

IDENTIFICATION

Product Code: Maindec-08-D3BB-D
Product Name: TC01 Basic Exerciser (Maindec 850)
Date: March 14, 1967
Maintainer: Diagnostic Group
Author: Keith F. Nelson

1. ABSTRACT

The TC01 Basic Exerciser is a series of test programs that may be used to gain a high degree of confidence in the data handling ability of a TC01 DECTape Control and one to eight TU55 DECTape Transports. The Basic Exerciser consists of several basic routines that may be individually selected; each routine will operate on any configuration of one to eight drives. These routines include a Basic Motion Routine, Search Find All Blocks Test, Basic Search Routine, Start/Stop/Turnaround Test, Basic Write/Read Data Test with eight selectable patterns, and a Parity Generation and Checking Test. The operation of the Basic Motion Routine and the Basic Search Routine are controlled by keyboard input. Also, a Write Data Scope Loop, Read Data Scope Loop, and a Search Scope Loop are provided to keep the tape moving from end zone to end zone.

2. REQUIREMENTS

2.1 Equipment

PDP-8 (standard)
 TC01 DECTape Control
 One to eight TU55 DECTape Transports

2.2 Storage

The program occupies most of memory from address 0000 to 6377 and utilizes three buffer areas as follows:

<u>Address</u>	<u>Function</u>
6774-7174	Output buffer Program storage for Motion Test (0200) Block Number storage for Basic Search (0202)
7175-7375	INPUT buffer 1
7376-7576	INPUT buffer 2

2.3 Preliminary Programs (None)

3. LOADING PROCEDURE

3.1 Method

Use normal binary loading procedures from paper tape.

4. STARTING PROCEDURE

4.1 Control Switch Settings

Any configuration of one to eight drives may be selected in SWITCH REGISTER bits 0 to 7. Each bit is a master bit for selection of a drive. When the switch is a 1 the drive is selected; when a 0 the drive is not selected.

<u>Switch</u>	<u>Drive</u>
0	8
1	1
2	2
3	3
4	4
5	5
6	6
7	7

4.2 Starting Addresses of Routines

<u>Address</u>	<u>Routine</u>	<u>Paragraph</u>
0200	Basic Motion Routine	9.1
0201	Search Find All Blocks	9.2
0202	Basic Search Routine	9.3
0203	Start/Stop/Turnaround	9.4
0204	Write/Read Data Test	9.5
0205	Parity Generation Test	9.6
0206	Write Data Scope Loop	9.7
0207	Read Data Scope Loop	9.8
0210	Search Scope Loop	9.9

4.3 Program and/or Operator Action

- a. Place the select address for the routine desired in the SWITCH REGISTER and press LOAD ADDRESS.
- b. Set SWITCH REGISTER bits 0 to 7 to select drives. (Any configuration except all 0s is valid.)
- c. Press START. The processor halts at address 0223 with bits 0 to 7 of the SWITCH REGISTER displayed in the AC.

d. A halt at address 0311 indicates bits 0 to 7 were all 0s. Select drives and press CONTINUE to recover.

e. Set all SWITCH REGISTER bits to 0, or as desired according to paragraph 5.1, and press CONTINUE.

A detailed description of how the routines can be used to initially check out the control and drives can be found in paragraph 5.3.

5. OPERATING PROCEDURE

5.1 Operational Switch Settings

5.1.1 Routines with no Switch Settings - Four of the routines require different switch settings to control program flow. The routines that have no switch settings are:

0200	Basic Motion Routine
0202	Basic Search Routine
0205	Parity Generation
0207	Read Scope Loop
0210	Search Scope Loop

5.1.2 Search Find All Blocks - The Search Find All Blocks Routine (0201) has one switch setting. Setting SW11 to 1 deletes the halt at the end of test.

5.1.3 Write/Read Data Test - The Write/Read Data Test (0204) utilizes switches 3 to 11 to control pattern selection and program flow as follows:

<u>Switch</u>	<u>Operation</u>
3	Delete all error detection where the motion bit in status A remains 1 (parity, data compare errors, and WC (word count register) not equal to 0).
4	Run patterns sequentially; i.e., After making one complete pass the length of tape with pattern 5, the next pass is made with pattern 6.
5	Read data only (after the first write pass).
6	Write data only (SW5 overrides SW6).
7	Write and read sequence, one block at a time.
8	Write and read sequence, 32 blocks at a time. (SW7 overrides SW8, when both switches = 0, the write and read sequence occurs for the length of the tape).

<u>Switch</u>	<u>Operation</u>
9, 10, and 11	Indicate pattern selection as follows:
000	All 0s
001	All 1s
010	Alternate words of 0s and 1s
011	Words of 2525
100	Words of 5252
101	Words of 0707
110	Words of 7070
111	Alternate words of 2525 and 5252

5.1.4 Write Scope Loop - The Write Scope Loop (0206) utilizes switches 9, 10, and 11 for pattern selection in the same manner as the Write Data/Read Data Test.

5.1.5 Start/Stop/Turnaround - The Start/Stop/Turnaround Test (0203) uses switch 1 to delete stop after error, and halt at end of test.

5.2 Subroutine Aspects (None)

5.3 Program and/or Operator Action

This series of routines is designed for initial check-out of a TC01 DECTape Control and its associated drives, or maintenance and repair of the control and drives after installation.

The following procedure is used for initial check-out of the control and drives and can be followed to repair malfunctions once the control and drives have been operating:

5.3.1 Operation Check - The first routine utilized is the Basic Motion Routine (0200). It is used to visually verify the following operations with the use of an oscilloscope, the indicators on the TC01 indicator panel, and by watching the motion of the tape on the DECTape drive.

5.3.1.1 Initial Control State - When power is initially applied to the TC01 Control, status A, the error and DECTape flags, and the data flag can come up in any state. A short manual procedure will prevent erasing DECTapes and having to reload programs.

Set the SWITCH REGISTER to 0.

Press LOAD ADDRESS.

Select SINGLE STEP.

Press DEPOSIT.

Press LOAD ADDRESS.

Press START (to generate a POWER CLEAR).

Now examine the TC01 indicator panel, the following indicators should all be off, indicating a 0.

- DTF (DECtape flag)
- DF (data flag)
- All ERROR flags
- W (WREN write enable)
- Status A bit 4 (motion)
- US (up to speed)
- C0 to C3 can be in any stable state (not counting)
- All state register bits except I should be 0, and bit I should be a 1 (state idle)

5.3.1.2 Clear and Load Status A (IOT 762, 764, and 766) - The basic operation and existence of these DECTape IOTs can be verified as follows:

Start the Basic Motion Routine with all drives SWITCH REGISTER selected and off line. Type the following program:

```
"F" WD
"W" AIT 0100
"C" HNG
"R" PT 0002
"D" O
```

Now, watch the status A indicators 0, 1, and 2. They should go to 0_g and remain there for slightly more than a second, then proceed to 1_g, 2_g, 3_g etc., up to 7_g, and return to 0_g and repeat the process. In addition, a select error should be generated for each drive selection and the MOTION bit should be set to 0. By increasing the "W"AIT count or restarting the program with each drive individually selected, the decoding of the drive number to a single select line can be monitored with an oscilloscope; or the selection indicators over the rotary select switch on the TU55s may be used by placing the drives on line and including a "S"TOP command after the "F"WD.

5.3.1.3 Tape Motion and Timing Pulse Generation - Put one drive on line and start the Basic Motion Routine with that drive SWITCH REGISTER selected. To verify basic operations of the control and drive motion controls: type the following series of short programs:

```
"F"WD
"D"O
```

The tape on the selected drive should start moving forward (off the left-hand reel and onto the right-hand reel). A select error should not be generated and bit 4 of status A should remain 1 unless end zone is reached and detected. C0 to C3 should appear to be counting, indicating timing pulse generation; US (up to speed) should set to 1 within a short period after tape starts moving. The DTF should not set. Now type:

```
"S"TOP
"D"O
```

Forward tape motion on the selected drive should stop. (Bit 4 of status A should go to 0). The left-hand brake on the drive should be set and the right-hand reel should be free with a small amount of torque holding the tape tight. Again, no select error occurs. Now type:

```
"B"KWD
"D"O
```

The tape on the selected drive should start moving backwards (off the right-hand reel and onto the left). Status A bit 3 should be 1 (BKWD). All other indicator observations for forward should be true. Again type:

```
"S"TOP
"D"O
```

Backward tape motion should stop. Bit 3 of status A should remain 1 and bit 4 should go to 0. The right-hand drive brake should be set and the left-hand reel should be free with only enough torque to hold the tape tight.

