	(Note: Normal STAT B)
7400 WC	(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	(The complements of the REVCK, DATA and CKSUM did not XOR to 77)
0033 REV CKSUM	(as read from tape)
0022 DATA CKSUM CALCULATED	(XOR of the data's complement)
6700 CKSUM	(as read from tape)
0076 CALCULATED CKSUM	(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                  (or READ ALL C MODE)
DATA ERROR
0265 BLOCK

4632 COR                           (Pattern word generated)
4432 INC                           (Pattern word read)
7237 ADRS INC                      (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 attemps 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.

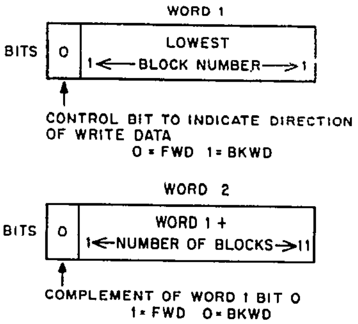
For the read errors (read data and read all) the direction of the second read operation is opposite the direction that the error was originally detected in. If a second error is detected, a typeout and halt again occur. Pressing CONTINUE causes the drive to be rewound to the end zone. If no error occurs 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.

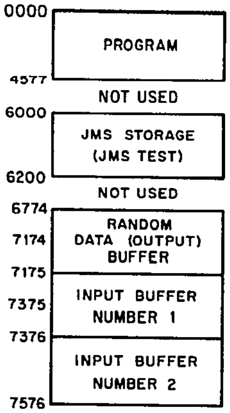
WORD 2 is 1 greater than the highest numbered block in the series bit 0 in word 2 is the complement of bit 0 in WORD 1.

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  14ØØ
WRTDTY  Ø1Ø1
WRT1A   1455
WSTERR  1526
WTCNTR  3125
WTKON1  3123
WTKON2  3124
XORSAV  2565
XORSUM  2531
```

```
                /TCØ1 RANDOM EXERCISER
                /PAGE Ø CONSTANTS AND TEMP STOR.
                RANBFR=6774
                BUFFRS=RANBFR+2Ø1
                IOT=6ØØØ

                /GO TO PROCESS INTERRUPT

                *1
ØØØ1  54Ø2          JMP I .+1
ØØØ2  3Ø34          IRECD

                *4Ø

ØØ4Ø  ØØØØ      RECORD,     Ø               /SEARCH ID
ØØ41  ØØØØ      BLKFND,     Ø               /BLOCK FROM TCØ1
ØØ42  ØØ41      IDCON,      .-1             /FOR SRCH CA
ØØ43  7754      WCLOC,      7754            /TO STORE WC
ØØ44  7755      CALOC,      7755            /TO STORE CA
ØØ45  ØØØØ      CDRIVE,     Ø               /CURRENT DRIVE NUMBER
ØØ46  ØØØØ      UNFUNC,     Ø               /SAME FOR TCØ1 COMMAND

ØØ47  ØØØØ      MSBITS,     Ø               /DRIVE SELECT SWITCHES
ØØ5Ø  ØØØØ      COMBIT,     Ø
ØØ51  ØØØØ      POSITN,     Ø               /TO GET DRIVE POSITION
ØØ52  ØØØØ      DIRECT,     Ø               /TO GET DRIVE DIRECTION
ØØ53  ØØØØ      LSTBLK,     Ø               /TO GET LAST BLOCK WRITTEN
ØØ54  ØØØØ      LSTDRV,     Ø               /=LAST DRV SELECTED
```

```
                /SUBROUTINE ADDRESES

ØØ55    36ØØ    SRCHIT,       SEARCH
ØØ56    34ØØ    REWIND,       REPOSI
ØØ57    3437    NEWDRV,       CHNGDR
ØØ6Ø    1ØØØ    SELRAN,       RANSEL
ØØ61    Ø6ØØ    GENPAT,       PATGEN
ØØ62    Ø735    RANGEN,       GENRAN
ØØ63    Ø4ØØ    READ1,        RDDATA
ØØ64    Ø547    REGENP,       PREGEN
ØØ65    1225    DATACO,       CODATA
ØØ66    3ØØØ    WAITI,        WATINT
ØØ67    4321    ERRSTP,       ERSTP
ØØ7Ø    12ØØ    DATAMV,       MVDATA
ØØ71    ØØ4Ø    RECRDK,       RECORD
ØØ72    7777    COCNTR,       7777              /-1 NO DATA ERRORS
ØØ73    6774    RBUFST,       RANBFR
ØØ74    6773    RBFLOC,       RANBFR-1
ØØ75    7174    BF1LOC,       BUFFRS-1
ØØ76    ØØØØ    DIRFLG,       Ø
ØØ77    ØØØØ    NUMBLK,       Ø
Ø1ØØ    7174    RBUFND,       RANBFR+2ØØ
Ø1Ø1    4441    WRTDTY,       TYWDAT
Ø1Ø2    6775    RBFWD2,       RANBFR+1
Ø1Ø3    7376    BF2LOC,       BUFFRS+2Ø1
Ø1Ø4    Ø42Ø    READ2,        REREAD
Ø1Ø5    Ø47Ø    READ2A,       REREDA
Ø1Ø6    4272    TYPCON,       TYCONT
Ø1Ø7    42ØØ    TYPTEX,       TYTEXT
```

```
                /MASK AND COMPARE CONSTANTS

Ø11Ø  776Ø  K776Ø,      776Ø
Ø111  ØØ37  KØØ37,      37
Ø112  777Ø  K777Ø,      777Ø
Ø113  Ø2ØØ  KØ2ØØ,      2ØØ
Ø114  ØØ4Ø  KØØ4Ø,      4Ø
Ø115  Ø614  KØ614,      614
Ø116  ØØ3Ø  KØØ3Ø,      3Ø
Ø117  7577  K7577,      7577

Ø12Ø  ØØØ7  KØØØ7,      7
Ø121  4ØØØ  K4ØØØ,      4ØØØ
Ø122  Ø7ØØ  KØ7ØØ,      7ØØ
Ø123  ØØ7Ø  KØØ7Ø,      7Ø
Ø124  7ØØØ  K7ØØØ,      7ØØØ
Ø125  ØØØ3  KØØØ3,      3
Ø126  Ø214  KØ214,      214
Ø127  77ØØ  K77ØØ,      77ØØ
Ø13Ø  4511  DRIVTY,     TYDRV
Ø131  1ØØØ  K1ØØØ,      1ØØØ
            EZBIT=K1ØØØ
Ø132  ØØ1Ø  KØØ1Ø,      1Ø
Ø133  ØØ2Ø  KØØ2Ø,      2Ø
Ø134  ØØ77  KØØ77,      77
Ø135  Ø1ØØ  KØ1ØØ,      1ØØ
Ø136  Ø24Ø  KØ24Ø,      24Ø
Ø137  Ø4ØØ  KØ4ØØ,      4ØØ
Ø14Ø  Ø6Ø4  KØ6Ø4,      6Ø4
Ø141  ØØ16  K16,        16
Ø142  Ø26Ø  K26Ø,       26Ø
Ø143  3777  K3777,      3777
Ø144  4423  READTY,     TYRDAT
Ø145  4411  SRCHTY,     TYSRCH
Ø146  5Ø75  K5Ø75,      5Ø75
Ø147  5Ø76  K5Ø76,      5Ø76
Ø15Ø  ØØ5Ø  KØØ5Ø,      5Ø
Ø151  ØØØØ  PASFLG,     Ø
Ø152  ØØØ2  KØØØ2,      2
Ø153  27Ø1  K27Ø1,      27Ø1
Ø154  6ØØØ  K6ØØØ,      6ØØØ

            /CONSTANTS FOR SEARCH ROUTINE
Ø155  4ØØØ  SRCHER,     SRHERR
Ø156  ØØØØ  TAPONT,     Ø
Ø157  ØØØØ  BLKFLG,     Ø
Ø16Ø  ØØØØ  PREBLK,     Ø

            /JMS RETURN FOR PROCESSOR TEST

Ø161  3321  RETUJM,     JMRETU
Ø162  ØØØØ  FRSWAT,     Ø
```

```
                /TCØ1 DECTAPE
                /RANDOM DATA RANDOM SEQUENCE EXERCISER
                /TESTS CONCURRENT OPERATION OF ANY CONFIGURATION OF 1 TO 8 DECTAPE
                /DRIVES.
                /MASTER BIT SELECTION BIT Ø=DRV8 BIT1=DRV1 TO BIT7=DRV7

                *2ØØ

Ø2ØØ    76Ø4    RANDEX,     CLA OSR
Ø2Ø1    Ø11Ø        AND K776Ø
Ø2Ø2    744Ø        SZA                 /SELECT ANY AT ALL
Ø2Ø3    52Ø6        JMP .+3             /YES
Ø2Ø4    74Ø2        HLT                 /DIT NOT SELECT ANY DRIVES
Ø2Ø5    52ØØ        JMP RANDEX
Ø2Ø6    3Ø47        DCA MSBITS
Ø2Ø7    74Ø2        HLT                 /WAIT CLR SELECTION
Ø21Ø    7Ø4Ø        CMA
Ø211    3162        DCA FRSWAT
Ø212    6774        IOT 774             /CLEAR STAT B
Ø213    4456        JMS I REWIND        /PUT ALL DRIVES IN END ZONE

Ø214    3453        DCA I LSTBLK        /CLR NUMBER OF BLOCKS
Ø215    4457        JMS I NEWDRV        /CLRD ALL
Ø216    5214        JMP .-2             /NO
Ø217    72Ø1        IAC CLA
Ø22Ø    3Ø54        DCA LSTDRV

                /MEAT OF THE PROGRAM
                /RANDOM SELECTION OF OPERATIONS
                /FIRST SELECT DRIVE
Ø221    446Ø    MOFPRO,     JMS I SELRAN    /RANDOMLY SELECT A DRIVE
Ø222    724Ø        CLA CMA
Ø223    3151        DCA PASFLG
Ø224    1Ø45        TAD CDRIVE
Ø225    1Ø54        TAD LSTDRV
Ø226    765Ø        SNA CLA
Ø227    5325        JMP SAMDRV
Ø23Ø    1Ø45        TAD CDRIVE
Ø231    7Ø41        CMA IAC
Ø232    3Ø54        DCA LSTDRV

Ø233    7Ø4Ø        CMA
Ø234    335Ø        DCA RSQFLG
                /SELECT 1 TO 32 BLOCKS FOR OPERATION
Ø235    4462    MOFPR1,     JMS I RANGEN    /GET RAN NUMBER
Ø236    Ø111        AND KØØ37
Ø237    7Ø4Ø        CMA                 /MAKE -1 TO -32 DECI
Ø24Ø    3Ø77        DCA NUMBLK
```

```
                /SELECT DIRECTION TO GO
Ø241  4462          JMS I RANGEN    /GET RANDOM NUMBER
Ø242  Ø125          AND KØØØ3
Ø243  765Ø          SNA CLA
Ø244  7Ø4Ø          CMA      /SET BACKWARDS
Ø245  3Ø76          DCA DIRFLG                /DIRECTION FLAG
Ø246  1453          TAD I LSTBLK
Ø247  765Ø          SNA CLA         /DRIVE BEEN WRITTEN ON
Ø25Ø  5724          JMP I WRITES    /NO, WRITE
Ø251  71ØØ          CLL
Ø252  1Ø76          TAD DIRFLG

                /GENERATE BLOCK SELECT POS + OR-NUMBER
Ø253  764Ø          SZA CLA
Ø254  712Ø          STL
Ø255  1Ø77          TAD NUMBLK
Ø256  742Ø          SNL
Ø257  7Ø41          CMA IAC
Ø26Ø  1451          TAD I POSITN
Ø261  3Ø4Ø          DCA RECORD
Ø262  1Ø4Ø          TAD RECORD
Ø263  771Ø          SPA CLA /-RECORD
Ø264  3Ø4Ø          DCA RECORD      /MAKE =Ø
                /HAS BLOCK SELECTED BEEN WRITTEN
                /IF NOT WRITE OPERATION SELECTED
Ø265  1Ø4Ø          TAD RECORD
Ø266  7Ø4Ø          CMA
Ø267  1453          TAD I LSTBLK
Ø27Ø  771Ø          SPA CLA         /BLOCK BEEN WRITTEN
Ø271  5724          JMP I WRITES    /NO
Ø272  4462          JMS I RANGEN
Ø273  Ø12Ø          AND KØØØ7
Ø274  745Ø          SNA
Ø275  57Ø3          JMP I .+6
Ø276  Ø125          AND KØØØ3
Ø277  765Ø          SNA CLA
Ø3ØØ  53Ø4          JMP .+4
Ø3Ø1  57Ø2          JMP I .+1
Ø3Ø2  2ØØØ          RDCMOD
Ø3Ø3  24ØØ          RALLTS
Ø3Ø4  4463      RDTAB,      JMS I READ1
Ø3Ø5  55Ø4          JMP I READ2     /STATUS ERROR RETURN
```

```
                /BLOCK HAS BEEN READ WITHOUT PAR ERR
Ø3Ø6   1113         TAD K Ø2ØØ
Ø3Ø7   6764         IOT 764          /STOP TAPE
Ø31Ø   1Ø4Ø         TAD RECORD
Ø311   3451         DCA I POSITN     /NEW POSITION
Ø312   1Ø76         TAD DIRFLG
Ø313   3452         DCA I DIRECT     /DIRECTION READ

                /MOVE FIRST FOUR WORDS TO REGENERATE
Ø314   4464         JMS I REGENP     /REGERATE DATA PATTERN
Ø315   4465         JMS I DATACO
Ø316   6774         RANBFR
Ø317   7175         BUFFRS
Ø32Ø   Ø2Ø1         2Ø1
Ø321   2Ø72         ISZ COCNTR
Ø322   55Ø5         JMP I READ2A
Ø323   5221         JMP MOFPRO                /SELECT NEW DRIVE
Ø324   14ØØ     WRITES,     WRITE1

                /DRIVE SELECTED SAME AS LAST TIME TEST FOR SERIES
Ø325   4462     SAMDRV,     JMS I RANGEN
Ø326   751Ø         SPA              /READ THIS STRING
Ø327   5236         JMP MOFPR1+1     /FIND NEW BLOCK
Ø33Ø   235Ø         ISZ RSQFLG
Ø331   5236         JMP MOFPR1+1
Ø332   Ø125         AND KØØØ3
Ø333   765Ø         SNA CLA          /FWD
Ø334   7Ø4Ø         CMA              /NO GO BACKWARD
Ø335   3Ø76         DCA DIRFLG
Ø336   1Ø76         TAD DIRFLG
Ø337   7Ø41         CMA IAC
Ø34Ø   1Ø73         TAD RBUFST                /FWD GET
Ø341   3Ø4Ø         DCA RECORD                /FIRST BLOCK
Ø342   1Ø76         TAD DIRFLG                /BKWD GETS
Ø343   144Ø         TAD I RECORD     /LST BLOCK+1-1
Ø344   Ø143         AND K3777
Ø345   3Ø4Ø         DCA RECORD
Ø346   5747         JMP I .+1
Ø347   16ØØ         RDSEQ
Ø35Ø   ØØØØ     RSQFLG,     Ø
```

```
                /READ DATA SUBROUTINE 1 BLOCK
                /DIRFLG=7777 IS BACKWARDS=Ø IS FWD

                *4ØØ

Ø4ØØ   52ØØ    RDDATA,     JMP .
Ø4Ø1   72ØØ        CLA
Ø4Ø2   4455        JMS I SRCHIT
Ø4Ø3   1116        TAD KØØ3Ø        /SEARCH TO READ DATA
Ø4Ø4   6764        IOT 764
Ø4Ø5   1117        TAD K7577                  /129 WORDS
Ø4Ø6   3443        DCA I WCLOC
Ø4Ø7   1Ø75        TAD BF1LOC             /INTO FIRST BUFFER
Ø41Ø   3444        DCA I CALOC
Ø411   4466        JMS I WAITI
Ø412   ØØØ1        1
Ø413   56ØØ        JMP I RDDATA           /STATUS ERR(READ EXIT)
Ø414   1443        TAD I WCLOC
Ø415   765Ø        SNA CLA
Ø416   22ØØ        ISZ RDDATA
Ø417   56ØØ        JMP I RDDATA           /NORMAL READ EXIT
```