5.3.1.4 New U + M Delay - The new unit and motion delay can be generated by any of several short programs, but its operation must be monitored with an oscilloscope. Since the delay time could change at a later date (for some currently unknown reason), the time will not be mentioned here; but it can be determined from the TC01 logic diagrams. An example of a program that could be used is:

Manually move the tape until approximately an even amount of tape is on both reels and type:

```
"F"WD
"W"AIT 0020
"B"KWD
"W"AIT 0020
"R"PT 0040 (or a shorter or longer count as desired)
"D"O
```

5.3.1.5 End Zone Detection - The program can be used to determine if the end zone is being detected by starting the tape in either direction and watching whether or not the tape runs off the reel. Also

watch the end bit in the error status. If the END indicator lights and the tape does not stop, error stop in the control is not being generated. In either case, return the tape to the reel a short distance from the start of the reel (less than 10 feet of tape on right hand reel) and use following program to scope the end-zone detection.

```
"F"WD
"W"AIT 0020
"B"KWD
"W"AIT 0016
"R"PT 0040 (or may be made longer or shorter as desired)
"S"TOP
"D"O
```

The forward wait count can be decreased if the tape rocks forward or increased if the end zone approaches too quickly or if the tape runs off the reel. (At that end of the reel, the tape will move backward faster than forward.)

5.3.2 Check End-Zone Detection - For the next sequence of operations, any of the three search routines (0201, 0202, or 0210) could be used; but the Search Scope Loop (0210) is the most practical and least complicated. When the routine is initiated, the tape starts forward until the end zone is detected and then runs backward until end zone is again detected and then forward again.

If the tape runs off the reel, either the end zone was not detected or bit 2 of status B (END) did not read to the processor accumulator during a Read Status B IOT. As the tape is moving forward, make the following observations:

C0 to C3 should appear to be incrementing, indicating timing pulses are being generated.

US (up to speed) should indicate a 1 shortly after the tape starts moving and should stay on.

The STATE REGISTER should circulate and appear to remain mainly in state data.

The DECTape flag indicator should glow visibly, dim, and glow again as the tape moves forward (The program does not monitor DTF but simply waits in an ISZ loop and periodically monitors END and MOTION).

No error statuses should be generated except end zone.

The processor accumulator should appear to be incrementing by 1 as each successive block number is read from tape and displayed.

The timing in the control should be monitored with an oscilloscope with reference to the DEC-tape TC01 timing diagrams.

With the DECTape searching backward, the same observations may be made as forward except the processor accumulator should appear to decrement.

5.3.3 Correct Block Number - At this point it is suggested that the Search Find All Blocks Routine starting at 0201 be used to prove that the control will correctly read block numbers. The Basic Search Routine starting at 0202 may be used to gain more information if 0201 does not run without error typeouts.

5.3.4 Check Read Data Timing - The next step should be to verify the Read Data Timing with an oscilloscope utilizing the Read Scope Loop (0207) and the TC01 timing diagrams.

5.3.5 Check Write Data Timing - Next, the Write Scope Loop (0206) may be run and the Write Data timing verified. This routine changes to Search Between Blocks as an effort to keep from writing over block numbers. (Recheck the tape with 0201 or 0202 to verify this).

The different data patterns may be utilized visually as follows, (W (WREN) should indicate 1 for all patterns).

Pattern 0 (all 0s)	DATA BUFFER bit indicators 6, 7, and 8 should glow dimly and the rest of the DATA BUFFER should appear to be 0s. RWB bits 3, 4, and 5 should appear to remain 0s. RWB bits 0, 1, and 2 should be complementing and should glow fairly brightly but not solidly. The LPB should complement every six bits and will glow dimly.
Pattern 1 (all 1s)	DATA BUFFER bits 6, 7, and 8 should glow dimly and the rest of the DATA BUFFER should appear to be steady 1s. RWB bits 3, 4, and 5 should appear to remain steady 1s; bits 0, 1, and 2 should complement and glow fairly brightly but not solidly. The LPB contents are not predictable but the rate of change should be fairly slow and discernable. (The LPB only complements on 0s and will contain the complement of the reverse checksum of the block it is passing over).
Pattern 2 (alternate words of 0s and 1s)	All bits in the DATA BUFFER, RWB, and LPB should glow dimly.
Pattern 3 (2525)	The even numbered bits of the buffers should act as pattern 0 and the odd numbered bits as pattern 1.
Pattern 4 (5252)	The even numbered bits of the buffers should act as pattern 1 and the odd numbered bits as pattern 0.
Pattern 5 (0707)	The rightmost three bits (of each six bits) should appear as pattern 1 and the leftmost as pattern 0.

Pattern 6 (7070)	The leftmost three bits (of each six bits) should appear as pattern 1 and the rightmost as pattern 0.
Pattern 7 (2525 alternate with 5252)	Should appear as pattern 2. No steady states discernible in the buffers.

5.3.6 Prepare Tape for Read - The Write Scope Loop may now be used to prepare a tape for the Read Scope Loop and for a further visual verification. Patterns 3, 4, 5, and 6 appearing in the BUFFER(s) indicators should read the same in either direction.

Note that the DATA BUFFER bits 6, 7, and 8 appear to be in a steady state and not to complement. Patterns 0 and 1 should be complemented when read in the direction opposite that in which they were written. No steady states should be discernible with patterns 2 and 7.

5.3.7 Check Correct Data - Run the Write/Read Data Test to verify that data is correctly read and written. Utilize the different switch configurations (see paragraph 5.1) for a complete test or to scope loop the reads or writes. This routine does not change to search between blocks, thus the possibility that block numbers may be written over is greater than that of the Write Data Scope Loop.

5.3.8 Check Checksum Generation - The Parity Generation Test verifies that checksums are being generated properly and that parity errors will be detected if they occur.

5.3.9 Check Turnaround Function - Run the Start/Stop/Turnaround Test (0203). All of the other routines are designed to eliminate the possibility of a turnaround error, but this routine tests this function to a much tighter limit.

6. ERRORS

Almost all hardware malfunctions detected by the program result in an error message typed on the Teletype. Each error message includes drive number, operation, direction, mode, error status, block being operated on, and correct and incorrect data, if applicable.

6.1 Error Typeout Descriptions

6.1.1 Search Error Typeouts - The Search Error Typeouts are in several formats. The Search Routine used by the Parity Test and Write/Read Data Test uses the following format:

DRIVE X	(A)
SEARCH FWD (or BKWD)	(B)
XXXX BLOCK WANTED FWD (or BKWD)	(C)
XXXX BLOCK FOUND	(D)
XXXX LAST BLOCK (if BLOCKS READ \geq 002)	(E)
XXXX BLOCKS READ	(F)
XXXX STAT B	(G)

A. This will be the first line of every typeout. Drive X is the drive that was being operated at the time of the error.

B. The second line of every typeout indicates the DECTape function, direction and mode. (Typeout will be C MODE for continuous mode).

C. This is the block number that the search routine should find as an end result and the direction that the block should be found in. If the direction in line B is the same as the direction in line C, the turnaround for finding the block has already been made. If the two directions are different, the error occurred before turnaround.

D. This is the contents of symbolic register BLKFND and could indicate one of the following:

1. Should be ignored if BLOCKS READ = 0000 and the directions in line B and C disagree. It could indicate the turnaround block, if the directions are the same and BLOCKS READ = 0000.

2. That the DECTape did not turn around in two PDP-8 block lengths, if BLOCKS READ = 0001, STAT B = 0001, and the directions indicated are the same.

3. The BLOCK in error, if BLOCKS READ does not = 0000 and STAT B is an error status (i.e., 6000 MARK TRACK ERROR) other than END ZONE (5000).

E. This line of the typeout is included only if two or more block numbers have been received since the search operation was started, or since the direction bit in status A was complemented for turnaround. Examine STAT B and if it does not equal 0001 ignore this line. If STAT B does = 0001, LAST BLOCK compared against BLOCK FOUND will indicate that the last two block numbers read were not sequential.

F. The number of block numbers received since the search operation was initiated or since turnaround.

G. This is the DECTape status B register; if STAT B does not = 0001, this is the error condition that caused the typeout. If STAT B equals 5000 (end-zone interrupt), and the directions in lines B and C are the same, it means that the drive made one turnaround and went the length of the tape without finding the block that the search routine was looking for. An end-zone error before turnaround indicates that at least one block number had been read, and that the block wanted was two or more blocks from end zone in the direction opposite the search. (i.e., BLOCK 3 WANTED FWD or BLOCK 2677 WANTED BKWD.)

The Start/Stop Turnaround Test has two formats for search error typeouts:

```

DRIVE 1
SEARCH BKWD
0005 BLOCK (Tape should have been up to speed by this block)
0006 FWD LAST POS (Last known tape position)
0004 FOUND (Block number in error)
0001 STAT B (If not 0001 indicates error was a status error)
    
```

In this case, notice that the difference between BLOCK and LAST POSITION is 1 and that the operations were in opposite directions. This indicates a turnaround error. If these lines differ by more than 1, the error would have been on a start-up.

The other error timeout format occurs if block numbers are not sequential.

```
DRIVE 1
SEARCH FWD
BLK # ERROR
0010 BLOCK
0006 LAST (Block 6 should have been followed by block 7)
0010 THIS (Not by block 10)
0001 STAT B
```

6.1.2 Read Data Status Error Timeouts - The first three lines of the read-data timeouts are in the same format as the search timeouts. The first two lines contain drive number, operation and direction, and the third line is the block being operated on. Again, depending upon which test routine is being run, one of several timeouts could occur.

```
DRIVE 1
READ DATA FWD
0046 BLOCK
4301 STAT B
```

(Combination parity error and timing error)

```
DRIVE 2
READ DATA BKWD
0100 BLOCK
4201 STAT B
7757
```

(This timeout is used by the Parity Generation Test. The last line of this timeout indicates the data pattern written to test parity. In this case, the reverse checksum is 20; CHECKSUM going forward was 75. The LPB at the end of a block in read data should always be 77 for normal operation).

```
DRIVE 2
READ DATA FWD
0100 BLOCK
PARITY ERROR EXPECTED
0001 STAT B
0200
```

(This timeout is also used by the Parity Generation Test and could follow the one above. The timeout indicates that a parity error should have been generated, but was not received. Again, the last line of the timeout indicates the data pattern written to test the parity circuitry. Notice the complement observe relationship between the two data timeouts. In this case, the CHECKSUM has been rewritten to 02 in WRITE ALL, it was 75 after WRITE DATA, and the LPB should have been 00 after reading the block. READ DATA and STATE CHECK going to 0 and LPB not equal to 77 is 1 to PARITY ERROR. See paragraph 9.6 for a complete description of the parity test.)

```
DRIVE 4
READ DATA FWD
0077 BLOCK
0001 STAT B
7777 WC
```

In the read data timeouts, the contents of the word count register (address 7754) are included only if the WC did not go to 0. Or if the DECTape status B was normal (0001) and the WC did not go to 0, the above timeout would occur.

6.1.3 Checksum Error Typeouts - The Parity Generation Test writes various data patterns in the first and second characters of each block. Since the reverse checksum is written to 00, the checksum generated by the TC01 should either be the complement of the first character in the block or if the first two characters are written should equal 77. The following typeouts could occur if the parity generation is failing.

DRIVE 1	
CKSUM ERROR	
2000 DATA	(First word of block, as read from tape)
7700 CK SUM	(As read from tape in READ ALL, in this case should equal 5700).

DRIVE 1	
CKSUM ERROR	
5757 DATA	(As read from tape)
5700 CK SUM	(As read from tape in READ ALL, in this case should be 7700)

6.1.4 Write Data Status Error Typeouts - Write data error typeouts also include drive, operation and direction, block being operated on, and the error status.

DRIVE 6	
WRITE DATA FWD	
0765 BLOCK	
6000 STAT B	
7715 WC	(This typeout indicates a mark-track error while doing a Write Data Forward on block 0765. The WC typeout indicates that the error occurred with 13g words left to be written.)

DRIVE 7	
WRITE DATA BKWD	
1000 BLOCK	
0001 STAT B	
7777 WC	(If STAT B indicates a normal block interrupt (0001) and the WC has not gone to 0, this typeout occurs.)

6.1.5 Data Error Typeouts - A data error may or may not follow a parity error typeout; it could also occur without a parity error. Again, the first three lines of the typeout are the same as for search errors: drive, operation and direction, and block number.

DRIVE 4	
READ DATA BKWD DATA ERROR	
0325 BLOCK	
0000 KNOWN	(Data written)
7773 UNKNOWN	(Data read)
7000 ADDRS KNOWN	(Buffer Address of data written)

6.1.6 Error Halts - The cause of any error halt not accompanying a typeout can be found by examining the program listing at the address of the halt. These may be caused by:

- a. A status other than EZ while in MOVE TAPE.
- b. The AC not being cleared after an IOT766 or 764.
- c. AC bits 0 to 7 equal to all 0s when initially starting.
- d. Program interrupt and no DECTape skip.
- e. No program interrupt for 45 seconds.

6.2 Error Recovery

There are no manual error-recovery procedures. In the cases of read data and read errors in the Parity Test, the programs proceed to the next sequential block in an effort to gain more information about the failure. For search (except 0201) or write errors, the same operation is attempted again.

In Search Test 0201, the program attempts to pick up the next block in sequence.

Any error halt that occurs without a timeout may indicate a completely non-logical type of failure. Examine the program listing to determine the meaning of the halt.

9. PROGRAM DESCRIPTION

9.1 Basic Motion Routine (0200)

This routine is a visual verification of the operation of the DECTape drives and some sections of the TC01 Control. The sequence of operations is selected by keyboard input from the Teletype. The keys that may be typed to select operations are "F," "B," "S," "C," "W," "R," and "D." All other keys will cause the execute table to be reset and previous selections to be lost. The operations selected by the individual keys are as follows:

<u>Key</u>	<u>Operation</u>
F FWD (Timeout)	Start moving tape on the currently selected drive in the forward direction.
B BKWD (Timeout)	Start moving tape on the currently selected drive in the backward direction.
S STOP (Timeout)	Stop tape on the currently selected drive.
C CHNG (Timeout)	Change drive selection and repeat from the beginning of the execute table or from the last "C."
W WAIT (Timeout)	Wait a variable number of blocks. The number of blocks to wait is typed in, immediately following the timeout "WAIT," and is a 4-digit number from 0000 to 7777. NOTE: The program does not actually count blocks but sits in an ISZ loop 18 msec for every increment typed in.

<u>Key</u>	<u>Operation</u>
R RPT (Typeout)	Repeat the sequence of operations from the start of the execute table or from the last "R." Again, the number of times to repeat is typed in immediately following the typeout "RPT" and is a 4-digit octal number from 0000 to 7777.
D DO (Typeout)	Causes the sequence of operations previously typed in to be executed NOTE: "D" can only be typed in as the first character after a sequence of operations has once been executed. This is true each time that the routine is restarted from address 0200. Typing a "D" as the first character causes the last sequence of operations to be executed.

9.2 Search Find All Blocks (0201)

Before a program can verify that the DECTape system can write correctly, it must prove that the system can read correctly. Since a DECTape with a, so-called, virgin tape pattern is not always readily available and DECTape with correctly written block numbers is usually available, the first verification of read operations must be a Search Test. Search Find All Blocks moves the DECTape backward into the end zone, reads the tape forward and verifies that blocks are numbered 0000 to 2701; then moves the tape into forward end zone, reverses the tape and tests that blocks are numbered 2701 to 0000. If SW11 is 0, the processor halts; press CONTINUE, and the program will repeat. If SW11 is 1, the processor will not halt and the program will repeat.

9.3 Basic Search Routine (0202)

In this routine, the tape is searched in either direction until a series of 129 block numbers is read and stored. (Or until end zone is reached or some error status is generated). The decision is made to either type out all of the block numbers or to have the program verify that the block numbers read are sequential. When started the program types:

```
DRIVE 8 (or whichever drive is selected)
TYPE IN F FOR FORWARD
ALL OTHERS BACKWARD
```

At this point, type in an "F" to search forward or any other key to search backwards. The program will search in the direction selected until an error status or end zone occurs, or until 129 block numbers have been read and stored in memory. It then types:

```
XXXX STAT B           (If an error status and then repeat the initial typeout)
END ZONE              (If the tape went into end zone before 129 blocks were
                      read)
NO BLOCKS             (If no block numbers were read)
XXXX FIRST BLOCK      (First block number read)
TYPE C FOR COMPARE
ALL OTHERS PRINT
```

To have the program verify that the block numbers are sequential, type in a "C." Any other character typed in causes the program to type out the complete series of block numbers. If a "C" is typed, the program types out block numbers that are not sequential. The program always types the last block number read as follows:

XXXX LAST

9.4 Start/Stop/Turnaround Test (0203)

When the ability to correctly read block numbers has been established, a more thorough test of the DECTape motion controls can be given. The Start/Stop/Turnaround Test verifies the following operations:

TURN AROUND Both directions on BLOCK 0
 Start FORWARD/STOP
 Start BACKWARD/STOP
 Start FORWARD/Wait UP TO SPEED/Turnaround
 Start BACKWARD/Wait UP TO SPEED/Turnaround

The sequence is repeated for the length of tape. Turnaround occurs in both directions on block 2701.