```
                /TYPE STATUS ERROR ON READ 1 BLJCK
                /TEST FOR DATA ERRORS IF FULL REAJ

Ø42Ø   724Ø   REREAD,    CLA CMA
Ø421   3151        DCA PASFLG
Ø422   4467        JMS I ERRSTP     /STOP TAPE
Ø423   4544        JMS I READTY
Ø424   1Ø71        TAD RECRDK
Ø425   45Ø6        JMS I TYPCON     /TYPE BLOCK NJMBER
Ø426   45Ø7        JMS I TYPTEX
Ø427   ØØ42        ØØ42

Ø43Ø   5457        5457
Ø431   4353        4353
Ø432   77ØØ        77ØØ
Ø433   6772        IOT 772
Ø434   3Ø1Ø        DCA 1Ø
Ø435   1132        TAD KØØ1Ø
Ø436   45Ø6        JMS I TYPCON             /TYPE STATUS B
Ø437   45Ø7        JMS I TYPTEX
Ø44Ø   ØØ63        ØØ63
Ø441   6441        6441
Ø442   64ØØ        64ØØ
Ø443   42ØØ        42ØØ
Ø444   7777        7777
Ø445   77ØØ        77ØØ
Ø446   1443        TAD I WCLOC
Ø447   765Ø        SNA CLA          /READ 129 WJRJ

Ø45Ø   526Ø        JMP .+1Ø                 /YES
Ø451   1Ø43        TAD WCLOC
Ø452   45Ø6        JMS I TYPCON     /SHORT BUFFER
Ø453   45Ø7        JMS I TYPTEX     /TYPEOUT
Ø454   ØØ67        ØØ67
Ø455   1643        1643
Ø456   1677        1677
Ø457   527Ø        JMP REREDA
Ø46Ø   4464        JMS I REGENP     /REGENERATE PATTERN
Ø461   1444        TAD I CALOC
Ø462   1343        TAD K76ØØ
Ø463   3266        DCA .+3
Ø464   4465        JMS I DATACO             /COMPARE FOR TYPEOUTS
Ø465   6774        RANBFR
Ø466   7175        BUFFRS
Ø467   Ø2Ø1        Ø2Ø1
```

```
Ø47Ø  76Ø4   REREDA,     LAS
Ø471  Ø154         AND K6ØØØ
Ø472  765Ø         SNA CLA
Ø473  74Ø2         HLT
Ø474  2151         ISZ PASFLG
Ø475  5325         JMP REWDRV
Ø476  1Ø76         TAD DIRFLG
Ø477  7Ø4Ø         CMA
Ø5ØØ  3Ø76         DCA DIRFLG
Ø5Ø1  4463         JMS I READ1      /READ OPPOSITE DIRECTION
Ø5Ø2  5222         JMP REREAD+2     /ERROR, AGAIN, TRY OVER
Ø5Ø3  4464         JMS I REGENP     /REGEN PATTERN
Ø5Ø4  4465         JMS I DATACO     /COMPARE
Ø5Ø5  6774         RANBFR
Ø5Ø6  7175         BUFFRS
Ø5Ø7  Ø2Ø1         Ø2Ø1

Ø51Ø  2Ø72         ISZ COCNTR
Ø511  527Ø         JMP REREDA

Ø512  1Ø76         TAD DIRFLG
Ø513  7Ø4Ø         CMA
Ø514  3Ø76         DCA DIRFLG
Ø515  4463         JMS I READ1

Ø516  5222         JMP REREAD+2     /ANOTHER STATUS ERROR
Ø517  4465         JMS I DATACO
Ø52Ø  6774         RANBFR
Ø521  7175         BUFFRS
Ø522  Ø2Ø1         Ø2Ø1
Ø523  2Ø72         ISZ COCNTR
Ø524  527Ø         JMP REREDA

Ø525  114Ø   REWDRV,     TAD KØ6Ø4
Ø526  1Ø46         TAD UNFUNC
Ø527  6766         IOT 766
Ø53Ø  4466         JMS I WAITI
Ø531  5ØØØ         5ØØØ
Ø532  5344         JMP K76ØØ+1
Ø533  3453         DCA I LSTBLK
Ø534  3451         DCA I POSITN
Ø535  7Ø4Ø         CMA
Ø536  3452         DCA I DIRECT
Ø537  7ØØ1         IAC
Ø54Ø  3Ø54         DCA LSTDRV
Ø541  5742         JMP I .+1
Ø542  Ø221         MOFPRO
Ø543  76ØØ   K76ØØ,      76ØØ
Ø544  4746         JMS I .+2
Ø545  5325         JMP REWDRV
Ø546  3542         MOVER+2
```

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

Ø547  5347      PREGEN,     JMP .
Ø55Ø  1Ø74          TAD RBFLOC
Ø551  3Ø1Ø          DCA 1Ø
Ø552  1372          TAD GETRED+2
Ø553  3Ø17          DCA 17
Ø554  1444          TAD I CALOC
Ø555  337Ø          DCA GETRED
Ø556  177Ø          TAD I GETRED
Ø557  4771          JMS I GETRED+1
Ø56Ø  341Ø          DCA I 1Ø
Ø561  7Ø4Ø          CMA
Ø562  137Ø          TAD GETRED
Ø563  337Ø          DCA GETRED
Ø564  2Ø17          ISZ 17
Ø565  5356          JMP .-7
Ø566  4461          JMS I GENPAT
Ø567  5747          JMP I PREGEN
Ø57Ø  ØØØØ      GETRED,     Ø
Ø571  Ø7Ø3          MCOMOB
Ø572  7774          7774
```

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

                  *6ØØ
Ø6ØØ    52ØØ      PATGEN,    JMP .
Ø6Ø1    1353         TAD MIN4
Ø6Ø2    3275         DCA GENDEX       /TO COUNT FIRST 4 WORDS
Ø6Ø3    1Ø73         TAD RBUFST
Ø6Ø4    3276         DCA GENDEX+1     /TO STORE FROM START
Ø6Ø5    11ØØ         TAD RBUFND
Ø6Ø6    3277         DCA GENDEX+2     /TO STORE FROM END
Ø6Ø7    13Ø1         TAD RPATO+1      /TO GET
Ø61Ø    33ØØ         DCA RPATO                  /RANDOMS
Ø611    1676         TAD I GENDEX+1
Ø612    33Ø1         DCA RPATO+1
Ø613    13Ø1         TAD RPATO+1
Ø614    43Ø3         JMS MCOMOB                 /MAKE COMPLIMENT OBVERSE
Ø615    3677         DCA I GENDEX+2
Ø616    2276         ISZ GENDEX+1     /INC ADDRESS
Ø617    724Ø         CLA CMA
Ø62Ø    1277         TAD GENDEX+2     /DECREMENT OTHER
Ø621    2275         ISZ GENDEX       /DONE FIRST 4
Ø622    52Ø6         JMP PATGEN+6     /NO

Ø623    3277         DCA GENDEX+2
Ø624    1354         TAD MIN6Ø                  /TO COUNT 6Ø WORDS
Ø625    3275         DCA GENDEX
Ø626    13ØØ      GNPATR,    TAD RPATO
Ø627    711Ø         CLL RAR          /FORM NEXT
Ø63Ø    743Ø         SZL              /WORD OF
Ø631    1121         TAD K4ØØØ                  /RANDOM PATTERN
Ø632    33Ø2         DCA RPATO+2

Ø633    13Ø2         TAD RPATO+2
Ø634    3676         DCA I GENDEX+1
Ø635    13Ø2         TAD RPATO+2
Ø636    43Ø3         JMS MCOMOB
Ø637    3677         DCA I GENDEX+2
Ø64Ø    13Ø1         TAD RPATO+1      /MOVE WORDS
Ø641    33ØØ         DCA RPATO                  /FOR NEXT PASS
Ø642    13Ø2         TAD RPATO+2
Ø643    33Ø1         DCA RPATO+1
Ø644    2276         ISZ GENDEX+1     /INCREMENT LOWER
Ø645    724Ø         CLA CMA
Ø646    1277         TAD GENDEX+2     /DECREMENT UPPER ADDRESS
Ø647    3277         DCA GENDEX+2
Ø65Ø    2275         ISZ GENDEX       /DONE ALL
Ø651    5226         JMP GNPATR       /NO
```

```
Ø652  33Ø2        DCA RPATO+2       /CLR FOR COUNTING BITS
Ø653  13ØØ        TAD RPATO
Ø654  711Ø        CLL RAR
Ø655  743Ø        SZL               /BIT=1
Ø656  23Ø2        ISZ RPATO+2       /YES, COUNT
Ø657  744Ø        SZA               /DONE, FIRST WORD
Ø66Ø  5254        JMP .-4           /NO
Ø661  13Ø1        TAD RPATO+1       /GET 2ND

Ø662  711Ø        CLL RAR
Ø663  743Ø        SZL               /BIT=1
Ø664  23Ø2        ISZ RPATO+2       /YES
Ø665  744Ø        SZA               /DONE 2ND
Ø666  5262        JMP .-4           /NO
Ø667  13Ø2        TAD RPATO+2
Ø67Ø  43Ø3        JMS MCOMOB                /MAKE OBVERSE
Ø671  Ø127        AND K77ØØ

Ø672  13Ø2        TAD RPATO+2       /MAKE WORD 65
Ø673  3676        DCA I GENDEX+1
Ø674  56ØØ        JMP I PATGEN      /EXIT

Ø675  ØØØØ  GENDEX,     Ø                   /TO COUNT WORDS
Ø676  ØØØØ        Ø                 /TO STORE LOWER
Ø677  ØØØØ        Ø                 /TO STORE UPPER
Ø7ØØ  ØØØØ  RPATO,      Ø                   /PATTERN STORAGE
Ø7Ø1  ØØØØ        Ø
Ø7Ø2  ØØØØ        Ø                 /TO COUNT BITS
```

```
                /MAKE COMPLIMENT OBVERSE OF AC

Ø7Ø3   53Ø3    MCOMOB,     JMP .
Ø7Ø4   3355        DCA COMSTR        /SAVE ORIGINAL
Ø7Ø5   1355        TAD COMSTR
Ø7Ø6   Ø122        AND KØ7ØØ
Ø7Ø7   711Ø        CLL RAR
Ø71Ø   7Ø12        RTR
Ø711   3356        DCA COMSTR+1
Ø712   1355        TAD COMSTR

Ø713   Ø123        AND KØØ7Ø
Ø714   7ØØ6        RTL
Ø715   7ØØ4        RAL
Ø716   1356        TAD COMSTR+1
Ø717   3356        DCA COMSTR+1
Ø72Ø   1355        TAD COMSTR
Ø721   Ø12Ø        AND KØØØ7
Ø722   7Ø12        RTR
Ø723   7Ø12        RTR
Ø724   1356        TAD COMSTR+1
Ø725   3356        DCA COMSTR+1
Ø726   1355        TAD COMSTR
Ø727   Ø124        AND K7ØØØ
Ø73Ø   7ØØ6        RTL
Ø731   7ØØ6        RTL
Ø732   1356        TAD COMSTR+1
Ø733   7Ø4Ø        CMA
Ø734   57Ø3        JMP I MCOMOB

                /RANDOM NUMBER GENERATOR

Ø735   5335    GENRAN,     JMP .
Ø736   72ØØ        CLA
Ø737   1352        TAD RANVAR
Ø74Ø   71Ø4        CLL RAL
Ø741   743Ø        SZL
Ø742   7ØØ1        IAC
Ø743   3352        DCA RANVAR
Ø744   1352        TAD RANVAR
Ø745   1351        TAD RANNØ
Ø746   3351        DCA RANNØ
Ø747   1351        TAD RANNØ
Ø75Ø   5735        JMP I GENRAN     /EXIT AC=RANDOM

Ø751   2634    RANNØ,      2634
Ø752   4263    RANVAR,     4263
Ø753   7774    MIN4,       7774
Ø754   77Ø4    MIN6Ø,      77Ø4
Ø755   ØØØØ    COMSTR,     Ø
```

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

                *1ØØØ
1ØØØ    52ØØ    RANSEL,     JMP .
1ØØ1    4246       JMS SELGEN              /GET RANDOM NUMBER
1ØØ2    Ø12Ø       AND KØØØ7               /SELECT DRIVE
1ØØ3    3Ø45       DCA CDRIVE
1ØØ4    1Ø45       TAD CDRIVE
1ØØ5    711Ø       CLL RAR
1ØØ6    7Ø12       RTR
1ØØ7    7Ø1Ø       RAR

1Ø1Ø    3Ø46       DCA UNFUNC      /POSITION DRIVE NUMBER
1Ø11    1Ø45       TAD CDRIVE
1Ø12    7Ø4Ø       CMA
1Ø13    3Ø1Ø       DCA 1Ø          /MAKE NEG FOR COUNT
1Ø14    1121       TAD K4ØØØ
1Ø15    3Ø11       DCA 11
1Ø16    2Ø1Ø       ISZ 1Ø          /BIT IN DRIVE POSITION
1Ø17    741Ø       SKP

1Ø2Ø    5225       JMP EXIST               /BIT IS IN POSITION
1Ø21    1Ø11       TAD 11
1Ø22    711Ø       CLL RAR         /MOVE UNIT BIT
1Ø23    3Ø11       DCA 11
1Ø24    5216       JMP .-6
1Ø25    1Ø11    EXIST,     TAD 11          /GET UNIT BIT
1Ø26    ØØ47       AND MSBITS      /MASK WITH DRIVES SELECT
1Ø27    765Ø       SNA CLA         /DOES DRIVE EXIST

1Ø3Ø    52Ø1       JMP RANSEL+1    /NO, TRY AGAIN
1Ø31    1Ø45       TAD CDRIVE
1Ø32    1243       TAD POSTBL
1Ø33    3Ø51       DCA POSITN      /FORM POSITION POINTER
1Ø34    1Ø45       TAD CDRIVE
1Ø35    1244       TAD DIRTBL
1Ø36    3Ø52       DCA DIRECT      /FORM DIRECTION POINTER
1Ø37    1Ø45       TAD CDRIVE
1Ø4Ø    1245       TAD LSTTBL      /GENERATE LAST BLOCK
1Ø41    3Ø53       DCA LSTBLK      /WRITTEN POINTER
1Ø42    56ØØ       JMP I RANSEL
```

```
1Ø43  35Ø2  POSTBL,     PNTRS+1          /TO GET DRIVE POSITION
1Ø44  3513  DIRTBL,     PNTRS+12                 /DIRECTION
1Ø45  3524  LSTTBL,     PNTRS+23                 /LAST BLOCK WRITTEN
1Ø46  5246  SELGEN,     JMP .
1Ø47  1261      TAD SELVAR
1Ø5Ø  71Ø4      CLL RAL
1Ø51  743Ø      SZL
1Ø52  7ØØ1      IAC
1Ø53  3261      DCA SELVAR
1Ø54  1261      TAD SELVAR
1Ø55  1262      TAD SELVAR+1
1Ø56  3262      DCA SELVAR+1
1Ø57  1262      TAD SELVAR+1
1Ø6Ø  5646      JMP I SELGEN
1Ø61  4263  SELVAR,     4263
1Ø62  2634      2634

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

            *12ØØ
12ØØ  52ØØ  MVDATA,     JMP .
12Ø1  72ØØ      CLA
12Ø2  16ØØ      TAD I MVDATA
12Ø3  22ØØ      ISZ MVDATA
12Ø4  3222      DCA MOVDEX
12Ø5  16ØØ      TAD I MVDATA
12Ø6  22ØØ      ISZ MVDATA
12Ø7  3223      DCA MOVDEX+1

121Ø  16ØØ      TAD I MVDATA
1211  22ØØ      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
122Ø  5213      JMP .-5
1221  56ØØ      JMP I MVDATA

1222  ØØØØ  MOVDEX,     Ø
1223  ØØØØ      Ø
1224  ØØØØ      Ø
```

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

1225  5225  CODATA,     JMP .
1226  72ØØ      CLA
1227  1625      TAD I CODATA     /GET ADDRESS 1 CORRECT
123Ø  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  7Ø41      CMA IAC
1237  3257      DCA CODEX+2
124Ø  2225      ISZ CODATA
1241  7Ø4Ø      CMA
1242  3Ø72      DCA COCNTR       /SET NO ERROR FLAG
1243  1655  COLOOP,     TAD I CODEX
1244  7Ø41      CMA IAC
1245  1656      TAD I CODEX+1