Since the tape is up to full speed before turnaround, the tape must be up to speed again by the time it returns to that same point on the tape.

9.5 Write/Read Data Test (0204)

The search routines establish a minimum capability to read known data from tape. This routine establishes the ability to write data and further establishes the ability to read data. The test includes eight selectable data patterns and three selectable modes of operation. The basic sequence of operation is write forward, read backward, read forward, write backward, read forward, read backward. The sequence may be selected for 1 block at a time, 32 blocks at a time, or the length of tape. The program recycles and runs until STOP is depressed. At the end of each complete sequence (the length of tape), the program types out the pattern number and END. The eight write patterns are as follows:

0 0000
 1 7777
 2 0000, 7777, 0000
 3 2525
 4 5252
 5 0707
 6 7070
 7 2525, 5252, 2525

The pattern to be written is selected in SWITCH REGISTER bits 9, 10, and 11. Place the number of the pattern desired in these switches.

Switches 7 and 8 are used to select the sequence of operation as follows:

<u>SW7</u>	<u>SW8</u>	<u>Operation</u>
0	0	Write and read sequence the length of the tape.
0	1	Write and read sequence in 32 block increments.
1	0	Write and read sequence one block at a time.
or 1	1	
SW4 = 0		Take the next pattern to be exercised from SWs 9, 10, and 11.
SW4 = 1		Exercise sequentially through the patterns; i.e., after one complete sequence the length of tape with pattern number 3, exercise pattern number 4, after exercising 4 go to 5. Patterns are not changed until block 2701 has been written backwards.
SW3 = 0		Type out parity error information and data errors.
SW3 = 1		Ignore parity and data errors. Mark track, timing, and select errors are not ignored.
SW6 = 0		Sequence from write to read data.
SW6 = 1		Write data only.
SW5 = 0		Sequence from read data to write data.
SW5 = 1		Read data only (SW5 overrides SW6).

9.6 Parity Generation and Checking Test (0205)

The complete test of parity generation and checking requires several passes over a series of blocks. The steps that the program takes for a complete test of the parity circuitry are as follows:

- STEP 1 Write reverse checksums to 0 (Actually written to 77 going backward and should equal 00 going forward).
- STEP 2 Write data patterns
Various data patterns are written in the first and second characters of each block and the rest of the block is written to zeros (Note: the checksums generated are either the complement of the first character or 77, if the first two characters are written).
- STEP 3 Read/Verify checksums
The checksums are read back and verified to be the complement of the first character in the block or 77, if the first two characters of block are non-zero.
- STEP 4 Test no parity errors
The blocks are read in both directions and no parity errors should be generated.

- STEP 5 Write blocks to wrong parity
 The checksums are written to be the same as the first character in the block so that the LPB will not equal 77 when the block is read.
- STEP 6 Test for parity errors
 The blocks are read in both directions and parity errors should be generated.

The program then repeats from step 1 and will run until STOP is depressed.

If an error timeout is generated indicating PARITY ERROR EXPECTED, the contents of the LPB can be determined by the following procedure:

- a. The timeout includes the first data word of the block if read forward or the last word of the block if read backward (actually same word but complement obverse if read backward).
- b. This word will contain either one or two non-zero 6-bit characters, (FWD); or one or two characters that do not equal 77 (BKWD).
- c. If there is only one 6-bit character, the LPB should be all 0s at the time it is strobed for parity error. This is true whether read occurred in a forward or a backward direction.
- d. If the read direction is forward and there are two non-zero characters in the first word, the LPB should be equal to one of the characters at the time it is strobed for parity error; i.e., WORD = 0202, LPB = 02.
- e. If the read direction is backward and there are two characters not equal to 77, the LPB should be equal to the complement of one of the characters when it was strobed for parity error; i.e., WORD = 5757, LPB = 20.

9.7 Write Data Scope Loop (0206)

This routine starts forward in search. When a block number is found, the program changes to write data for one block, then back to search and then to write data again. The program continues in that mode until end zone. Upon reaching end zone the tape is started backwards in search and is again changed to write data when a block is found. Each time an end zone interrupt is received, the tape direction is reversed. For any other error status, the function is reset to search and tape direction is not reversed. Any of the eight data patterns in the Write/Read Data Test may be selected by placing the pattern number in switches 9, 10, 11. (See paragraph (5.1.3)). The routine has to be restarted from 0207 to change pattern selection. This routine contains error halts if the AC is not cleared after an IOT764 or 766.

9.8 Read Scope Loop (0207)

This routine starts forward in read data and reads in 129-word blocks. When end zone is reached the tape is run backwards in read data. For any other error, the tape continues in read data in the same direction. Each time an end zone is reached, tape direction is reversed. This routine also contains error halts that indicate the accumulator was not cleared after an IOT766 or 764.

9.9 Search Scope Loop (0210)

This routine starts forward in search function and reverses direction at end zones. The DEC-tape flag and all error statuses except end zone are ignored. The program starts forward in search and displays the last block number received in the AC while doing an ISZ/JMP .-1 loop for approximately 13 msec. At completion of the ISZ loop, the program tests for end-zone status and complements the direction bit if end zone was reached. If end zone was not reached, search enables are reset and the motion bit in status A is set to a 1 if it was cleared. This routine contains error halts if the AC is not cleared after an IOT766 or 764 and if the motion bit is not cleared by EZ.

10. LISTING

```

/TC01 BASIC EXERCISER TAPE 1
/PAGE 0 CONSTANTS AND TEMP STORAGE
IOT=6000
BUFFRS=6774

```

```

BUFFER2=BUFFRS+201
BUFFER3=BUFFER2+201

```

```

*1
0001 5402 JMP I 2 /FOR INTERRUPTS

```

```

*20

```

```

0020 0000 RECORD, 0 /BLOCK OPERATED ON
0021 0000 BLKFND, 0 /BLOCK FOUND BY SEARCH
0022 0000 POSITN, 0 /TO GET TAPE POSITION
0023 0000 DIRECT, 0 /TO GET LAST DIRECTION
0024 0000 LSTBLK, 0 /TO GET LAST BLOCK WRITTEN
0025 0021 IDCON, BLKFND /FOR SRCH CA
0026 0020 RECRDK, RECORD /FOR TYPEOUTS

```

```

/ADDRESSES FOR INDIRECT TAD AND DCA

```

```

0027 7754 WCLOC, 7754
0030 7755 CALOC, 7755
0031 6774 BF1WD1, BUFFRS
0032 6773 BF1LOC, BUFFRS-1
0033 7174 BF2LOC, BUFFER2-1
0034 7375 BF3LOC, BUFFER3-1

```

```

/SUBROUTINE ADDRESS
0035 0600 SRCHIT, SEARCH
0036 0400 REWIND, REPOSI
0037 0437 NEWDRV, CHNGDR
0040 0234 WAITI, WTINT
0041 0313 WTHALF, WT500
0042 5000 DATACO, CODATA
0043 4100 SAVPAD, DAPSAV
0044 1310 ERRSTP, ERSTP
0045 1511 DRIVTY, TYDRV
0046 1424 RDATTY, TYRDAT
0047 1316 SBTYPE, TSTATB
0050 1442 WDATTY, TYWDAT

```

0051	1261	TYPCON,	TYCONT
0052	1411	SRCHTY,	TYSRCH
0053	4000	ERSSTA,	SSTAER
0054	3600	SSTFWD,	SSTAFW
0055	3651	SSTBKW,	SSTABW
0056	1200	TYPTX,	TYTEXT

/TEMP STORAGE FOR DRIVE SELECTION

0057	0000	CDRIVE,	0	/DRIVE NUMBER AND
0060	0000	UNFUNC,	0	/POSITIONED FOR STAT
0061	0000	MSBITS,	0	/DRIVES SELECTED
0062	0000	COMBIT,	0	

/OTHER CONSTANTS

0063	7760	K7760,	7760
0064	7577	K7577,	7577
0065	0604	K0604,	0604
0066	4000	K4000,	4000
0067	7767	K7767,	7767
0070	7700	K7700,	7700
0071	0077	K0077,	77
0072	0240	K0240,	240
0073	0007	K0007,	7
0074	0020	K0020,	20
0075	0200	K0200,	200
0076	0003	K0003,	3
0077	5077	K5077,	5077
0100	2701	K2701,	2701
0101	0614	K0614,	614
0102	0400	K0400,	400
0103	0214	K0214,	214
0104	0016	SFAFK,	16
0105	0006	SFABK,	6
0106	7763	SBABK,	7763
0107	7772	SBAFK,	7772
0110	1000	EZBIT,	1000
0111	0000	POSSAV,	0
0112	0000	DIRSAV,	0
0113	0000	BLKINC,	0
0114	0000	DIRFLG,	0
0115	0050	K0050,	50
0116	0030	K0030,	30
0117	0030	BLKBTS,	0030
0120	0020	BLKBIT,	0020
0121	0040	K0040,	40
0122	0170	K0170,	170
0123	0100	K0100,	100
0124	0101	K0101,	101
0125	0204	K0204,	204
0126	5076	K5076,	5076
0127	0037	K0037,	37
0130	0010	K0010,	10

/SELECT AND START TESTS
 /SWITCHES = MASTER BIT SELECTION
 /FOR TAPES

*200

0200 4211 JMS CIPHER
 0201 4211 JMS CIPHER
 0202 4211 JMS CIPHER
 0203 4211 JMS CIPHER
 0204 4211 JMS CIPHER
 0205 4211 JMS CIPHER
 0206 4211 JMS CIPHER
 0207 4211 JMS CIPHER
 0210 4211 JMS CIPHER

0211 0000 CIPHER, 0
 0212 7404 OSR
 0213 0063 AND K7760
 0214 7440 SZA
 0215 5220 JMP ,+3
 0216 7402 HLT
 0217 5212 JMP CIPHER+1
 0220 3061 DCA MSBITS
 0221 6774 IOT 774 /CLR STATUS B MEM FIELD

0222 1061 TAD MSBITS
 0223 7402 HLT
 0224 7200 CLA
 0225 1211 TAD CIPHER
 0226 1064 TAD K7577 /-201
 0227 1277 TAD TSTTBL
 0230 3232 DCA ,+2
 0231 4633 JMS I ,+2
 0232 5700 JMP I TSTTBL+1 /GO TO TEST SELECTED
 0233 0417 RSFDRV

/ABOVE JMP I IS CHANGED TO JMP I TST TBL+1 + THE
 /TEST NUMBER SELECTED

0234 5234 WTINT, JMP ,
 0235 1272 TAD WTIJMP
 0236 3001 DCA 1
 0237 1273 TAD WTIJMP+1
 0240 3002 DCA 2
 0241 3274 DCA WTIJMP+2
 0242 1276 TAD K4215
 0243 3275 DCA WTIJMP+3
 0244 6001 ION
 0245 2274 ISZ WTIJMP+2 /WAIT A MAXIMJM
 0246 5245 JMP ,-1 /OF 35 SECONDS
 0247 2275 ISZ WTIJMP+3 /FOR AN INTERRUPT
 0250 5245 JMP ,-3
 0251 6002 IOF
 0252 7402 HLT
 0253 5252 JMP ,-1

```

0254 6771 IRECD,      IOT 771      /DIF JR DTEF = 1
0255 7410      SKP
0256 5261      JMP ,+3
0257 7402      HLT
0260 5254      JMP IRECD
0261 6772      IOT 772
0262 3274      DCA WTIJMP+2
0263 1634      TAD I WTINT

0264 7040      CMA
0265 0274      AND WTIJMP+2
0266 7650      SNA CLA
0267 2234      ISZ WTINT
0270 2234      ISZ WTINT
0271 5634      JMP I WTINT
0272 5402      WTIJMP,      JMP I 2

0273 0254      IRECD
0274 0000      0
0275 0000      0
0276 4215      K4215,      4215

0277 5700      TSTTBL,      JMP I ,+1
    
```

/STARTING ADDRESSES OF TESTS

```

0300 2000      MVTEST
0301 2400      SRCH1
0302 2600      SERCH2
0303 3200      SSTRNA
0304 4400      WRTTST
0305 5600      PARTST
0306 1600      WRSCOP
0307 1667      RDSCOP
0310 1734      SRSCOP
    
```

```

0311 7402      HLTNS,      HLT      /TEST SELECTED
0312 5311      JMP , -1      /NOT AVAILABLE
0313 5313      WT500,      JMP ,
0314 7200      CLA
0315 3274      DCA WTIJMP+2      /TIME OUT
0316 1325      TAD KM25
0317 3275      DCA WTIJMP+3      /APPRX 500 MSEC
0320 2274      ISZ WTIJMP+2
0321 5320      JMP , -1
0322 2275      ISZ WTIJMP+3
0323 5320      JMP , -3
0324 5713      JMP I WT500
0325 7747      KM25,      7747
    
```

/REWIND ALL DRIVES SELECTED
/TO END ZONE AT START OF TAPE

```

*400
0400 5200 REPOSI,      JMP ,
0401 4217     JMS RSFDRV      /RESET POINTRS TO FIRST DRIVE
0402 1065     TAD K0604        /MOVE BACKWARDS
0403 1060     TAD UNFUNC      /*POSITIONED JNIT NO
0404 6766     IOT 766         /SET STATUS A
0405 4440     JMS I WAITI
0406 5001     5001            /INDICATE EXPECT END
0407 7402     HLT             /NOT STATUS EXPECTED

0410 7240     CLA CMA
0411 3422     DCA I POSITN    /INDICATE END ZONE
0412 7240     CLA CMA
0413 3423     DCA I DIRECT    /INDICATE BACKWARDS
0414 4237     JMS CHNGDR      /SET UP NEXT DRIVE
0415 5202     JMP REPOSI+2    /REWIND NEXT DRIVE
0416 5600     JMP I REPOSI    /GOT ALL DRIVES, EXIT

```

/RESET CURRENT DRIVE POINTERS TO
/FIRST DRIVE SELECTED

```

0417 5217 RSFDRV,      JMP ,
0420 7200     CLA
0421 3057     DCA CDRIVE      /SET INITIALLY TO 0
0422 1066     TAD K4000
0423 3062     DCA COMBIT
0424 1061     TAD MSBITS

0425 0062     AND COMBIT
0426 7640     SZA CLA         /THIS DRIVE SELECTED
0427 5235     JMP RSFDR1      /YES, SET POINTER
0430 1062     TAD COMBIT
0431 7110     CLL RAR
0432 3062     DCA COMBIT      /MOVE COMPARE BIT
0433 2057     ISZ CDRIVE      /INCREMENT DRIVE NUM.
0434 5224     JMP RSFDRV+5

```

/HAVE FOUND FIRST DRIVE SELECTED

```

0435 4261 RSFDR1,     JMS GNPTRS    /GENERATE CONTROL POINTERS
0436 5617     JMP I RSFDRV        /EXIT

```



```

/SELECT NEXT DRIVE OR
/RESET TO FIRST DRIVE AND SKIP
0437 5237 CHNGDR, JMP .
0440 7200 CLA
0441 1062 TAD COMBIT /GET DRIVE COMPARE BIT
0442 7110 CLL RAR /MOVE IT TO NEXT
0443 0067 AND K7767
0444 7440 SZA /LAST DRIVE NJM 7
0445 5251 JMP ,+4 /NO
0446 4217 JMS RSFDRV /RESET TO FIRST

0447 2237 ISZ CHNGDR /INCR. EXIT, SKIP
0450 5637 JMP I CHNGDR /EXIT
0451 3062 DCA COMBIT
0452 2057 ISZ CDRIVE
0453 1062 TAD COMBIT
0454 0061 AND MSBITS
0455 7650 SNA CLA /THIS DRIVE SELECTED
0456 5240 JMP CHNGDR+1 /NO

0457 4261 JMS GNPTRS /GENERATE DRIVE POINTERS
0460 5637 JMP I CHNGDR

```

```

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

```

```

0461 5261 GNPTRS, JMP .
0462 1057 TAD CDRIVE /DRIVE NUMBER
0463 7112 CLL RTR
0464 7012 RTR /POSITION TO BITS 0,1,2
0465 3060 DCA UNFUNC
0466 1057 TAD CDRIVE /DRIVE NUMBER
0467 1300 TAD PNTRS /* POS, PNTR ADDRS.
0470 3022 DCA POSITN /FOR INDIRECTS:

0471 1057 TAD CDRIVE
0472 1311 TAD PNTRS+11 /* DIRECTION PNTR
0473 3023 DCA DIRECT /FOR INDIRECTS.
0474 1322 TAD PNTRS+22
0475 1057 TAD CDRIVE
0476 3024 DCA LSTBLK
0477 5661 JMP I GNPTRS