1246  764Ø      SZA CLA          /WORDS =
1247  5261      JMP COERRO       /NO, TYPE OJT
125Ø  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  ØØØØ  CODEX,      Ø                /KNOWN DATA ADDRESS
1256  ØØØØ      Ø                /UNKNOWN DATA ADDRESS
1257  ØØØØ      Ø                /LENGTH
126Ø  1256      .-2

                  /DATA ERROR PRINTOUT

1261  2Ø72  COERRO,     ISZ COCNTR       /FIRST DATA ERROR
1262  5313      JMP COERR1       /NOT FIRST
1263  4467      JMS I ERRSTP
1264  6761      IOT 761
1265  Ø132      AND KØØ1Ø
1266  765Ø      SNA CLA
1267  5273      JMP .+4
127Ø  4672      JMS I .+2
1271  5274      JMP .+3
1272  4432      TYRALL
1273  4544      JMS I READTY     /FIRST ERROR
1274  45Ø7      JMS I TYPTEX     /TYPE HEADER
```

```
1275  7777        7777
1276  4441        4441              /TYPE
1277  6441        6441              /CDATA ERROR)
13ØØ  ØØ45        ØØ45

13Ø1  6262        6262
13Ø2  5762        5762
13Ø3  77ØØ        77ØØ
13Ø4  1Ø71        TAD RECRDK                  /TYPE THE OCTAL
13Ø5  45Ø6        JMS I TYPCON      /BLOCK NUMBER
13Ø6  45Ø7        JMS I TYPTEX
13Ø7  ØØ42        ØØ42
131Ø  5457        5457              /TYPE (BLOCK)

1311  4353        4353
1312  77ØØ        77ØØ
1313  76Ø4   COERR1,     CLA OSR
1314  Ø131        AND K1ØØØ
1315  765Ø        SNA CLA /ONLY TYPE4
1316  5323        JMP .+5 /TYPE ALL ERRORS
1317  1Ø72        TAD COCNTR
132Ø  1354        TAD K7774
1321  77ØØ        SMA CLA /DONE 4 DATA TYPEOUTS
1322  525Ø        JMP COLOOP+5      /YES, DELETE REST
1323  45Ø7        JMS I TYPTEX
1324  7777        7777              /BLANK LINE BETWEEN
1325  77ØØ        77ØØ              /EACH PAIR
1326  1255        TAD CODEX
1327  45Ø6        JMS I TYPCON      /TYPE CORRECT IN OCTAL
133Ø  45Ø7        JMS I TYPTEX

1331  ØØ43        ØØ43              /TYPE (COR)
1332  5762        5762
1333  77ØØ        77ØØ
1334  1256        TAD CODEX+1
1335  45Ø6        JMS I TYPCON      /TYPE INCOR OCTAL
1336  45Ø7        JMS I TYPTEX      /TYPE (INC)
1337  ØØ51        ØØ51
134Ø  5643        5643
1341  77ØØ        77ØØ
1342  126Ø        TAD CODEX+3
1343  45Ø6        JMS I TYPCON
1344  45Ø7        JMS I TYPTEX
1345  ØØ41        41
1346  4444        4444
1347  6263        6263
135Ø  ØØ51        51
1351  5643        5643
1352  77ØØ        77ØØ
1353  525Ø        JMP COLOOP+5
1354  7774   K7774,      7774
```

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

                *14ØØ
14ØØ    1Ø74    WRITE1,     TAD RBFLOC
14Ø1    3Ø1Ø        DCA 1Ø
14Ø2    1453        TAD I LSTBLK
14Ø3    341Ø        DCA I 1Ø                /FIRST BLOCK WRITTEN
14Ø4    1Ø77        TAD NUMBLK
14Ø5    7Ø41        CMA IAC
14Ø6    1453        TAD I LSTBLK    /LST BLOCK+1
14Ø7    3Ø4Ø        DCA RECORD
141Ø    1Ø4Ø        TAD RECORD
1411    1146        TAD K5Ø75
1412    77ØØ        SMA CLA
1413    5317        JMP REWCK
1414    1Ø4Ø        TAD RECORD

1415    341Ø        DCA I 1Ø
1416    4462        JMS I RANGEN
1417    341Ø        DCA I 1Ø                /FIRST RANDOM WORD
142Ø    4462        JMS I RANGEN
1421    341Ø        DCA I 1Ø                /2ND RANDOM WORD
1422    4462        JMS I RANGEN
1423    Ø125        AND KØØØ3
1424    765Ø        SNA CLA
1425    7Ø4Ø        CMA
1426    3Ø76        DCA DIRFLG
1427    1Ø76        TAD DIRFLG
143Ø    1Ø73        TAD RBUFST
1431    3Ø1Ø        DCA 1Ø

1432    1Ø1Ø        TAD 1Ø
1433    3Ø11        DCA 11          /MAKE FIRST WORD
1434    1121        TAD K4ØØØ               /INDICATE BACKWARD
1435    141Ø        TAD I 1Ø                /OR 2ND WORD
1436    3411        DCA I 11                INDICATE FORWARD
1437    4461        JMS I GENPAT    /GENERATE 129 WORD PATTERN
144Ø    1Ø77        TAD NUMBLK
1441    337Ø        DCA SAVNUM
1442    1Ø77    CORT1R,     TAD NUMBLK
1443    7Ø4Ø        CMA

1444    3Ø4Ø        DCA RECORD
1445    1Ø76        TAD DIRFLG
1446    764Ø        SZA CLA         /BACKWARD IS
1447    1Ø4Ø        TAD RECORD      /LAST BLOCK+NUM BLOCKS
145Ø    1453        TAD I LSTBLK
1451    3Ø4Ø        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  22ØØ          WDCMOD
1454  4455          JMS I SRCHIT

1455  115Ø      WRT1A,      TAD KØØ5Ø
1456  6764          IOT 764           /SRCH TO WRITE DATA
1457  1117          TAD K7577
146Ø  3443          DCA I WCLOC
1461  1074          TAD RBFLOC
1462  3444          DCA I CALOC
1463  4466          JMS I WAITI

1464  ØØØ1          1
1465  5326          JMP WSTERR
1466  1443          TAD I WCLOC
1467  764Ø          SZA CLA
147Ø  5326          JMP WSTERR
1471  1Ø76          TAD DIRFLG        /Ø OR -1
1472  71ØØ          CLL
1473  1Ø4Ø          TAD RECORD
1474  751Ø          SPA
1475  732Ø          CLA STL
1476  3Ø4Ø          DCA RECORD
1477  742Ø          SNL               /RECORD-1
15ØØ  2Ø4Ø          ISZ RECORD                  /NO +1

15Ø1  2Ø77          ISZ NUMBLK        /DONE ALL BLOCKS
15Ø2  5256          JMP WRT1A+1       /WRITE DATA 1 MORE
15Ø3  1Ø4Ø      WDINC,      TAD RECORD                  /NEW POSITION
15Ø4  3451          DCA I POSITN
15Ø5  1Ø76          TAD DIRFLG

15Ø6  3452          DCA I DIRECT      /DIRECTION FLAG
15Ø7  15Ø2          TAD I RBFWD2
151Ø  Ø143          AND K3777
1511  3453          DCA I LSTBLK
1512  1113          TAD KØ2ØØ
1513  6764          IOT 764           /STOP TAPE
1514  5715          JMP I .+1
1515  Ø221          MOFPRO            /RANDOM SELECT AGAIN

1516  Ø525          REWDRV
1517  4462      REWCK,      JMS I RANGEN    /GET RANDOM NUMBER
152Ø  771Ø          SPA CLA           /+ OR -
1521  5716          JMP I REWCK-1
1522  3Ø4Ø          DCA RECORD                  /+READ BLOCK Ø
1523  3Ø76          DCA DIRFLG                  /FORWARD
1524  5725          JMP I .+1
1525  Ø3Ø4          RDTAB
```

```
1526  4467  WSTERR,     JMS I ERRSTP
1527  4501        JMS I WRTDTY
153Ø  1Ø71        TAD RECRDK
1531  45Ø6        JMS I TYPCON
1532  45Ø7        JMS I TYPTEX
1533  ØØ42        42
1534  5457        5457
1535  4353        4353
1536  77ØØ        77ØØ
1537  137Ø        TAD SAVNUM
154Ø  3Ø77        DCA NUMBLK
1541  6772        IOT 772
1542  3372        DCA SAVNUM+2
1543  1371        TAD SAVNUM+1
1544  45Ø6        JMS I TYPCON
1545  45Ø7        JMS I TYPTEX
1546  ØØ63        ØØ63
1547  6441        6441
155Ø  64ØØ        64ØØ
1551  4277        4277
1552  1Ø43        TAD WCLOC
1553  45Ø6        JMS I TYPCON
1554  45Ø7        JMS I TYPTEX
1555  ØØ67        67
1556  1643        1643
1557  1677        1677
156Ø  76Ø4        LAS
1561  Ø154        AND K6ØØØ
1562  765Ø        SNA CLA
1563  74Ø2        HLT
1564  2151        ISZ PASFLG                /2ND ERROR
1565  5767        JMP I .+2                 /REWIND DRIVE
1566  5242        JMP CORT1R
1567  Ø525        REWDRV
157Ø  ØØØØ  SAVNUM,     Ø
1571  1572        .+1
```

```
                /READ ENTIRE SEQUENCE OF
                /BLOCKS AS ORIGINALLY WRITTEN

                *16ØØ
16ØØ  1473      RDSEQ,    TAD I RBUFST    /GET LOWEST BLOCK
16Ø1  Ø143          AND K3777
16Ø2  3Ø4Ø          DCA RECORD
16Ø3  15Ø2          TAD I RBFWD2    /GET LST BLOCK +1
16Ø4  Ø143          AND K3777
16Ø5  3Ø77          DCA NUMBLK      /MAKE NUMBER OF BLOCKS
16Ø6  1Ø4Ø          TAD RECORD
16Ø7  7Ø4Ø          CMA
161Ø  1Ø77          TAD NUMBLK
1611  3Ø77          DCA NUMBLK
1612  4462          JMS I RANGEN
1613  71ØØ          CLL
1614  771Ø          SPA CLA /READ FWD
1615  7Ø4Ø          CMA     /NO BKWD
1616  744Ø          SZA
1617  712Ø          STL     /L=1 IF BKWD
162Ø  3Ø76          DCA DIRFLG
1621  1Ø4Ø          TAD RECORD
1622  743Ø          SZL
1623  1Ø77          TAD NUMBLK
1624  3Ø4Ø          DCA RECORD
1625  1Ø77          TAD NUMBLK      /MAKE NUMBER OF BLKS-
1626  7Ø4Ø          CMA

1627  3Ø77          DCA NUMBLK              /MAKE-
163Ø  4463          JMS I READ1     /READ FIRST BLOCK
1631  55Ø4          JMP I READ2     /STATUS ERROR ON READ
1632  2Ø77
```

```
              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 BUFFR 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 BUFFR
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  745Ø        SNA               /LAST BLOCK + OR-
1713  7ØØ1        IAC               /1
1714  1Ø4Ø        TAD RECORD
1715  3Ø4Ø        DCA RECORD
1716  4466        JMS I WAITI       /WAIT TO FINISH READ
1717  ØØØ1        1
172Ø  55Ø4        JMP I READ2       /STATUS ERROR, REREAD
1721  1443        TAD I WCLOC
1722  764Ø        SZA CLA
1723  55Ø4        JMP I READ2
1724  5232        JMP RDBUF1                /COMPARE FIRST BUFFR

1725  1Ø4Ø  NDOFRD,     TAD RECORD
1726  3451        DCA I POSITN      /NEW POSITION
1727  1Ø76        TAD DIRFLG
173Ø  3452        DCA I DIRECT      /INDICATE DIRECTION
1731  5732        JMP I .+1
1732  Ø221        MOFPRO

            PAUSE

            /TCØ1 DECTREX 1  -  TAPE 2
            /READ DATA CONTINUOUS MODE
            /TWO BLOCKS AND COMPARE DATA READ

            *2ØØØ
2ØØØ  1Ø76  RDCMOD,     TAD DIRFLG
2ØØ1  764Ø        SZA CLA           /FORWARDS
2ØØ2  5212        JMP .+1Ø                  /NO
2ØØ3  1Ø4Ø        TAD RECORD
2ØØ4  7Ø4Ø        CMA
2ØØ5  1453        TAD I LSTBLK      /BLOCK+1 WRITTEN
2ØØ6  744Ø        SZA
2ØØ7  5215        JMP .+6
2Ø1Ø  5611        JMP I .+1                 /BLOCK NOT WRITTEN
2Ø11  Ø3Ø4        RDTAB             /READ ONLY 1
2Ø12  1Ø4Ø        TAD RECORD
2Ø13  765Ø        SNA CLA           /BLOCK Ø BACKWARDS
2Ø14  5611        JMP I .-3                 /READ ONLY 1

2Ø15  4455        JMS I SRCHIT      /FIND FIRST BLOCK
2Ø16  1313        TAD KØ13Ø                 /READ DATA
2Ø17  6764        IOT 764           /CONTINUOUS MODE
2Ø2Ø  1117        TAD K7577
2Ø21  1117        TAD K7577
2Ø22  3443        DCA I WCLOC
2Ø23  1Ø75        TAD BF1LOC
2Ø24  3444        DCA I CALOC
2Ø25  4466        JMS I WAITI
2Ø26  ØØØ1        1
2Ø27  5275        JMP RDCERR        /SEE WHICH BLOCK IN ERROR
2Ø3Ø  1443        TAD I WCLOC
2Ø31  764Ø        SZA CLA
2Ø32  55Ø4        JMP I READ2
2Ø33  1113        TAD KØ2ØØ
2Ø34  6764        IOT 764           /STOP TAPE
2Ø35  1Ø76        TAD DIRFLG
2Ø36  745Ø        SNA
2Ø37  7ØØ1        IAC
```

```
2040  1040       TAD RECORD
2041  3040       DCA RECORD
2042  4464       JMS I REGENP      /VERIFY LAS1 3LOCK
2043  4465       JMS I DATACO      /FIRST
2044  6774       RANBFR
2045  7376       BUFFRS+201
2046  0201       0201
2047  2072       ISZ COCNTR        /ANY DATA E?R?RS
2050  5505       JMP I READ2A      /YES REREAD
2051  7040
```

```
                     CMA
2Ø52  11Ø3           TAD BF2LOC
2Ø53  3444           DCA I CALOC

2Ø54  4464           JMS I REGENP
2Ø55  1Ø76           TAD DIRFLG
2Ø56  745Ø           SNA
2Ø57  7ØØ1           IAC
2Ø6Ø  7Ø41           CMA IAC
2Ø61  1Ø4Ø           TAD RECORD
2Ø62  3Ø4Ø           DCA RECORD         /VERIFIFY FIRST
2Ø63  4465           JMS I DATACO       /BLOCK READ

2Ø64  6774           RANBFR
2Ø65  7175           BUFFRS.
2Ø66  Ø2Ø1           Ø2Ø1
2Ø67  2Ø72           ISZ COCNTR         /ANY DATA ERRORS
2Ø7Ø  55Ø5           JMP I READ2A       /YES, REREAD
2Ø71  1Ø4Ø           TAD RECORD
2Ø72  3451           DCA I POSITN
2Ø73  5674           JMP I .+1
2Ø74  Ø221           MOFPRO

2Ø75  1443     RDCERR,     TAD I WCLOC
2Ø76  745Ø           SNA                /ERROR IN 2ND BLOCK
2Ø77  53Ø5           JMP .+6            /YES COUNT BLOCK
21ØØ  7Ø41           CMA IAC
21Ø1  1117           TAD K7577                 /WC-129
21Ø2  3443           DCA I WCLOC
21Ø3  57Ø4           JMP I .+1                 /TYPE STATUS ERROR
21Ø4  Ø42Ø           REREAD

21Ø5  1Ø76           TAD DIRFLG
21Ø6  745Ø           SNA                /BACKWARDS-1
21Ø7  7ØØ1           IAC                /FORWARD+1
211Ø  1Ø4Ø           TAD RECORD
2111  3Ø4Ø           DCA RECORD
2112  57Ø4           JMP I .-6                 /TYPE STATUS ERROR

2113  Ø13Ø     KØ13Ø,      13Ø
```

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

                *22ØØ

22ØØ  3277      WDCMOD,    DCA TRECTR
22Ø1  1Ø77         TAD NUMBLK
22Ø2  1125         TAD KØØØ3
22Ø3  745Ø         SNA             /AN INC OF 3
22Ø4  5212         JMP SWCMOD      /YES WRITE C MODE
22Ø5  75ØØ         SMA             /GONE PAST Ø
22Ø6  5611         JMP I .+3              /NOT INC OF 3
22Ø7  2277         ISZ TRECTR
221Ø  52Ø2         JMP WDCMOD+2
2211  1454         WRT1A-1

2212  1277      SWCMOD,    TAD TRECTR     /MAKE NUMBER
2213  7Ø4Ø         CMA             /OF GROUPS OF 3
2214  3277         DCA TRECTR             /2'S COMPLIMENT
2215  447Ø         JMS I DATAMV    /MAKE PATTERN
2216  6774         RANBFR          /3 BUFFERS LONG
2217  7175         BUFFRS
222Ø  7376         7376
2221  4455         JMS I SRCHIT    /FIND FIRST BLOCK
2222  13Ø1         TAD KØ15Ø              /WRITE DATA CONTINUOUSLY
2223  6764      SWCMDL,    IOT 764
2224  13ØØ         TAD K7175              /-129  3 TIMES
2225  3443         DCA I WCLOC
2226  1Ø74         TAD RBFLOC
2227  3444         DCA I CALOC
223Ø  4466         JMS I WAITI
2231  ØØØ1         1
2232  5253         JMP WDCERR      /NOT NORMAL INTERRUPT
2233  1443         TAD I WCLOC
2234  764Ø         SZA CLA
2235  5253         JMP WDCERR             /WC NOT ZERO
2236  1Ø76         TAD DIRFLG
2237  71ØØ         CLL
224Ø  764Ø         SZA CLA
2241  712Ø         STL
2242  1125         TAD KØØØ3
2243  743Ø         SZL             /BACKWARDS IS -3
2244  7Ø41         CMA IAC
2245  1Ø4Ø         TAD RECORD
2246  3Ø4Ø         DCA RECORD
2247  2277         ISZ TRECTR      /DONE ALL
225Ø  5223         JMP SWCMDL      /NO, DO NEXT 3 BLOCKS
2251  5652         JMP I .+1
2252  15Ø3         WDINC
```

```
2253  3277  WDCERR,     DCA TRECTR       /FIND WHICH
2254  1443        TAD I WCLOC      /BLOCK WAS IN
2255  7Ø41        CMA IAC          /ERROR
2256  1117        TAD K7577
2257  751Ø        SPA
226Ø  5263        JMP .+3
2261  2277        ISZ TRECTR
2262  5256        JMP .-4
2263  72ØØ        CLA
2264  1Ø76        TAD DIRFLG
2265  71ØØ        CLL
2266  764Ø        SZA CLA          /FORWARD
2267  712Ø        STL              /BACKWARD MAKE-1 OR -2

227Ø  1277        TAD TRECTR
2271  743Ø        SZL
2272  7Ø41        CMA IAC
2273  1Ø4Ø        TAD RECORD
2274  3Ø4Ø        DCA RECORD       /BLOCK IN ERROR
2275  5676        JMP I .+1                  /TYPEOUT WRITE ERROR
2276  1526        WSTERR

2277  ØØØØ  TRECTR,     Ø
23ØØ  7175  K7175,      7175
23Ø1  Ø15Ø  KØ15Ø,      15Ø
```

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

                *24ØØ

24ØØ    1Ø76    RALLTS,     TAD DIRFLG
24Ø1    745Ø        SNA             /BACKWARDS
24Ø2    7ØØ1        IAC             /FORWARDS
24Ø3    7Ø41        CMA IAC         /+1 FOR BACK-1 FOR FWD
24Ø4    1Ø4Ø        TAD RECORD
24Ø5    3Ø4Ø        DCA RECORD
24Ø6    1Ø4Ø        TAD RECORD
24Ø7    751Ø        SPA
241Ø    5214        JMP .+4
2411    1147        TAD K5Ø76
2412    771Ø        SPA CLA
2413    5217        JMP .+4
2414    72ØØ        CLA
2415    5616        JMP I .+1               /CANT DO Ø OR 27Ø1
2416    Ø235        MOFPR1

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

2417    3377        DCA RAEFLG              /CLEAR READ ALL ERR. FL.
242Ø    4455        JMS I SRCHIT
2421    6764        IOT 764
2422    1Ø76        TAD DIRFLG
2423    745Ø        SNA
2424    7ØØ1        IAC
2425    1Ø4Ø        TAD RECORD
2426    3Ø4Ø        DCA RECORD
2427    6773        IOT 773
243Ø    5227        JMP .-1
2431    771Ø        SPA CLA
2432    5371        JMP RASERR
2433    1Ø41        TAD BLKFND
2434    7Ø41        CMA IAC
2435    1Ø4Ø        TAD RECORD
2436    764Ø        SZA CLA
2437    5371        JMP RASERR
```

```
                    /CHANGE TO READ ALL CONTINUOUS
2440   1374             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=ØØ
2456  1376              TAD RADLOC
2457  3Ø1Ø              DCA 1Ø
246Ø  1117              TAD K7577
2461  3Ø11              DCA 11
2462  3364              DCA CKSUMR        /CLR ACCUMU_ATED CKSUM
2463  141Ø              TAD I 1Ø
2464  Ø134              AND KØØ77
2465  3363              DCA REVCHK        /SAVE REVERSE CKSUM
2466  141Ø              TAD I 1Ø
2467  4331              JMS XORSUM        /GENERATE DATA SUM
247Ø  2Ø11              ISZ 11
2471  5266              JMP .-3
2472  1364              TAD CKSUMR
2473  7Ø4Ø              CMA
2474  Ø134              AND KØØ77
2475  3361              DCA DATASM        /SAVE DATA SUM
2476  141Ø              TAD I 1Ø
2477  Ø127              AND K77ØØ
25ØØ  3362              DCA FWDCHK        /SAVE FORWARD CHECKSUM
25Ø1  1361              TAD DATASM
25Ø2  3364              DCA CKSUMR
25Ø3  1362              TAD FWDCHK
25Ø4  4331              JMS XORSUM        /DATA SUM + FJRWARD CKECK
25Ø5  1363              TAD REVCHK
25Ø6  4331              JMS XORSUM        /MAKE FINAL SJM
25Ø7  1364              TAD CKSUMR
251Ø  764Ø              SZA CLA           /SUM OK
2511  576Ø              JMP I CKSERR      /NO, ERROR
2512  1Ø4Ø       RADCHK,      TAD RECORD
2513  3451              DCA I POSITN
2514  7Ø4Ø              CMA
2515  1444              TAD I CALOC
2516  3444              DCA I CALOC
2517  4464              JMS I REGENP
252Ø  4465              JMS I DATACO
2521  6774              RANBFR
2522  72Ø2              BUFFRS+5
2523  Ø2Ø1              Ø2Ø1
2524  2377              ISZ RAEFLG
2525  2Ø72              ISZ COCNTR
2526  55Ø5              JMP I READ2A
2527  573Ø              JMP I .+1
253Ø  Ø217              MOFPRO-2
```

```
            /FORM 6 BIT XOR OF AC AND CKSUMR
            /SAVE RESULTS IN CKSUMR
2531  5331  XORSUM,     JMP .
2532  7Ø4Ø      CMA
2533  3365      DCA XORSAV
2534  7Ø4Ø      CMA
2535  3366      DCA XORSAV+1    /TO COUNT PASSES
2536  1365      TAD XORSAV
2537  7Ø4Ø      CMA
254Ø  Ø364      AND CKSUMR
2541  3367      DCA XORSAV+2    /MAKE PARTIAL SUM
2542  1364      TAD CKSUMR
2543  7Ø4Ø      CMA
2544  Ø365      AND XORSAV      /OTHER HALF SJM
2545  1367      TAD XORSAV+2    /COMBINE
2546  Ø134      AND KØØ77               /CLR TO LWR 6
2547  3364      DCA CKSUMR
255Ø  2366      ISZ XORSAV+1    /DONE UPPER 6?
2551  5731      JMP I XORSUM    /DONE ALL EXIT
2552  1365      TAD XORSAV
2553  7Ø12      RTR             /MOVE OVER 6
2554  7Ø12      RTR
2555  7Ø12      RTR
2556  3365      DCA XORSAV
2557  5336      JMP XORSUM+5    /XOR UPPER

256Ø  26ØØ  CKSERR,     ERRCKS
2561  ØØØØ  DATASM,     Ø
2562  ØØØØ  FWDCHK,     Ø
2563  ØØØØ  REVCHK,     Ø
2564  ØØØØ  CKSUMR,     Ø
2565  ØØØØ  XORSAV,     Ø
2566  ØØØØ      Ø
2567  ØØØØ      Ø
257Ø  27Ø7  RARERR,     ERRRAR
2571  3151  RASERR,     DCA PASFLG
2572  5773      JMP I .+1
2573  4ØØØ      SRHERR
2574  Ø12Ø  KØ12Ø,      Ø12Ø
2575  7571  K7571,      7571
2576  72ØØ  RADLOC,     BUFFRS+3
2577  ØØØØ  RAEFLG,     Ø
```

```
                /CHECK SUM ERROR TYPEOUT

                *26ØØ

26ØØ  47Ø1  ERRCKS,     JMS I CSECON     /TYRA_L
26Ø1  1Ø71      TAD RECRDK
26Ø2  45Ø6      JMS I TYPCON
26Ø3  45Ø7      JMS I TYPTEX
26Ø4  ØØ42      42
26Ø5  5457      5457
26Ø6  4353      4353
26Ø7  7777      7777
261Ø  4353      4353
2611  6365      6365
2612  55ØØ      55ØØ
2613  4562      4562
2614  6257      6257
2615  6277      6277
2616  13Ø2      TAD CSECON+1     /REVCHK
2617  45Ø6      JMS I TYPCON
262Ø  45Ø7      JMS I TYPTEX
2621  ØØ62      62
2622  4566      4566
2623  77ØØ      77ØØ
2624  4257      JMS CKSTYP       /TYPE CHK SUM
2625  13Ø3      TAD CSECON+2     /DATASM
2626  45Ø6      JMS I TYPCON
2627  47Ø4      JMS I CSECON+3   /TYDATA
263Ø  4257      JMS CKSTYP       /CHECKSUM
2631  427Ø      JMS CALCTY
2632  13Ø5      TAD CSECON+4     /FWD CHECK
2633  45Ø6      JMS I TYPCON
2634  4257      JMS CKSTYP       /CHK SUM
2635  17Ø6      TAD I CSECON+5
2636  7Ø4Ø      CMA
2637  Ø134      AND KØØ77
264Ø  37Ø6      DCA I CSECON+5
2641  13Ø6      TAD CSECON+5     /CKSUMR
2642  45Ø6      JMS I TYPCON
2643  427Ø      JMS CALCTY
2644  45Ø7      JMS I TYPTEX
2645  ØØ54      54               /L
2646  166Ø      166Ø             /,P
2647  1642      1642             /,B
265Ø  1677      1677             /,
2651  7Ø4Ø      CMA
2652  3655      DCA I .+3
2653  5654      JMP I .+1
2654  2512      RADCHK
2655  2577      RAEFLG
2656  Ø525      REWDRV
```

```
2657  5257  CKSTYP,     JMP .
2660  4507        JMS I TYPTEX
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 TYPTEX
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
271Ø   47Ø1         JMS I CSECON
2711   1Ø71         TAD RECRDK
2712   45Ø6         JMS I TYPCON
2713   45Ø7         JMS I TYPTEX
2714   ØØ42         42
2715   5457         5457
2716   4353         4353
2717   77ØØ         77ØØ
272Ø   6772         IOT 772
2721   3345         DCA .+24
2722   1344         TAD .+22
2723   45Ø6         JMS I TYPCON
2724   45Ø7         JMS I TYPTEX
2725   ØØ63         63
2726   6441         6441
2727   64ØØ         64ØØ
273Ø   4277         4277
2731   1Ø43         TAD WCLOC
2732   45Ø6         JMS I TYPCON
2733   45Ø7         JMS I TYPTEX
2734   ØØ67         67
2735   1643         1643
2736   1677         1677
2737   76Ø4         LAS
274Ø   Ø154         AND K6ØØØ
2741   765Ø         SNA CLA
2742   74Ø2         HLT
2743   5656         JMP I CKSTYP-1
2744   2745         .+1
2745   ØØØØ         Ø
```

```
             /TC01 RANDOM EXERCISER
             /TYPEOUTS AND SOME CONTROL ROUTINES

             *3000

             /WAIT FOR INTERRUPT ROUTINE
             /SYNCHRONIZE PROCESSOR TESTS

3000  5200   WATINT,     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 PIDEX1
3025  2343       ISZ PIDEX1
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

3Ø34  3327  IRECD,     DCA ACSAVE              /SAVE ACCUMULATOR
3Ø35  7ØØ4      RAL
3Ø36  3326      DCA LKSAVE             /AND LINK FOR NEXT WAIT
3Ø37  6771      IOT 771
3Ø4Ø  5255      JMP NDTSKP             /NO DECTAPE FLAG SET
3Ø41  6772      IOT 772
3Ø42  333Ø      DCA SBRECD             /SAVE STATUS B
3Ø43  16ØØ      TAD I WATINT
3Ø44  7Ø4Ø      CMA             /GET STAT B COMPARE CON
3Ø45  Ø33Ø      AND SBRECD
3Ø46  765Ø      SNA CLA         /ANY ILLEGAL FLAGS
3Ø47  22ØØ      ISZ WATINT             /NO IS SKIP EXIT TWICE
3Ø5Ø  22ØØ      ISZ WATINT             /ONLY 1 ISZ IF NOT I EXPECTED
3Ø51  12ØØ      TAD WATINT
3Ø52  3744      DCA I PIDEX2
3Ø53  2344      ISZ PIDEX2
3Ø54  56ØØ      JMP I WATINT

3Ø55  4467  NDTSKP,    JMS I ERRSTP    /STOP TAPE
3Ø56  45Ø7      JMS I TYPTEX
3Ø57  7777      7777
3Ø6Ø  6Ø16      6Ø16
3Ø61  5116      5116
3Ø62  ØØ77      ØØ77
3Ø63  53Ø4      JMP PTSTNS

3Ø64  3Ø65  PTSTND,     .+1
3Ø65  2325      ISZ WTCNTR                    /WAITED LONG ENOUGH
3Ø66  5664      JMP I PTSTND    /NO DO NEXT TEST
3Ø67  6ØØ2      IOF
3Ø7Ø  6771      IOT 771         /ANY FLAGS SET
3Ø71  7Ø4Ø      CMA             /NO
3Ø72  333Ø      DCA SBRECD
3Ø73  4467      JMS I ERRSTP    /STOP TAPE
3Ø74  45Ø7      JMS I TYPTEX
3Ø75  7777      7777            /NO PI
3Ø76  5657      5657
3Ø77  ØØ6Ø      6Ø
31ØØ  1651      1651
31Ø1  1677      1677
31Ø2  233Ø      ISZ SBRECD
31Ø3  53Ø7      JMP .+4         /DECTAPE DID SKIP
```

```
3104  4507  PTSTNS,     JMS I TYPTEX
3105  0056        56
3106  5777        5777
3107  4507        JMS I TYPTEX
3110  ØØ44        44
3111  4543        4543
3112  6441        6441
3113  6Ø45        6Ø45
3114  ØØ63        63
3115  5351        5351
3116  6Ø77        6Ø77
3117  22ØØ        ISZ WATINT
312Ø  7Ø4Ø        CMA
3121  3162        DCA FRSWAT
3122  56ØØ        JMP I WATINT

3123  7655  WTKON1,     7655                /WAIT COUNT 5 SECONDS
3124  6422  WTKON2,     6422                /WAIT COUNT 45 SECONDS
3125  ØØØØ  WTCNTR,     Ø
3126  ØØØØ  LKSAVE,     Ø
3127  ØØØØ  ACSAVE,     Ø
313Ø  ØØØØ  SBRECD,     Ø

3131  5331  SETIDX,     JMP .
3132  1341        TAD K7772Y
3133  3342        DCA PICNTR
3134  1345        TAD PITBL
3135  3343        DCA PIDEX1
3136  1354        TAD PIRTBL
3137  3344        DCA PIDEX2
314Ø  5731        JMP I SETIDX

3141  7772  K7772Y,     7772
3142  ØØØØ  PICNTR,     Ø
3143  ØØØØ  PIDEX1,     Ø
3144  ØØØØ  PIDEX2,     Ø
3145  3146  PITBL,      .+1

            *PITBL+7
3154  3155  PIRTBL,     .+1
```