```

0500	0501	PNTRS,	.+1	/TO GET LAST RECORD NUMBER
0501	0000	0		/FOR DRIVE 8
0502	0000	0		/1
0503	0000	0		/2
0504	0000	0		/3
0505	0000	0		/4
0506	0000	0		/5
0507	0000	0		/6
0510	0000	0		/7
0511	0512	.+1		/TO GET LAST DIRECTION
0512	0000	0		/DIRECTION - JNIT 8
0513	0000	0		/1
0514	0000	0		/2
0515	0000	0		/3
0516	0000	0		/4
0517	0000	0		/5
0520	0000	0		/6
0521	0000	0		/7
0522	0523	.+1		/TO GET LAST WRITTEN
0523	0000	0		/8
0524	0000	0		/1
0525	0000	0		/2
0526	0000	0		/3
0527	0000	0		/4
0530	0000	0		/5
0531	0000	0		/6
0532	0000	0		/7

/SEARCH ROUTINE
 /FIND BLOCK IN (RECORD) IN
 /DIRFLG=7777 BKWD #0 FWD

*600

0600	5200	SEARCH,	JMP .	
0601	1114	TAD	DIRFLG	
0602	7100	CLL		
0603	7640	SZA	CLA	
0604	7120	STL		
0605	1076	TAD	K0003	/MAKE=3 IF FWD
0606	7420	SNL		
0607	7041	CMA	IAC	/MAKE 3 IF BKWD
0610	1020	TAD	RECORD	/BLOCK + OR - 2 FOR TA
0611	3351	DCA	TAPONT	
0612	1114	TAD	DIRFLG	
0613	7650	SNA	CLA	/FORWARD IS
0614	1102	TAD	K0400	/START BACKWARD
0615	1103	TAD	K0214	
0616	1060	TAD	UNFUNC	/+DRIVE NUMBER
0617	6766	IOT	766	
0620	7040	CMA		
0621	3347	DCA	BLKFLG	
0622	1025	TAD	IDCON	
0623	3430	DCA	I CALOC	
0624	4440	JMS	I WAITI	/WAIT FOR NORMAL
0625	0001	1		
0626	5321	JMP	SREZTS	/COULD BE EZ
0627	4255	JMS	SRCNCK	/FOUND TURN AROUND
0630	5234	JMP	SRTAFN	/YES, TURN AROUND
0631	5234	JMP	SRTAFN	/PAST IT, TURN AROUND
0632	6764	IOT	764	/NOT REACHED YET
0633	5224	JMP	, -7	
0634	1020	SRTAFN,	TAD RECORD	
0635	3351	DCA	TAPONT	
0636	6761	IOT	761	
0637	7040	CMA		
0640	0075	AND	K0200	/IN CASE MOTION=0
0641	1102	TAD	K0400	
0642	6764	IOT	764	/CHANGE DIRECTION
0643	7040	CMA		
0644	3347	DCA	BLKFLG	
0645	4440	JMS	I WAITI	
0646	0001	1		/HAS TO BE NORMAL
0647	5746	JMP	I SRCHER	/OR ERROR
0650	4255	JMS	SRCNCK	
0651	5600	JMP	I SEARCH	/FOUND BLOCK, EXIT
0652	5746	JMP	I SRCHER	/WENT PAST, ERROR
0653	6764	IOT	764	/NOT THERE YET
0654	5245	JMP	, -7	

0655	5255	SRCNCK,	JMP .	
0656	2347	ISZ	BLKFLG	/FIRST BLOCK IN
0657	7410	SKP		/NO
0660	5274	JMP	SBCONS	
0661	6761	IOT	761	
0662	0102	AND	K0400	
0663	7640	SZA	CLA	/BACKWARD IS
0664	7040	CMA		/-1
0665	7450	SNA		/FORWARD IS
0666	7001	IAC		/+1
0667	1350	TAD	PREBLK	
0670	7041	CMA	IAC	
0671	1021	TAD	BLKFND	/BLOCKS SEQUENTIAL
0672	7640	SZA	CLA	
0673	5746	JMP	I SRCHER	/NO, ERROR
0674	1351	SBCONS,	TAD TAPONT	
0675	7041	CMA	IAC	
0676	1021	TAD	BLKFND	
0677	7450	SNA		/FIND BLOCK YET
0700	5655	JMP	I SRCNCK	/YES, TA OR EXIT
0701	2255	ISZ	SRCNCK	/STEP ADDR
0702	7100	CLL		
0703	7710	SPA	CLA	
0704	7120	STL		/L=1 IS BLK FND LESS
0705	6761	IOT	761	
0706	0102	AND	K0400	
0707	7640	SZA	CLA	/FORWARD
0710	5316	JMP	,+6	/NO BACKWARD
0711	7430	SZL		/FORWARD AND: BLKFND
0712	2255	ISZ	SRCNCK	/LESS IS NOT THERE YET
0713	1021	TAD	BLKFND	
0714	3350	OCA	PREBLK	
0715	5655	JMP	I SRCNCK	
0716	7420	SNL		/BACKWARD AND BLKFND
0717	2255	ISZ	SRCNCK	/LESS IS GONE PAST
0720	5313	JMP	, -5	

0721	6772	SREZTS,	IOT 772	
0722	0110	AND	EZBIT	
0723	7650	SNA	CLA	
0724	5746	JMP	I SRCHER	/NOT END ZONE, ERROR
0725	1351	TAD	TAPONT	
0726	7510	SPA		
0727	5333	JMP	,+4	/BLOCK 0 OR 1
0730	1126	TAD	K5076	
0731	7710	SPA	CLA	/BLOCK 2700 OR 2701
0732	5342	JMP	,+10	
0733	7200	CLA		
0734	1075	TAD	K0200	
0735	6764	IOT	764	/SET MOTION
0736	4440	JMS	I WAITI	/WAIT FOR EZ AGAIN
0737	5000	5000		
0740	5746	JMP	I SRCHER	
0741	5234	JMP	SRTAFN	/DO TURN AROUND
0742	1347	TAD	BLKFLG	/IF EZ WAS
0743	7700	SMA	CLA	/FIRST INT WAS VALID
0744	5746	JMP	I SRCHER	/IF NOT FIRS IS INVALID
0745	5234	JMP	SRTAFN	
0746	1000	SRCHER,	SRHERR	
0747	0000	BLKFLG,	0	
0750	0000	PREBLK,	0	
0751	0000	TAPONT,	0	

/SEARCH ERROR TYPEOUT

	*1000			
1000	4444	SRHERR,	JMS I ERRSTP	
1001	4452	JMS	I SRCHTY	
1002	1026	TAD	RECRDK	
1003	4451	JMS	I TYPCON	/TYPE BLOCK SEARCHED
1004	4456	JMS	I TYPTEX	
1005	0042	42		
1006	5457	5457		
1007	4353	4353		
1010	0067	67		
1011	4156	4156		
1012	6445	6445		
1013	4400	4400		
1014	7700	7700		
1015	1114	TAD	DIRFLG	
1016	7650	SNA	CLA	
1017	5222	JMP	,+3	
1020	4675	JMS	I BACKTY	
1021	7410	SKP		
1022	4676	JMS	I FORDTY	
1023	1025	TAD	IDCON	
1024	4451	JMS	I TYPCON	/TYPE BLOCK FOUND
1025	4456	JMS	I TYPTEX	
1026	0042	42		
1027	5457	5457		
1030	4353	4353		

1031	0046	46	
1032	5765	5765	
1033	5644	5644	
1034	7700	7700	
1035	7040	CMA	
1036	1671	TAD I SEKONS	/BLKFLG
1037	7710	SPA CLA	/MORE THAN 1 BLOCK
1040	5252	JMP ,+12	/NO
1041	1272	TAD SEKONS+1	/PREBLK
1042	4451	JMS I TYP CON	/TYPE LAST BLOCK
1043	4456	JMS I TYPTX	/FOUND
1044	0054	54	
1045	4163	4163	
1046	6400	6400	
1047	4254	4254	
1050	5743	5743	
1051	5377	5377	
1052	2671	ISZ I SEKONS	
1053	7000	NOP	
1054	1271	TAD SEKONS	
1055	4451	JMS I TYP CON	/TYPE NUMBER OF
1056	4456	JMS I TYPTX	/BLKS NUM READ
1057	0042	42	
1060	5457	5457	
1061	4353	4353	
1062	6300	6300	
1063	6245	6245	
1064	4144	4144	
1065	7700	7700	
1066	4447	JMS I SBTYPE	
1067	5670	JMP I ,+1	
1070	0601	SEARCH+1	
1071	0747	SEKONS, BLKFLG	
1072	0750	PREBLK	
1073	1074	,+1	
1074	0000	0	
1075	1545	BACKTY, TYBKW	
1076	1555	FORDTY, TYFWD	