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

                *32ØØ

                /ISZ TEST ABOUT 61 MILLISECONDS
32ØØ   7Ø4Ø     ISZTST,    CMA
32Ø1   3345        DCA TEMP5
32Ø2   3342        DCA TEMP2
32Ø3   3341        DCA TEMP1
32Ø4   2341        ISZ TEMP1
32Ø5   2342        ISZ TEMP2
32Ø6   52Ø4        JMP .-2
32Ø7   1342        TAD TEMP2
321Ø   744Ø        SZA
3211   74Ø2        HLT
3212   724Ø        CLA CMA
3213   1341        TAD TEMP1
3214   744Ø        SZA
3215   74Ø2        HLT
3216   2345        ISZ TEMP5
3217   741Ø        SKP
322Ø   52Ø3        JMP ISZTST+3
3221   475Ø        JMS I NDPTST

                /ROTATE 1 TEST ABOUT 67 MILLISECONDS

3222   1342     ROT1TS,    TAD TEMP2
3223   713Ø        STL RAR
3224   7ØØ4        RAL
3225   742Ø        SNL
3226   74Ø2        HLT
3227   7Ø41        CMA IAC
323Ø   1342        TAD TEMP2
3231   744Ø        SZA
3232   74Ø2        HLT
3233   2342        ISZ TEMP2
3234   5222        JMP ROT1TS
3235   475Ø        JMS I NDPTST
```

```
                /ROTATE 2 TEST ALSO ABOUT 67 MI_LISECONDS

3236  1342      ROT2TS,     TAD TEMP2
3237  71Ø6          CLL RTL
324Ø  7Ø12          RTR
3241  743Ø          SZL
3242  74Ø2          HLT
3243  7Ø41          CMA IAC
3244  1342          TAD TEMP2
3245  744Ø          SZA
3246  74Ø2          HLT
3247  2342          ISZ TEMP2
325Ø  5236          JMP ROT2TS
3251  475Ø          JMS I NDPTST

                /TAD TEST ADD EVERY COM TO RAN NO
                /ABOUT 86 MILLISECONDS
3252  3343      TADTST,     DCA TEMP3
3253  1346          TAD PRAN1
3254  71Ø4          CLL RAL
3255  743Ø          SZL
3256  7ØØ1          IAC
3257  3346          DCA PRAN1
326Ø  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  7Ø41          CMA IAC
327Ø  1344          TAD TEMP4
3271  744Ø          SZA
3272  74Ø2          HLT
3273  2344          ISZ TEMP4
3274  7ØØØ          NOP
3275  2343          ISZ TEMP3
3276  5265          JMP .-11
3277  475Ø          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 K76ØØX
3303  3342           DCA TEMP2
3304  1351           TAD JMSLOC
3305  3343           DCA TEMP3
3306  1352           TAD JMSKON
3307  3344           DCA TEMP4
331Ø  1344           TAD TEMP4                 /STORE 128 JMS .
3311  3743           DCA I TEMP3        /STARTING AT ADDRESS
3312  2344           ISZ TEMP4                 /6ØØØ
3313  2343           ISZ TEMP3
3314  2342           ISZ TEMP2
3315  531Ø           JMP .-5
3316  1355           TAD RETJMP                /STORE JMP I RETUJM
3317  3743           DCA I TEMP3        /TO RETURN FROM JMS
332Ø  5751           JMP I JMSLOC       /EXECUTE 128 JMS

3321  1353      JMRETU,      TAD K76ØØX               /RETURN FROM EXECUTE
3322  3342           DCA TEMP2
3323  1351           TAD JMSLOC
3324  3343           DCA TEMP3                 /COMPARE ADDRESSES
3325  1343           TAD TEMP3                 /FOR .+1
3326  7Ø4Ø           CMA
3327  1743           TAD I TEMP3
333Ø  744Ø           SZA
3331  74Ø2           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  53Ø2           JMP JMSTST+2
3337  475Ø           JMS I NDPTST
334Ø  52ØØ           JMP ISZTST

3341  ØØØØ      TEMP1,       Ø
3342  ØØØØ      TEMP2,       Ø
3343  ØØØØ      TEMP3,       Ø
3344  ØØØØ      TEMP4,       Ø
3345  ØØØØ      TEMP5,       Ø
3346  4263      PRAN1,       4263
3347  2634      PRAN2,       2634

335Ø  3Ø64      NDPTST,      PTSTND
3351  6ØØØ      JMSLOC,      6ØØØ
3352  42ØØ      JMSKON,      42ØØ              /JMS + 2ØØ
3353  76ØØ      K76ØØX,      76ØØ              /FOR COUNTING JMS
3354  7763      K7763X,      7763
3355  5561      RETJMP,      JMP I RETUJM      /(RETUJM) ARE JMRETU
```

```
                PAUSE

                /TCØ1 DECTEX 1  - TAPE 3
                /REWIND ALL DRIVES SELECTED
                /TO END ZONE AT START OF TAPE

                *34ØØ
34ØØ   52ØØ     REPOSI,     JMP .
34Ø1   4217         JMS RSFDRV      /RESET POINTRS TO FIRST DRIVE
34Ø2   114Ø         TAD KØ6Ø4               /MOVE BACKWARDS
34Ø3   1Ø46         TAD UNFUNC              /+POSITIONED UNIT NO
34Ø4   6766         IOT 766         /SET STATUS A
34Ø5   4466         JMS I WAITI
34Ø6   5ØØØ         5ØØØ            /INDICATE EXPECT END
34Ø7   534Ø         JMP MOVER               /NOT STATUS EXPECTED

341Ø   724Ø         CLA CMA
3411   3451         DCA I POSITN    /INDICATE END ZONE
3412   724Ø         CLA CMA
3413   3452         DCA I DIRECT    /INDICATE BACKWARDS
3414   4237         JMS CHNGDR      /SET UP NEXT DRIVE
3415   52Ø2         JMP REPOSI+2    /REWIND NEXT DRIVE
3416   56ØØ         JMP I REPOSI    /GOT ALL DRIVES, EXIT


                /RESET CURRENT DRIVE POINTERS TO
                /FIRST DRIVE SELECTED
3417   5217     RSFDRV,     JMP .
342Ø   72ØØ         CLA
3421   3Ø45         DCA CDRIVE      /SET INITIALLY TO Ø
3422   1121         TAD K4ØØØ
3423   3Ø5Ø         DCA COMBIT
3424   1Ø47         TAD MSBITS

3425   ØØ5Ø         AND COMBIT
3426   764Ø         SZA CLA         /THIS DRIVE SELECTED
3427   5235         JMP RSFDR1      /YES, SET POINTER
343Ø   1Ø5Ø         TAD COMBIT
3431   711Ø         CLL RAR
3432   3Ø5Ø         DCA COMBIT              /MOVE COMPARE BIT
3433   2Ø45         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 .
344Ø  72ØØ          CLA
3441  1Ø5Ø          TAD COMBIT        /GET DRIVE COMPARE BIT
3442  711Ø          CLL RAR           /MOVE IT TO NEXT
3443  Ø3ØØ          AND NBIT8
3444  744Ø          SZA               /LAST DRIVE NUM 7
3445  5251          JMP .+4           /NO
3446  4217          JMS RSFDRV        /RESET TO FIRST

3447  2237          ISZ CHNGDR        /INCR. EXIT, SKIP
345Ø  5637          JMP I CHNGDR      /EXIT
3451  3Ø5Ø          DCA COMBIT
3452  2Ø45          ISZ CDRIVE
3453  1Ø5Ø          TAD COMBIT
3454  ØØ47          AND MSBITS
3455  765Ø          SNA CLA           /THIS DRIVE SELECTED
3456  524Ø          JMP CHNGDR+1      /NO

3457  4261          JMS GNPTRS        /GENERATE DRIVE POINTERS
346Ø  5637          JMP I CHNGDR

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

3461  5261      GNPTRS,     JMP .
3462  1Ø45          TAD CDRIVE        /DRIVE NUMBER
3463  7112          CLL RTR
3464  7Ø12          RTR               /POSITION TO BITS Ø,1,2
3465  3Ø46          DCA UNFUNC
3466  1Ø45          TAD CDRIVE        /DRIVE NUMBER
3467  13Ø1          TAD PNTRS         /+ POS. PNTR ADDRS,
347Ø  3Ø51          DCA POSITN        /FOR INDIRECTS

3471  1Ø45          TAD CDRIVE
3472  1312          TAD PNTRS+11      /+ DIRECTION PNTR
3473  3Ø52          DCA DIRECT        /FOR INDIRECTS
3474  1Ø45          TAD CDRIVE
3475  1323          TAD PNTRS+22
3476  3Ø53          DCA LSTBLK
3477  5661          JMP I GNPTRS
35ØØ  7767      NBIT8,      7767
```

```
3501  3502  PNTRS,      .+1         /TO GET LAST RECORD NUMBER

3502  ØØØØ         Ø             /FOR DRIVE 8
3503  ØØØØ         Ø             /1
3504  ØØØØ         Ø             /2
3505  ØØØØ         Ø             /3
3506  ØØØØ         Ø             /4
3507  ØØØØ         Ø             /5
351Ø  ØØØØ         Ø             /6
3511  ØØØØ         Ø             /7

3512  3513         .+1           /TO GET LAST DIRECTION
3513  ØØØØ         Ø             /DIRECTION - UNIT 8
3514  ØØØØ         Ø             /1
3515  ØØØØ         Ø             /2
3516  ØØØØ         Ø             /3
3517  ØØØØ         Ø             /4
352Ø  ØØØØ         Ø             /5
3521  ØØØØ         Ø             /6
3522  ØØØØ         Ø             /7
3523  3524         .+1
3524  ØØØØ         Ø             /TO GET LAST BLOCK WRITTEN
3525  ØØØØ         Ø             /DRIVE 8
3526  ØØØØ         Ø             /1
3527  ØØØØ         Ø             /2
353Ø  ØØØØ         Ø             /3
3531  ØØØØ         Ø             /4
3532  ØØØØ         Ø             /5
3533  ØØØØ         Ø             /6
3534  ØØØØ         Ø             /7
3535  3536         .+1
3536  ØØØØ         Ø
3537  44ØØ         TYMOVE
354Ø  4342  MOVER,        JMS .+2
3541  52Ø2         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
355Ø  45Ø6         JMS I TYPCON
3551  45Ø7         JMS I TYPTEX
3552  ØØ63         63
3553  6441         6441
3554  64ØØ         64ØØ
3555  4277         4277
3556  76Ø4         LAS
3557  Ø154         AND K6ØØØ
356Ø  765Ø         SNA CLA
3561  74Ø2         HLT
3562  5742         JMP I MOVER+2
```

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

                   *36ØØ
36ØØ   52ØØ        SEARCH,     JMP .
36Ø1   73ØØ            CLL CLA
36Ø2   1Ø76            TAD DIRFLG
36Ø3   764Ø            SZA CLA             /FORWARD
36Ø4   712Ø            STL                 /NO, BACKWARD
36Ø5   1125            TAD KØØØ3                   /BACKWARD TA
36Ø6   742Ø            SNL                 /IS BLOCK+3
36Ø7   7Ø41            CMA IAC             /FWD IS BLOCK-3
361Ø   1Ø4Ø            TAD RECORD
3611   7Ø41            CMA IAC
3612   3156            DCA TAPONT          /INDICATE TA POINT
3613   1Ø76            TAD DIRFLG
3614   765Ø            SNA CLA             /FORWARD IS
3615   1137            TAD KØ4ØØ                   /START BKWD
3616   1126            TAD KØ214
3617   1Ø46            TAD UNFUNC

362Ø   6766            IOT 766
3621   1Ø42            TAD IDCON
3622   3444            DCA I CALOC
3623   7Ø4Ø            CMA
3624   3157            DCA BLKFLG
3625   4466            JMS I WAITI
3626   ØØØ1            1
3627   5326            JMP SREZTS -        /SEE IF EZ FOUND
363Ø   4263            JMS SRCONT
3631   5235            JMP SRTARN          /BLOCK=BLOCKFND

3632   5235            JMP SRTARN          /GONE PAST BLCK
3633   6764            IOT 764             /HAVEN'T REACHED BLOCK
3634   5225            JMP .-7             /FOR TURN AROUND YET
```

```
3635  6761  SRTARN,     IOT 761
3636  7Ø4Ø      CMA
3637  Ø113      AND KØ2ØØ               /IN CASE MOTION=Ø
3640  1137      TAD KØ4ØØ
3641  6764      IOT 764         /DO TURN AROUND
3642  1Ø4Ø      TAD RECORD
3643  7Ø41      CMA IAC
3644  3156      DCA TAPONT
3645  7Ø4Ø      CMA
3646  3157      DCA BLKFLG
3647  4466      JMS I WAITI
365Ø  ØØØ1      1
3651  5555      JMP I SRCHER
3652  4263      JMS SRCONT      /CHK RELATION OF BLOCK
3653  56ØØ      JMP I SEARCH    /FOUND BLOCK, EXIT
3654  5555      JMP I SRCHER    /GONE PAST, ERROR
3655  7Ø4Ø      CMA
3656  1157      TAD BLKFLG
3657  765Ø      SNA CLA         /READ 2 BLOCK NUMS
366Ø  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
3663  5263  SRCONT,     JMP .
3664  2157      ISZ BLKFLG      /FIRST BLOCK NUMBER
3665  741Ø      SKP             /NOT FIRST
3666  53Ø2      JMP SRFBLK      /FIRST, JUST SAVE IT
3667  6761      IOT 761
367Ø  Ø137      AND KØ4ØØ
3671  764Ø      SZA CLA         /BKWRDS IS
3672  7Ø4Ø      CMA             /-1

3673  745Ø      SNA             /FWD IS
3674  7ØØ1      IAC             /+1
3675  116Ø      TAD PREBLK      /LAST BLOCK +OR-1
3676  7Ø41      CMA IAC
3677  1Ø41      TAD BLKFND
37ØØ  764Ø      SZA CLA         /BLOCKS CONSECUTIVE
37Ø1  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  7100       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  1Ø41  SRCMOD,     TAD BLKFND              /FIND DIFFERENCE
3756  7Ø41        CMA IAC         /IN NUMBER OF BLOCKS
3757  1Ø4Ø        TAD RECORD
376Ø  75ØØ        SMA
3761  7Ø41        CMA IAC         /MAKE -
3762  3443        DCA I WCLOC     /FOR WORD COUNT
3763  1135        TAD KØ1ØØ
3764  6764        IOT 764         /SET CONTIN MODE
3765  4466        JMS I WAITI
3766  ØØØ1        1
3767  5555        JMP I SRCHER    /NOT NORMAL INTERRUPT
377Ø  1Ø41        TAD BLKFND
3771  7Ø41        CMA IAC
3772  1Ø4Ø        TAD RECORD
3773  764Ø        SZA CLA         /RIGHT BLOCK
3774  5555        JMP I SRCHER            /NO, ERROR
3775  1135        TAD KØ1ØØ               /CLEAR CONTIN MODE
3776  6764        IOT 764
3777  56ØØ        JMP I SEARCH
```

```
                /SEARCH ERROR TYPEOUT

                *4ØØØ

4ØØØ   4467     SRHERR,     JMS I ERRSTP
4ØØ1   4545        JMS I SRCHTY
4ØØ2   1Ø71        TAD RECRDK
4ØØ3   45Ø6        JMS I TYPCON
4ØØ4   45Ø7        JMS I TYPTEX
4ØØ5   ØØ42        42
4ØØ6   5457        5457

4ØØ7   4353        4353
4Ø1Ø   ØØ67        67
4Ø11   4156        4156
4Ø12   6445        6445
4Ø13   44ØØ        44ØØ
4Ø14   77ØØ        77ØØ
4Ø15   1Ø76        TAD DIRFLG
4Ø16   765Ø     SNA CLA
4Ø17   5222        JMP .+3
4Ø2Ø   4714        JMS I BACKTY
4Ø21   741Ø        SKP
4Ø22   4715        JMS I FORDTY
4Ø23   1Ø42        TAD IDCON
4Ø24   45Ø6        JMS I TYPCON
4Ø25   45Ø7        JMS I TYPTEX
4Ø26   ØØ42        42
4Ø27   5457        5457
4Ø3Ø   4353        4353
4Ø31   ØØ46        46
4Ø32   5765        5765
4Ø33   5644        5644
4Ø34   77ØØ        77ØØ
4Ø35   7Ø4Ø        CMA
4Ø36   171Ø        TAD I SEKONS            /BLKFLG
4Ø37   771Ø        SPA CLA
4Ø4Ø   5252        JMP .+12
```

```
4Ø41   1311        TAD SEKONS+1            /PREBLK
4Ø42   45Ø6        JMS I TYPCON
4Ø43   45Ø7        JMS I TYPTEX
4Ø44   ØØ54        54
4Ø45   4163        4163
4Ø46   64ØØ        640Ø
4Ø47   4254        4254
4Ø5Ø   5743        5743
4Ø51   5377        5377
4Ø52   271Ø        ISZ I SEKONS
4Ø53   7ØØØ        NOP
4Ø54   131Ø        TAD SEKONS
4Ø55   45Ø6        JMS I TYPCON
4Ø56   45Ø7        JMS I TYPTEX
4Ø57   ØØ42        42
4Ø6Ø   5457        5457
4Ø61   4353        4353
4Ø62   63ØØ        630Ø
4Ø63   6245        6245

4Ø64   4144        4144
4Ø65   77ØØ        770Ø
4Ø66   6772        IOT 772
4Ø67   3313        DCA SEKONS+3
4Ø7Ø   1312        TAD SEKONS+2
4Ø71   45Ø6        JMS I TYPCON
4Ø72   45Ø7        JMS I TYPTEX

4Ø73   ØØ63        63
4Ø74   6441        6441
4Ø75   64ØØ        64ØØ
4Ø76   4277        4277
4Ø77   76Ø4        LAS
41ØØ   Ø154        AND K6ØØØ
41Ø1   765Ø        SNA CLA
41Ø2   74Ø2        HLT
41Ø3   2151        ISZ PASFLG              /2ND ERROR
41Ø4   57Ø7        JMP I .+3               /REWIND DRIVE
41Ø5   57Ø6        JMP I .+1
41Ø6   36Ø1        SEARCH+1
41Ø7   Ø525        REWDRV
411Ø   Ø157   SEKONS,     BLKFLG
4111   Ø16Ø        PREBLK
4112   4113        .+1
4113   ØØØØ        Ø
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
```

```
4247  ØØØØ        Ø
4250  525Ø  CRLFLF,      JMP .
4251  76Ø4        LAS
4252  771Ø        SPA CLA          /DELETE TYPEOUTS
4253  565Ø        JMP I CRLFLF
4254  127Ø        TAD KØ215                 /CARRIAGE RETURN
4255  6Ø46        TLS
4256  6Ø41        TSF
4257  5256        JMP .-1
426Ø  72ØØ        CLA
4261  1271        TAD KØ212                 /LINE FEED
4262  6Ø46        TLS
4263  6Ø41        TSF
4264  5263        JMP .-1
4265  6Ø42        TCF
4266  72ØØ        CLA
4267  565Ø        JMP I CRLFLF
427Ø  Ø215  KØ215,       215
4271  Ø212  KØ212,       212
```

```
                /TYPE CONTENTS OF ADDRESS IN AC

4272  5272  TYCONT,     JMP .
4273  332Ø      DCA TXSTOR      /SAVE ADDRESS
4274  172Ø      TAD I TXSTOR    /GET CONTENTS
4275  332Ø      DCA TXSTOR
4276  425Ø      JMS CRLFLF      /CARRIAGE RETURN - LINE FEED
4277  43Ø4      JMS TYCOVR      /TYPE UPPER OCTAL
43ØØ  43Ø4      JMS TYCOVR
43Ø1  43Ø4      JMS TYCOVR
43Ø2  43Ø4      JMS TYCOVR
43Ø3  5672      JMP I TYCONT

43Ø4  53Ø4  TYCOVR,     JMP .
43Ø5  132Ø      TAD TXSTOR
43Ø6  7ØØ6      RTL
43Ø7  7ØØ4      RAL
431Ø  332Ø      DCA TXSTOR
4311  132Ø      TAD TXSTOR
4312  7ØØ4      RAL
4313  Ø12Ø      AND KØØØ7
4314  1133      TAD KØØ2Ø
4315  7Ø4Ø      CMA             /MAKE - FOR
4316  423Ø      JMS TYCHAR      /OUTPUT
4317  57Ø4      JMP I TYCOVR
432Ø  ØØØØ  TXSTOR,     Ø


            /STOP TAPE ON ERROR, LEAVE FLAGS SET
4321  5321  ERSTP,      JMP .
4322  6761      IOT 761
4323  Ø113      AND KØ2ØØ
4324  1125      TAD KØØØ3
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 TYPTEX
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 TYPTEX
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  434Ø          JMS TYDIR
4446  4365          JMS TYMODE
4447  5641          JMP I TYWDAT


                /TYPE WRITE ALL DIRECTION AND MODE
445Ø  525Ø      TYWALL,     JMP .
4451  4311          JMS TYDRV
4452  4265          JMS TYWRIT
4453  43Ø3          JMS TYALL
4454  434Ø          JMS TYDIR
4455  4365          JMS TYMODE
4456  565Ø          JMP I TYWALL


                /TYPE READ
4457  5257      TYREAD,     JMP .
446Ø  45Ø7          JMS I TYPTEX
4461  6245          6245
4462  4144          4144
4463  ØØ77          ØØ77
4464  5657          JMP I TYREAD

                /TYPE WRITE
4465  5265      TYWRIT,     JMP .
4466  45Ø7          JMS I TYPTEX
4467  6762          6762
447Ø  5164          5164
4471  45ØØ          45ØØ
4472  77ØØ          77ØØ
4473  5665          JMP I TYWRIT


                /TYPE DATA
4474  5274      TYDATA,     JMP .
4475  45Ø7          JMS I TYPTEX
4476  ØØØØ          Ø
4477  4441          4441
45ØØ  6441          6441
45Ø1  ØØ77          ØØ77
45Ø2  5674          JMP I TYDATA
```

```
                /TYPE ALL
45Ø3  53Ø3      TYALL,      JMP .
45Ø4  45Ø7          JMS I TYPTEX
45Ø5  4154          4154
45Ø6  54ØØ          54ØØ
45Ø7  77ØØ          77ØØ
451Ø  57Ø3          JMP I TYALL


                /TYPE DRIVE AND NUMBER
4511  5311      TYDRV,      JMP .
4512  76Ø4          LAS
4513  771Ø          SPA CLA
4514  5711          JMP I TYDRV
4515  45Ø7          JMS I TYPTEX
4516  7777          7777
4517  7777          7777
452Ø  4462          4462
4521  5166          5166
4522  45ØØ          45ØØ
4523  ØØ77          ØØ77
4524  1Ø45          TAD CDRIVE
4525  745Ø          SNA
4526  1132          TAD KØØ1Ø
4527  1142          TAD K26Ø
453Ø  6Ø46          TLS
4531  6Ø41          TSF
4532  5331          JMP .-1
4533  6Ø42          TCF
4534  45Ø7          JMS I TYPTEX
4535  7777          7777
4536  77ØØ          77ØØ
4537  5711          JMP I TYDRV


                /TYPE FORWARDS OR BACKWARD
454Ø  534Ø      TYDIR,      JMP .
4541  6761          IOT 761
4542  Ø137          AND KØ4ØØ
4543  765Ø          SNA CLA
4544  5355          JMP TYFWD-2
4545  4347          JMS TYBKW
4546  574Ø          JMP I TYDIR
4547  5347      TYBKW,      JMP .
455Ø  45Ø7          JMS I TYPTEX
4551  4253          4253
4552  6744          6744
4553  ØØ77          ØØ77
4554  5747          JMP I TYBKW

4555  4357          JMS TYFWD
4556  574Ø          JMP I TYDIR
4557  5357      TYFWD,      JMP .
456Ø  45Ø7          JMS I TYPTEX
4561  4667          4667
4562  44ØØ          44ØØ
4563  77ØØ          77ØØ
4564  5757          JMP I TYFWD
```

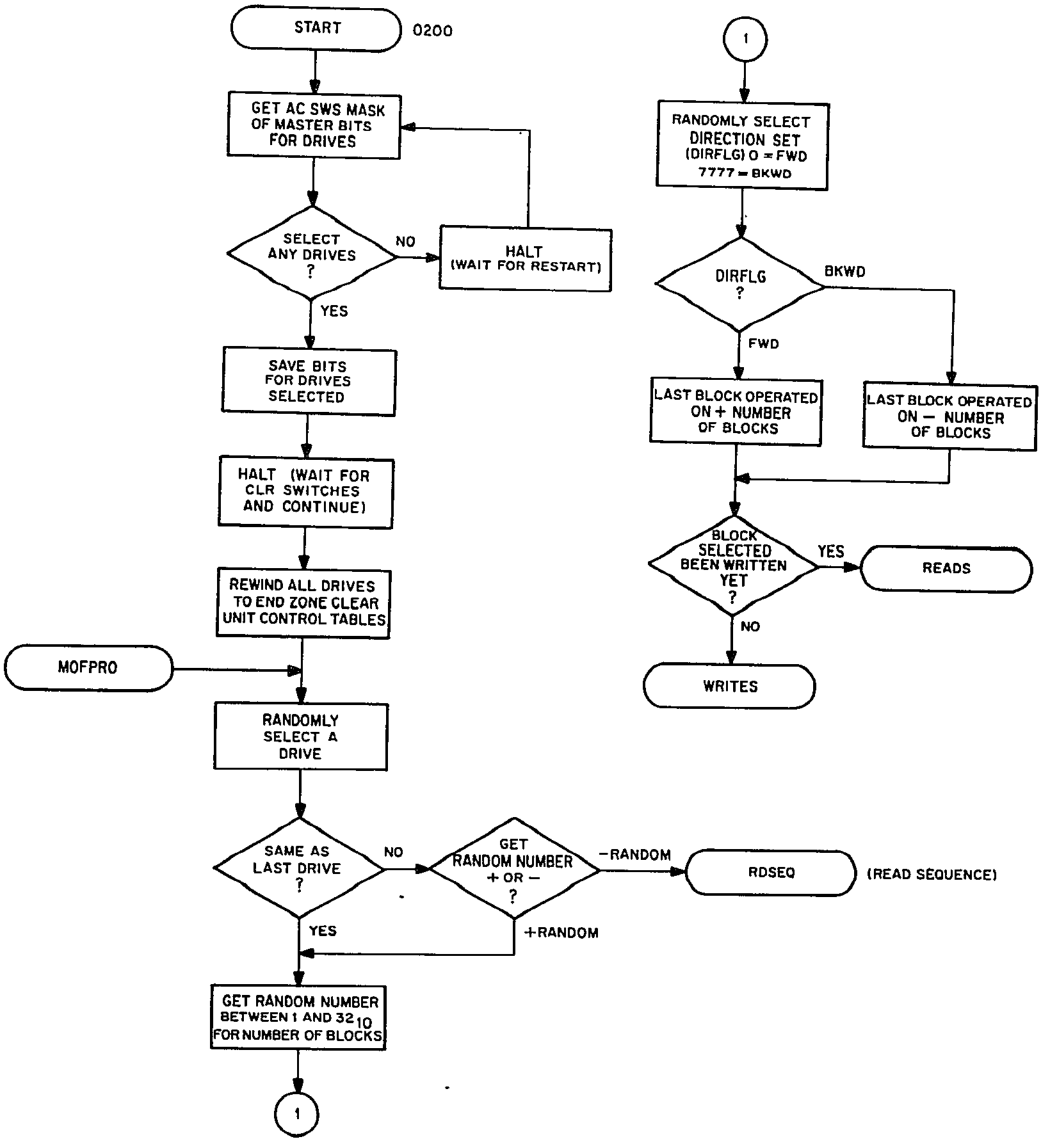
```
                /TYPE CONTINUOUS IF NOT NORMAL MODE
4565  5365      TYMODE,     JMP .
4566  6761          IOT 761
4567  Ø135          AND KØ1ØØ
457Ø  765Ø          SNA CLA
4571  5765          JMP I TYMODE
4572  45Ø7          JMS I TYPTEX
4573  43ØØ          43ØØ
4574  5557          5557
4575  4445          4445
4576  ØØ77          ØØ77
4577  5765          JMP I TYMODE


ACSAVE  3127
BACKTY  4114
BF1LOC  ØØ75
BF2LOC  Ø1Ø3
BLKFLG  Ø157
BLKFND  ØØ41
BUFFRS  7175
CALCTY  267Ø
CALOC   ØØ44
CDRIVE  ØØ45
CHNGDR  3437
CKSERR  256Ø
CKSTYP  2657
CKSUMR  2564
COCNTR  ØØ72
CODATA  1225
CODEX   1255
COERRO  1261
COERR1  1313
COLOOP  1243
COMBIT  ØØ5Ø
COMSTR  Ø755
CORT1R  1442
CRLFLF  425Ø
CSECON  27Ø1
DATACO  ØØ65
DATAMV  ØØ7Ø
DATASM  2561
DIRECT  ØØ52
DIRFLG  ØØ76
DIRTBL  1Ø44
DRIVTY  Ø13Ø
ERRCKS  26ØØ
ERRRAR  27Ø7
ERRSTP  ØØ67
ERSTP   4321
EXIST   1Ø25
EZBIT   Ø131
FORDTY  4115
FRSWAT  Ø162
FWDCHK  2562
GENDEX  Ø675
GENPAT  ØØ61
GENRAN  Ø735
GETRED  Ø57Ø
```