/TYPE TEXT ROUTINE

```

*1200
1200 5200 TYTEXT,      JMP .
1201 7200          CLA
1202 1600          TAD I TYTEXT      /GET NEXT 2 CHARACTERS
1203 7040          CMA                /MAKE -
1204 3307          DCA TXSTOR
1205 2200          ISZ TYTEXT
1206 1307          TAD TXSTOR
1207 7440          SZA                /CARRIAGE RETJRN - LINE FEED
1210 5213          JMP ,+3           /NO

1211 4242          JMS CRLFLF        /CR LF
1212 5201          JMP TYTEXT+1      /GET NEXT
1213 0070          AND K7700          /CLEAR TO UPR CHAR
1214 7450          SNA                /END OF MESSAGE
1215 5600          JMP I TYTEXT      /YES
1216 7012          RTR                /MOVE
1217 7012          RTR                /OVER
1220 7012          RTR                /6 PLACES
1221 4230          JMS TYCHAR        /OUTPUT
1222 1307          TAD TXSTOR
1223 0071          AND K0077
1224 7450          SNA                /END OF MESSAGE
1225 5600          JMP I TYTEXT      /YES EXIT

1226 4230          JMS TYCHAR        /OUTPUT
1227 5201          JMP TYTEXT+1      /GET NEXT 2

1230 5230 TYCHAR,      JMP .
1231 7040          CMA                /MAKE * AGAIN
1232 0071          AND K0077          /CLEAR TO LOWER 6
1233 1072          TAD K0240         /MAKE ASCII
1234 6046          TLS                /OUTPUT
1235 6041          TSF                /WAIT FLAG
1236 5235          JMP , -1
1237 7200          CLA
1240 6042          TCF                /CLEAR FLAG
1241 5630          JMP I TYCHAR      /DO NEXT

1242 5242 CRLFLF,      JMP .
1243 1257          TAD K0215         /CARRIAGE RETURN
1244 6046          TLS
1245 6041          TSF
1246 5245          JMP , -1
1247 7200          CLA
1250 1260          TAD K0212         /LINE FEED
1251 6046          TLS
1252 6041          TSF
1253 5252          JMP , -1
1254 6042          TCF
1255 7200          CLA
1256 5642          JMP I CRLFLF
1257 0215          <0215,          215
1260 0212          <0212,          212

```

/TYPE CONTENTS OF ADDRESS IN AC

```

1261 5261 TYCONT,      JMP ,
1262 3307      DCA TXSTOR      /SAVE ADDRESS
1263 1707      TAD I TXSTOR    /GET CONTENTS
1264 3307      DCA TXSTOR
1265 4242      JMS CRLFLF      /CARRIAGE RETURN - LINE FEED
1266 4273      JMS TYCOVR     /TYPE UPPER OCTAL
1267 4273      JMS TYCOVR
1270 4273      JMS TYCOVR
1271 4273      JMS TYCOVR
1272 5661      JMP I TYCONT
    
```

```

1273 5273 TYCOVR,      JMP ,
1274 1307      TAD TXSTOR
1275 7006      RTL
1276 7004      RAL
1277 3307      DCA TXSTOR
1300 1307      TAD TXSTOR
1301 7004      RAL
1302 0073      AND K0007
1303 1074      TAD K0020
1304 7040      CMA              /MAKE - FOR
1305 4230      JMS TYCHAR     /OUTPUT
1306 5673      JMP I TYCOVR
1307 0000 TXSTOR,      0
    
```

/STOP TAPE ON ERROR, LEAVE FLAGS SET

```

1310 5310 ERSTP,      JMP ,
1311 6761      IOT 761
1312 0075      AND K0200
1313 1076      TAD K0003
1314 6764      IOT 764
1315 5710      JMP I ERSTP
1316 5316 TSTATB,      JMP ,
1317 6772      IOT 772
1320 3331      DCA SBRECV
1321 1332      TAD SBRECV+1
1322 4451      JMS I TYP CON
1323 4456      JMS I TYPTX
1324 0063      63
1325 6441      6441
1326 6400      6400
1327 4277      4277
1330 5716      JMP I TSTATB
1331 0000 SBRECV,      0
1332 1331      .=1
    
```



```
*1400
/TYPE MOVE AND DIRECTION
1400 5200 TYMOVE,      JMP ,
1401 4311          JMS TYDRV
1402 4456          JMS I TYPTX
1403 7777          7777
1404 5557          5557
1405 6645          6645
1406 0077          0077
1407 4336          JMS TYDIR
1410 5600          JMP I TYMOVE

/TYPE SEARCH DIRECTION AND MODE
1411 5211 TYSRCH,      JMP ,
1412 4311          JMS TYDRV
1413 4456          JMS I TYPTX
1414 7777          7777
1415 6345          6345
1416 4162          4162
1417 4350          4350
1420 0077          0077
1421 4336          JMS TYDIR
1422 4363          JMS TYMODE
1423 5611          JMP I TYSRCH

/TYPE READ DATA DIRECTION AND MODE
1424 5224 TYRDAT,      JMP ,
1425 4311          JMS TYDRV
1426 4260          JMS TYREAD
1427 4275          JMS TYDATA
1430 4336          JMS TYDIR
1431 4363          JMS TYMODE
1432 5624          JMP I TYRDAT

/TYPE READ ALL DIRECTION AND MODE
1433 5233 TYRALL,      JMP ,
1434 4311          JMS TYDRV
1435 4260          JMS TYREAD
1436 4303          JMS TYALL
1437 4336          JMS TYDIR
1440 4363          JMS TYMODE
1441 5633          JMP I TYRALL
```

```
          /TYPE WRITE DATA DIRECTION AND MOJE
1442  5242  TYWDAT,      JMP ,
1443  4311          JMS TYDRV
1444  4266          JMS TYWRIT
1445  4275          JMS TYDATA
1446  4336          JMS TYDIR
1447  4363          JMS TYMODE
1450  5642          JMP I TYWDAT
```

```
          /TYPE WRITE ALL DIRECTION AND MODEL
1451  5251  TYWALL,      JMP ,
1452  4311          JMS TYDRV
1453  4266          JMS TYWRIT
1454  4303          JMS TYALL
1455  4336          JMS TYDIR
1456  4363          JMS TYMODE
1457  5651          JMP I TYWALL
```

```
          /TYPE READ
1460  5260  TYREAD,      JMP ,
1461  4456          JMS I TYPTX
1462  6245          6245
1463  4144          4144
1464  0077          0077
1465  5660          JMP I TYREAD
```

```
          /TYPE WRITE
1466  5266  TYWRIT,      JMP ,
1467  4456          JMS I TYPTX
1470  6762          6762
1471  5164          5164
1472  4500          4500
1473  7700          7700
1474  5666          JMP I TYWRIT
```

```
          /TYPE DATA
1475  5275  TYDATA,      JMP ,
1476  4456          JMS I TYPTX
1477  4441          4441
1500  6441          6441
1501  0077          0077
1502  5675          JMP I TYDATA
```

```

/TYPE ALL
1503 5303 TYALL, JMP .
1504 4456 JMS I TYPTX
1505 4154 4154
1506 5400 5400
1507 7700 7700
1510 5703 JMP I TYALL

```

```

/TYPE DRIVE AND NUMBER
1511 5311 TYDRV, JMP .
1512 4456 JMS I TYPTX
1513 7777 7777
1514 7777 7777
1515 4462 4462
1516 5166 5166
1517 4500 4500
1520 0077 0077
1521 1057 TAD CDRIVE
1522 7450 SNA
1523 1130 TAD K0010
1524 1335 TAD K260
1525 6046 TLS
1526 6041 TSF
1527 5326 JMP , -1
1530 7200 CLA
1531 4456 JMS I TYPTX

```

```

1532 7777 7777
1533 7700 7700
1534 5711 JMP I TYDRV
1535 0260 K260, 260

```

```

/TYPE FORWARDS OR BACKWARD
1536 5336 TYDIR, JMP .
1537 6761 IOT 761
1540 0102 AND K0400
1541 7650 SNA CLA
1542 5353 JMP TYFWD-2
1543 4345 JMS TYBKW
1544 5736 JMP I TYDIR
1545 5345 TYBKW, JMP .
1546 4456 JMS I TYPTX
1547 4253 4253
1550 6744 6744
1551 0077 0077
1552 5745 JMP I TYBKW

1553 4355 JMS TYFWD
1554 5736 JMP I TYDIR
1555 5355 TYFWD, JMP .
1556 4456 JMS I TYPTX
1557 4667 4667
1560 4400 4400
1561 7700 7700
1562 5755 JMP I TYFWD