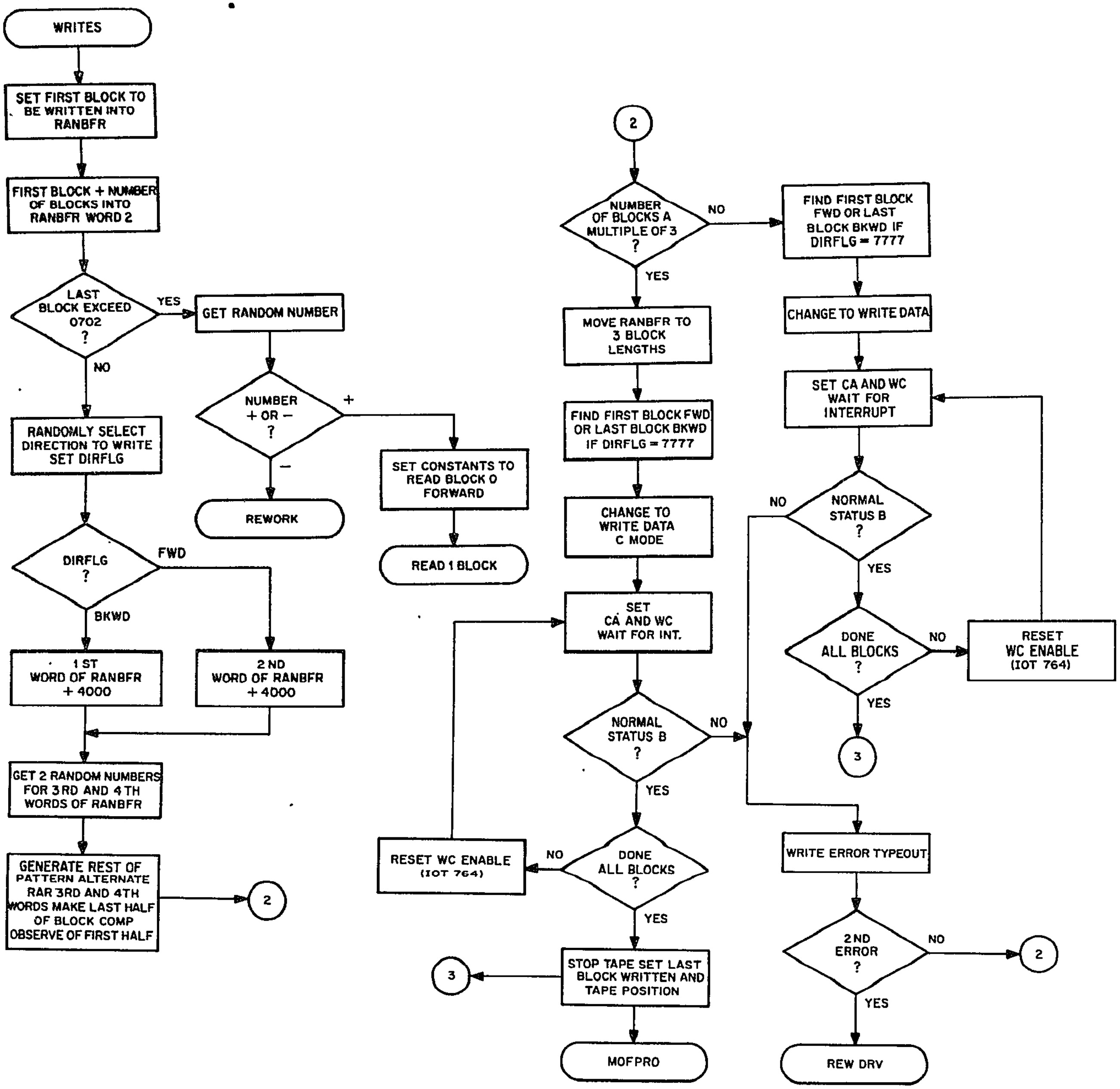
```
ACSAVE   3127
BACKTY   4114
BF1LOC   ØØ75
BF2LOC   Ø1Ø3
BLKFLG   Ø157
BLKFND   ØØ41
BUFFRS   7175
CALCTY   267Ø
CALOC    ØØ44
CDRIVE   ØØ45
CHNGDR   3437
CKSERR   256Ø
CKSTYP   2657
CKSUMR   2564
COCNTR   ØØ72
CODATA   1225
CODEX    1255
COERRO   1261
COERR1   1313
COLOOP   1243
COMBIT   ØØ5Ø
COMSTR   Ø755
CORT1R   1442
CRLFLF   425Ø
CSECON   27Ø1
DATACO   ØØ65
DATAMV   ØØ7Ø
DATASM   2561
DIRECT   ØØ52
DIRFLG   ØØ76
DIRTBL   1Ø44
DRIVTY   Ø13Ø
ERRCKS   26ØØ
ERRRAR   27Ø7
ERRSTP   ØØ67
ERSTP    4321
EXIST    1Ø25
EZBIT    Ø131
FORDTY   4115
FRSWAT   Ø162
FWDCHK   2562
GENDEX   Ø675
GENPAT   ØØ61
GENRAN   Ø735
GETRED   Ø57Ø
GNPATR   Ø626
GNPTRS   3461
IDCON    ØØ42
IOT      6ØØØ
IRECD    3Ø34
ISZTST   32ØØ
JMRETU   3321
JMSKON   3352
JMSLOC   3351
JMSTST   33ØØ
KØØØ2    Ø122
KØØØ3    Ø125
KØØØ7    Ø12Ø

KØØ1Ø    Ø132
KØØ2Ø    Ø133
KØØ3Ø    Ø116
KØØ37    Ø111
KØØ4Ø    Ø114
KØØ5Ø    Ø15Ø
KØØ7Ø    Ø123
KØØ77    Ø134
KØ1ØØ    Ø135
KØ12Ø    2574
KØ13Ø    2113
KØ15Ø    23Ø1
KØ2ØØ    Ø113
KØ212    4271
KØ214    Ø126
KØ215    427Ø
KØ24Ø    Ø136
KØ4ØØ    Ø137
KØ6Ø4    Ø14Ø
KØ614    Ø115
KØ7ØØ    Ø122
K1ØØØ    Ø131
K16      Ø141
K26Ø     Ø142
K27Ø1    Ø153
K3777    Ø143
K4ØØØ    Ø121
K5Ø75    Ø146
K5Ø76    Ø147
K6ØØØ    Ø154
K7ØØØ    Ø124
K7175    23ØØ
K7571    2575
K7577    Ø117
K76ØØ    Ø543
K76ØØX   3353
K77ØØ    Ø127
K776Ø    Ø11Ø
K7763X   3354
K777Ø    Ø112
K7772Y   3141
K7774    1354
LKSAVE   3126
LSTBLK   ØØ53
LSTDRV   ØØ54
LSTTBL   1Ø45
MCOMOB   Ø7Ø3
MIN4     Ø753
MIN6Ø    Ø754
MOFPRO   Ø221
MOFPR1   Ø235
MOVDEX   1222
MOVER    354Ø
MSBITS   ØØ47
MVDATA   12ØØ
NBIT8    35ØØ
NDOFRD   1725
NDPTST   335Ø
NDTSKP   3Ø55
NEWDRV   ØØ57

NUMBLK   ØØ77
PASFLG   Ø151
PATGEN   Ø62Ø
PICNTR   3142
PIDEX1   3143
PIDEX2   3144
PIRTBL   3154
PITBL    3145
PNTRS    35Ø1
POSITN   ØØ51
POSTBL   1Ø43
PRAN1    3346
PRAN2    3347
PREBLK   Ø16Ø
PREGEN   Ø547
PTSTND   3Ø64
PTSTNS   31Ø4
RADCHK   2512
RADLOC   2576
RAEFLG   2577
RALLTS   24ØØ
RANBFR   6774
RANDEX   Ø2ØØ
RANGEN   ØØ62
RANNØ    Ø751
RANSEL   1ØØØ
RANVAR   Ø752
RARERR   257Ø
RASERR   2571
RBFLOC   ØØ74
RBFWD2   Ø1Ø2
RBUFND   Ø1ØØ
RBUFST   ØØ73
RDBUF1   1632
RDCERR   2Ø75
RDCMOD   2ØØØ
RDDATA   Ø4ØØ
RDSEQ    16ØØ
RDTAB    Ø3Ø4
READTY   Ø144
READ1    ØØ63
READ2    Ø1Ø4
READ2A   Ø1Ø5
RECORD   ØØ4Ø
RECRDK   ØØ71
REGENP   ØØ64
REPOSI   34ØØ
REREAD   Ø42Ø
REREDA   Ø47Ø
RESETD   3Ø17
RETJMP   3355
RETUJM   Ø161
REVCHK   2563
REWCK    1517
REWDRV   Ø525
REWIND   ØØ56
ROT1TS   3222
ROT2TS   3236
RPATO    Ø7ØØ
RSFDRV   3417
```

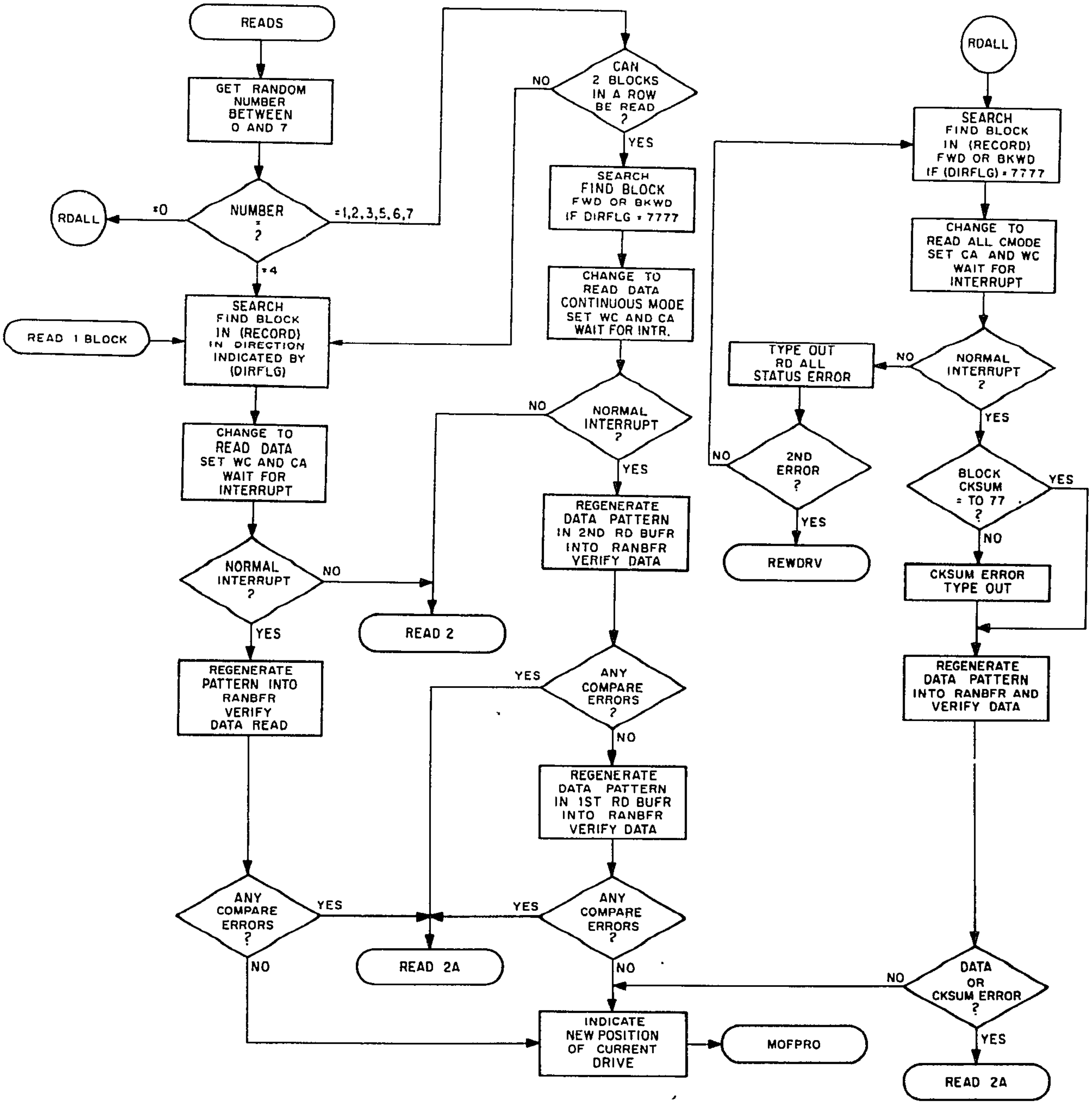
```
RSFDR1  3435        GNPATR  Ø626        MIN4    Ø723
RSQFLG  Ø35Ø        GNPTRS  3461        MIN6Ø   Ø724
SAMDRV  Ø325        IDCON   ØØ42        MOFPRO  Ø221
SAVNUM  157Ø        IOT     6ØØØ        MOFPR1  Ø235
SBRECD  313Ø        IRECD   3Ø34        MOVDEX  1222
SEARCH  36ØØ        ISZTST  32ØØ        MOVER   354Ø
SEKONS  411Ø        JMRETU  3321        MSBITS  ØØ47
SELGEN  1Ø46        JMSKON  3352        MVDATA  12ØØ
SELRAN  ØØ6Ø        JMSLOC  3351        NBIT8   35ØØ
SELVAR  1Ø61        JMSTST  33ØØ        NDOFRD  1725
SETIDX  3131        KØØØ2   Ø152        NDPTST  335Ø
SRCHER  Ø155        KØØØ3   Ø125        NDTSKP  3Ø55
SRCHIT  ØØ55        KØØØ7   Ø12Ø        NEWDRV  ØØ57
SRCHTY  Ø145        KØØ1Ø   Ø132        NUMBLK  ØØ77
SRCMOD  3755        KØØ2Ø   Ø133        PASFLG  Ø151
SRCONT  3663        KØØ3Ø   Ø116        PATGEN  Ø6ØØ
SREZTS  3726        KØØ37   Ø111        PICNTR  3142
SRFBLK  37Ø2        KØØ4Ø   Ø114        PIDEX1  3143
SRHERR  4ØØØ        KØØ5Ø   Ø15Ø        PIDEX2  3144
SRTARN  3635        KØØ7Ø   Ø123        PIRTBL  3154
SWCMDL  2223        KØØ77   Ø134        PITBL   3145
SWCMOD  2212        KØ1ØØ   Ø135        PNTRS   35Ø1
TADTST  3252        KØ12Ø   2574        POSITN  ØØ51
TAPONT  Ø156        KØ13Ø   2113        POSTBL  1Ø43
TEMP1   3341        KØ15Ø   23Ø1        PRAN1   3346
TEMP2   3342        KØ2ØØ   Ø113        PRAN2   3347
TEMP3   3343        KØ212   4271        PREBLK  Ø16Ø
TEMP4   3344        KØ214   Ø126        PREGEN  Ø547
TEMP5   3345        KØ215   427Ø        PTSTND  3Ø64
TRECTR  2277        KØ24Ø   Ø136        PTSTNS  31Ø4
TXSTOR  432Ø        KØ4ØØ   Ø137        RADCHK  2512
TYALL   45Ø3        KØ6Ø4   Ø14Ø        RADLOC  2576
TYBKW   4547        KØ614   Ø115        RAEFLG  2577
TYCHAR  423Ø        KØ7ØØ   Ø122        RALLTS  24ØØ
TYCONT  4272        K1ØØØ   Ø131        RANBFR  6774
TYCOVR  43Ø4        K16     Ø141        RANDEX  Ø2ØØ
TYDATA  4474        K26Ø    Ø142        RANGEN  ØØ62
TYDIR   454Ø        K27Ø1   Ø153        RANNØ   Ø751
TYDRV   4511        K3777   Ø143        RANSEL  1ØØØ
TYFWD   4557        K4ØØØ   Ø121        RANVAR  Ø752
TYMODE  4565        K5Ø75   Ø146        RARERR  257Ø
TYMOVE  44ØØ        K5Ø76   Ø147        RASERR  2571
TYPCON  Ø1Ø6        K6ØØØ   Ø154        RBFLOC  ØØ74
TYPTEX  Ø1Ø7        K7ØØØ   Ø124        RBFWD2  Ø1Ø2
TYRALL  4432        K7175   23ØØ        RBUFND  Ø1ØØ
TYRDAT  4423        K7571   2575        RBUFST  ØØ73
TYREAD  4457        K7577   Ø117        RDBUF1  1632
TYSRCH  4411        K76ØØ   Ø543        RDCERR  2Ø75
TYTEXT  42ØØ        K76ØØX  3353        RDCMOD  2ØØØ
TYWALL  445Ø        K77ØØ   Ø127        RDDATA  Ø4ØØ
TYWDAT  4441        K776Ø   Ø11Ø        RDSEQ   16ØØ
TYWRIT  4465        K7763X  3354        RDTAB   Ø3Ø4
UNFUNC  ØØ46        K777Ø   Ø112        READTY  Ø144
WAITI   ØØ66        K7772Y  3141        READ1   ØØ63
WATINT  3ØØØ        K7774   1354        READ2   Ø1Ø4
WCLOC   ØØ43        LKSAVE  3126        READ2A  Ø1Ø5
WDCERR  2253        LSTBLK  ØØ53        RECORD  ØØ4Ø
WDCMOD  22ØØ        LSTDRV  ØØ54        RECROK  ØØ71
WDINC   15Ø3        LSTTBL  1Ø45        REGENP  ØØ64
WRITES  Ø324        MCOMOB  Ø7Ø3        REPOSI  34ØØ
```

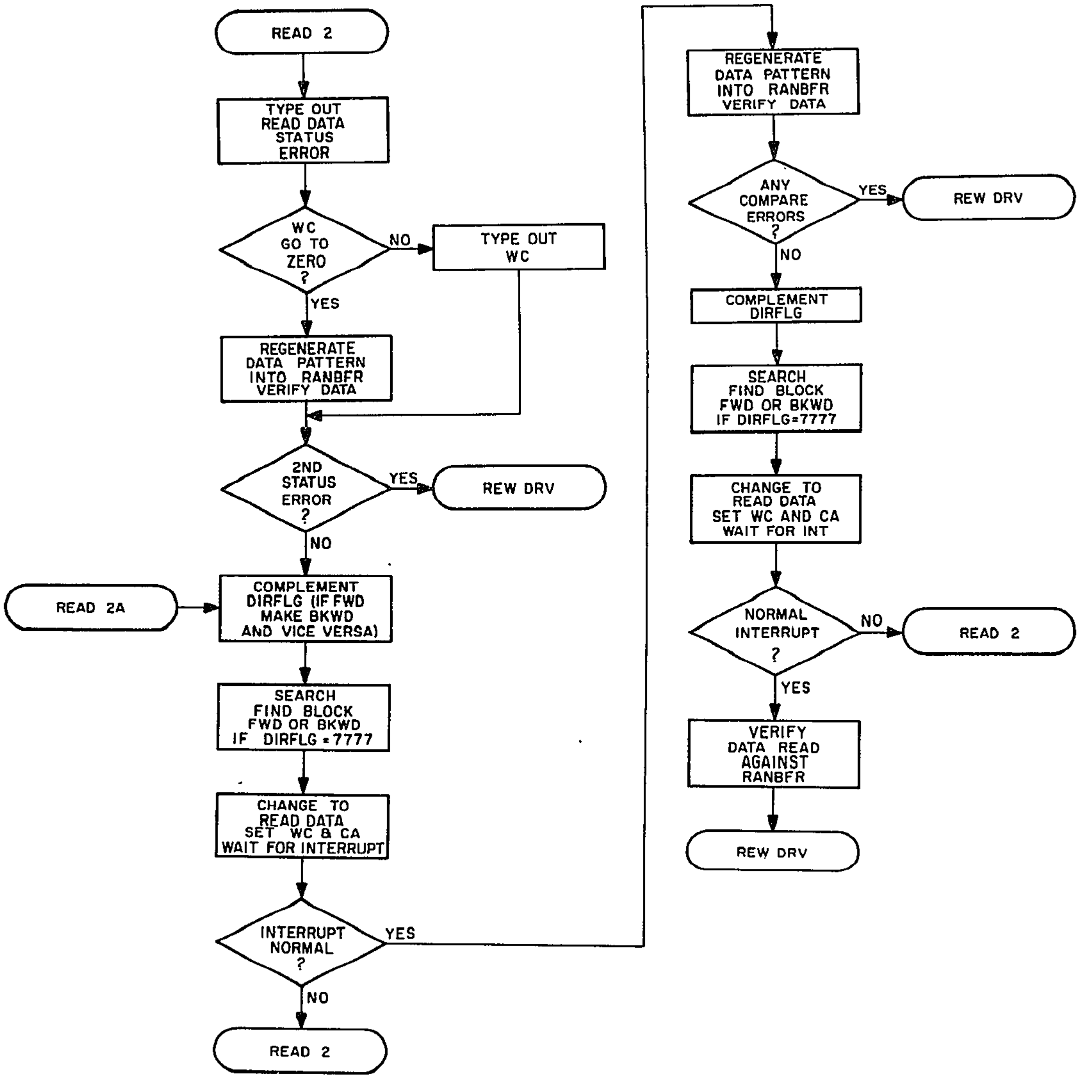
REREAD	Ø42Ø
REREDA	Ø47Ø
RESETD	3Ø17
RETJMP	3355
RETUJM	Ø161
REVCHK	2563
REWCK	1517
REWDRV	Ø525
REWIND	ØØ56
ROT1TS	3222
ROT2TS	3236
RPATO	Ø7ØØ
RSFDRV	3417
RSFDR1	3435
RSQFLG	Ø35Ø
SAMDRV	Ø325
SAVNUM	157Ø
SBRECD	313Ø
SEARCH	36ØØ
SEKONS	411Ø
SELGEN	1Ø46
SELRAN	ØØ6Ø
SELVAR	1Ø61
SETIDX	3131
SRCHER	Ø155
SRCHIT	ØØ55
SRCHTY	Ø145
SRCMOD	3755
SRCONT	3663
SREZTS	3726
SRFBLK	37Ø2
SRHERR	4ØØØ
SRTARN	3635
SWCMDL	2223
SWCMOD	2212
TADTST	3252
TAPONT	Ø156
TEMP1	3341
TEMP2	3342
TEMP3	3343
TEMP4	3344
TEMP5	3345
TRECTR	2277
TXSTOR	432Ø
TYALL	45Ø3
TYBKW	4547
TYCHAR	423Ø
TYCONT	4272
TYCOVR	43Ø4
TYDATA	4474
TYDIR	454Ø
TYDRV	4511
TYFWD	4557
TYMODE	4565
TYMOVE	44ØØ
TYPCON	Ø1Ø6
TYPTEX	Ø1Ø7
TYRALL	4432
TYRDAT	4423
TYREAD	4457
TYSRCH	4411
TYTEXT	42ØØ
TYWALL	445Ø
TYWDAT	4441
TYWRIT	4465
UNFUNC	ØØ46
WAITI	ØØ66
WATINT	3ØØØ
WCLOC	ØØ43
WDCERR	2253
WDCMOD	22ØØ
WDINC	15Ø3
WRITES	Ø324
WRITE1	14ØØ
WRTDTY	Ø1Ø1
WRT1A	1455
WSTERR	1526
WTCNTR	3125
WTKON1	3123
WTKON2	3124
XORSAV	2565
XORSUM	2531

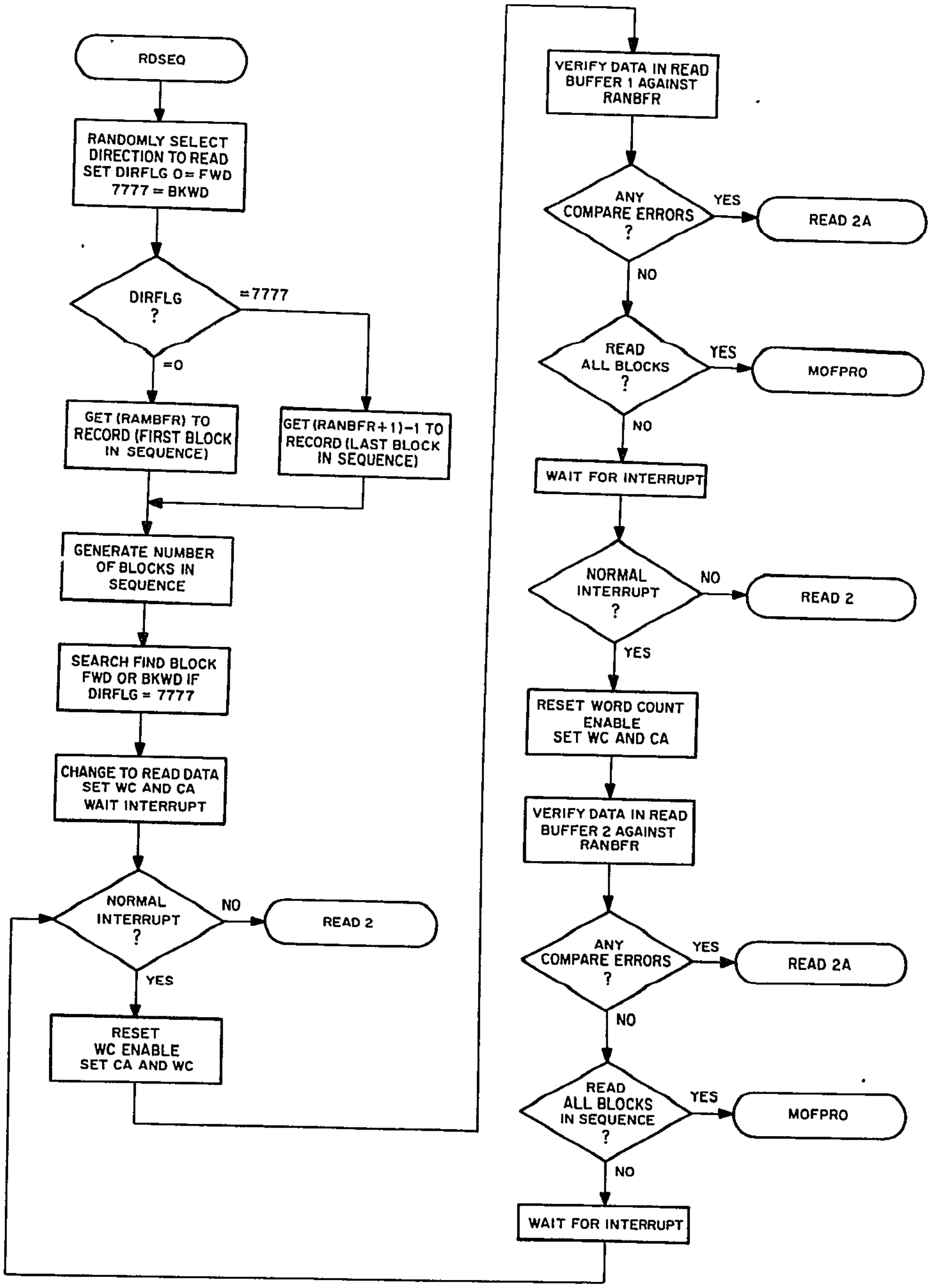
11. FLOW CHARTS


START
0200
GET AC SWS MASK OF MASTER BITS FOR DRIVES
SELECT ANY DRIVES ?
NO
HALT (WAIT FOR RESTART)
YES
SAVE BITS FOR DRIVES SELECTED
HALT (WAIT FOR CLR SWITCHES AND CONTINUE)
REWIND ALL DRIVES TO END ZONE CLEAR UNIT CONTROL TABLES
MOFPRO
RANDOMLY SELECT A DRIVE
SAME AS LAST DRIVE ?
NO
GET RANDOM NUMBER + OR − ?
−RANDOM
RDSEQ
(READ SEQUENCE)
YES
+RANDOM
GET RANDOM NUMBER BETWEEN 1 AND 32₁₀ FOR NUMBER OF BLOCKS
1
1
RANDOMLY SELECT DIRECTION SET (DIRFLG) 0 = FWD 7777 = BKWD
DIRFLG ?
BKWD
FWD
LAST BLOCK OPERATED ON + NUMBER OF BLOCKS
LAST BLOCK OPERATED ON − NUMBER OF BLOCKS
BLOCK SELECTED BEEN WRITTEN YET ?
YES
READS
NO
WRITES


WRITES
SET FIRST BLOCK TO BE WRITTEN INTO RANBFR
FIRST BLOCK + NUMBER OF BLOCKS INTO RANBFR WORD 2
LAST BLOCK EXCEED 0702 ?
YES
NO
GET RANDOM NUMBER
NUMBER + OR − ?
+
−
REWORK
SET CONSTANTS TO READ BLOCK 0 FORWARD
READ 1 BLOCK
RANDOMLY SELECT DIRECTION TO WRITE SET DIRFLG
DIRFLG ?
FWD
BKWD
1 ST WORD OF RANBFR + 4000
2 ND WORD OF RANBFR + 4000
GET 2 RANDOM NUMBERS FOR 3RD AND 4TH WORDS OF RANBFR
GENERATE REST OF PATTERN ALTERNATE RAR 3RD AND 4TH WORDS MAKE LAST HALF OF BLOCK COMP OBSERVE OF FIRST HALF
2
2
NUMBER OF BLOCKS A MULTIPLE OF 3 ?
NO
YES
FIND FIRST BLOCK FWD OR LAST BLOCK BKWD IF DIRFLG = 7777
CHANGE TO WRITE DATA
SET CA AND WC WAIT FOR INTERRUPT
NORMAL STATUS B ?
NO
YES
DONE ALL BLOCKS ?
NO
YES
RESET WC ENABLE (IOT 764)
3
MOVE RANBFR TO 3 BLOCK LENGTHS
FIND FIRST BLOCK FWD OR LAST BLOCK BKWD IF DIRFLG = 7777
CHANGE TO WRITE DATA C MODE
SET CA AND WC WAIT FOR INT.
NORMAL STATUS B ?
NO
YES
DONE ALL BLOCKS ?
NO
YES
RESET WC ENABLE (IOT 764)
STOP TAPE SET LAST BLOCK WRITTEN AND TAPE POSITION
3
MOFPRO
WRITE ERROR TYPEOUT
2ND ERROR ?
NO
YES
2
REW DRV


READS
GET RANDOM NUMBER BETWEEN 0 AND 7
NUMBER = ?
=0
RDALL
=1,2,3,5,6,7
=4
READ 1 BLOCK
SEARCH FIND BLOCK IN (RECORD) IN DIRECTION INDICATED BY (DIRFLG)
CHANGE TO READ DATA SET WC AND CA WAIT FOR INTERRUPT
NORMAL INTERRUPT ?
NO
YES
READ 2
REGENERATE PATTERN INTO RANBFR VERIFY DATA READ
ANY COMPARE ERRORS ?
YES
NO
READ 2A
CAN 2 BLOCKS IN A ROW BE READ ?
NO
YES
SEARCH FIND BLOCK FWD OR BKWD IF DIRFLG = 7777
CHANGE TO READ DATA CONTINUOUS MODE SET WC AND CA WAIT FOR INTR.
NORMAL INTERRUPT ?
NO
YES
REGENERATE DATA PATTERN IN 2ND RD BUFR INTO RANBFR VERIFY DATA
ANY COMPARE ERRORS ?
YES
NO
REGENERATE DATA PATTERN IN 1ST RD BUFR INTO RANBFR VERIFY DATA
ANY COMPARE ERRORS ?
YES
NO
INDICATE NEW POSITION OF CURRENT DRIVE
MOFPRO
RDALL
SEARCH FIND BLOCK IN (RECORD) FWD OR BKWD IF (DIRFLG) = 7777
CHANGE TO READ ALL CMODE SET CA AND WC WAIT FOR INTERRUPT
NORMAL INTERRUPT ?
NO
YES
TYPE OUT RD ALL STATUS ERROR
2ND ERROR ?
NO
YES
REWDRV
BLOCK CKSUM = TO 77 ?
YES
NO
CKSUM ERROR TYPE OUT
REGENERATE DATA PATTERN INTO RANBFR AND VERIFY DATA
DATA OR CKSUM ERROR ?
NO
YES
READ 2A


READ 2
TYPE OUT READ DATA STATUS ERROR
WC GO TO ZERO ?
NO
TYPE OUT WC
YES
REGENERATE DATA PATTERN INTO RANBFR VERIFY DATA
2ND STATUS ERROR ?
YES
REW DRV
NO
READ 2A
COMPLEMENT DIRFLG (IF FWD MAKE BKWD AND VICE VERSA)
SEARCH FIND BLOCK FWD OR BKWD IF DIRFLG = 7777
CHANGE TO READ DATA SET WC & CA WAIT FOR INTERRUPT
INTERRUPT NORMAL ?
YES
NO
READ 2
REGENERATE DATA PATTERN INTO RANBFR VERIFY DATA
ANY COMPARE ERRORS ?
YES
REW DRV
NO
COMPLEMENT DIRFLG
SEARCH FIND BLOCK FWD OR BKWD IF DIRFLG=7777
CHANGE TO READ DATA SET WC AND CA WAIT FOR INT
NORMAL INTERRUPT ?
NO
READ 2
YES
VERIFY DATA READ AGAINST RANBFR
REW DRV


RDSEQ
RANDOMLY SELECT DIRECTION TO READ SET DIRFLG 0= FWD 7777 = BKWD
DIRFLG ?
=7777
=0
GET (RAMBFR) TO RECORD (FIRST BLOCK IN SEQUENCE)
GET (RANBFR+1)-1 TO RECORD (LAST BLOCK IN SEQUENCE)
GENERATE NUMBER OF BLOCKS IN SEQUENCE
SEARCH FIND BLOCK FWD OR BKWD IF DIRFLG = 7777
CHANGE TO READ DATA SET WC AND CA WAIT INTERRUPT
NORMAL INTERRUPT ?
NO
READ 2
YES
RESET WC ENABLE SET CA AND WC
VERIFY DATA IN READ BUFFER 1 AGAINST RANBFR
ANY COMPARE ERRORS ?
YES
READ 2A
NO
READ ALL BLOCKS ?
YES
MOFPRO
NO
WAIT FOR INTERRUPT
NORMAL INTERRUPT ?
NO
READ 2
YES
RESET WORD COUNT ENABLE SET WC AND CA
VERIFY DATA IN READ BUFFER 2 AGAINST RANBFR
ANY COMPARE ERRORS ?
YES
READ 2A
NO
READ ALL BLOCKS IN SEQUENCE ?
YES
MOFPRO
NO
WAIT FOR INTERRUPT