```

```

/TYPE CONTINUOUS IF NOT NORMAL MOJE
1563 5363 TYMODE, JMP
1564 6761 IOT 761
1565 0123 AND K0100
1566 7650 SNA CLA
1567 5763 JMP I TYMODE
1570 4456 JMS I TYPTX
1571 4300 4300
1572 5557 5557
1573 4445 4445
1574 0077 0077
1575 5763 JMP I TYMODE
    
```

PAUSE

```

/TC01 EXERCISER - TAPE 2
/BASIC MOTION TEST, DECIPHER KEYBOARD INPUT
/SETUP EXECUTE TABLE, START DO LOOP
/ON FIRST CHARACTER IF A(D)
    
```

```

*2000
2000 4456 MVTEST, JMS I TYPTX
2001 7777 7777 /CR LF
2002 7777 7777 /CR LF
2003 7700 7700
2004 1032 TAD BF1LOC /SA OF EXECUTE TABLE
2005 3010 DCA 10 /FOR INDIRECTS:
2006 1010 TAD 10
2007 7001 IAC
2010 3011 DCA 11 /FOR RESET ON RPT LOOP
2011 7040 CMA
2012 3013 DCA 13 /FOR 1ST D
2013 1011 TAD 11
2014 3012 DCA 12 /FOR RESET ON CHNG DRIVES
2015 4240 JMS WAITIN /WAIT FOR INPJT
2016 4247 JMS MVEQUL
2017 0304 304 /1ST IN = D
2020 3013 MVREST, DCA 13 /0 TO CHAR COUNTER
2021 4247 JMS MVEQUL
2022 0306 306 /* F FORWARD
2023 4247 JMS MVEQUL
2024 0302 302 /* B BACKWARD
2025 4247 JMS MVEQUL
2026 0323 323 /* S STOP
2027 4247 JMS MVEQUL
2030 0303 303 /* C CHANGE
2031 4247 JMS MVEQUL
2032 0327 327 /* W WAIT
2033 4247 JMS MVEQUL
2034 0322 322 /* R REPEAT
2035 4247 JMS MVEQUL
2036 0304 304 /* D DO LOOP
2037 5200 JMP MVTEST
    
```

```

/WAIT KEYBOARD INPUT
2040 5240 WAITIN, JMP
2041 6031 KSF
2042 5241 JMP ,-1
    
```

2043 6036
2044 7041
2045 3014
2046 5640

KRB
CMA IAC
DCA 14
JMP I WAITIN

/14 = 2'S COMPLEMENT OF IN

```

/TEST FOR WHICH CHARACTER IN
2047 5247 MVEQUL, JMP ,
2050 1014 TAD 14
2051 1647 TAD I MVEQUL
2052 2247 ISZ MVEQUL
2053 7650 SNA CLA /RIGHT CHAR
2054 5260 JMP ,+4 /FOUND WHICH CHAR
2055 2013 ISZ 13 /INC POINTER
2056 7000 NOP
2057 5647 JMP I MVEQUL /TEST NEXT CHAR
2060 1013 TAD 13
2061 7040 CMA /FIRST IN = D
2062 7650 SNA CLA
2063 5712 JMP I DOTHEM /FIRST = D DO LAST
2064 1013 TAD 13
2065 7104 CLL RAL /NUMBER TIMES 2
2066 1273 TAD MVRTBL /FORM ADDRESS
2067 3014 DCA 14 /FOR INDIRECTS:
2070 1414 TAD I 14 /GET ROUTINE ADDRESS
2071 3410 DCA I 10 /TO EXECUTE TABLE
2072 5414 JMP I 14 /GO TO TYPEOUT

2073 2073 MVRTBL,
2074 2216 MVFWD /START TAPE FORWARD
2075 5313 JMP MVGFWD
2076 2223 MVBKWD /START BACKWARD
2077 5321 JMP MVGBKW
2100 2230 MVSTOP /STOP TAPE
2101 5327 JMP MVGSTP
2102 2274 MVCHNG /CHANGE DRIVES:
2103 5335 JMP MVGCHG
2104 2246 MVWAIT /WAIT, MARK TIME
2105 5347 JMP MVGWAT
2106 2257 MVRPT /REPEAT X TIMES
2107 5356 JMP MVGRPT
2110 2305 MVEND /END OF DO LOOP
2111 5712 JMP I ,+1
2112 2200 DOTHEM, DOLOOP /TO GET TO EXECUTE LOOP

/TYPE FORWARD
2113 4456 MVGFWD, JMS I TYPTX
2114 7777 7777
2115 4667 4667
2116 4477 4477
2117 4240 JMS WAITIN
2120 5220 JMP MVREST

```

```

/TYPE BACKWARD
2121 4456 MVGBKW,      JMS I TYPTX
2122 7777      7777
2123 4253      4253
2124 6744      6744
2125 7700      7700
2126 5317      JMP MVGFWD+4

/TYPE STOP
2127 4456 MVGSTP,      JMS I TYPTX
2130 7777      7777
2131 6364      6364
2132 5760      5760
2133 7700      7700
2134 5317      JMP MVGFWD+4

/TYPE CHANGE
2135 4456 MVGCHG,      JMS I TYPTX
2136 7777      7777
2137 4350      4350
2140 4777      4777
2141 1012      TAD 12
2142 3410      DCA I 10      /PUT WHERE TO REPEAT
2143 1010      TAD 10      /FROM ON CHANGING DRIVES
2144 7001      IAC
2145 3012      DCA 12
2146 5317      JMP MVGFWD+4

/TYPE WAIT
2147 4456 MVGWAT,      JMS I TYPTX
2150 7777      7777
2151 6741      6741
2152 5164      5164
2153 0077      0077
2154 4772      JMS I GET4IN      /GET WAIT CONSTANT
2155 5317      JMP MVGFWD+4

/TYPE REPEAT
2156 4456 MVGRPT,      JMS I TYPTX
2157 7777      7777
2160 6260      6260
2161 6400      6400
2162 0077      0077
2163 4772      JMS I GET4IN      /GET REPEAT CONSTANT
2164 1011      TAD 11      /GET REPEAT FROM ADDRESS
2165 3410      DCA I 10      /TO EXECUTE TABLE
2166 1010      TAD 10
2167 7001      IAC
2170 3011      DCA 11      /FOR NEXT REPEAT
2171 5317      JMP MVGFWD+4

2172 2310 GET4IN,      GETMIN      /TO GET 4 CHARACTERS
    
```

```

/DO LOOP, EXECUTE SELECTED SEQUENCE
*2200
2200 4456 JOLOOP,      JMS I TYPTX  /TYPE DO
2201 7777          7777
2202 4457          4457
2203 7777          7777
2204 7700          7700
2205 1032         TAD BF1LOC          /SET ROUTINE POINTER TO START
2206 3010         DCA 10
2207 3011         DCA 11          /TO COUNT RPTS:
2210 1410         TAD I 10        /GET ROUTINE ADDRESS
2211 3214         DCA ,+3        /FOR JMS I
2212 4614         JMS I ,+2      /EXECUTE ROUTINE
2213 5210         JMP , -3       /DO NEXT
2214 2214         .
2215 5215         JMP .
    
```

```

/BASIC MOTION TEST
/FORWARD, BACKWARD, STOP, WAIT
/REPEAT AND CHANGE DRIVE ROUTINES
    
```

```

/START FORWARD MOTION
2216 5216         MVFWD,        JMP .
2217 1075         TAD K0200      /MOVE TAPE FORWARD
2220 1060         TAD UNFUNC     /* DRIVE NUMBER
2221 6766         IOT 766
2222 5616         JMP I MVFWD
    
```

```

/START BACKWARD MOTION
2223 5223         MVBKWD,       JMP .
2224 1344         TAD K0600      /MOVE BACKWARD
2225 1060         TAD UNFUNC     /* DRIVE NUMBER
2226 6766         IOT 766
2227 5623         JMP I MVBKWD
    
```

```

/STOP TAPE
2230 5230         MVSTOP,       JMP .
2231 6761         IOT 761        /READ STAT A.
2232 0345         AND K7000      /CLEAR TO DRIVE NUM
2233 7041         CMA IAC
2234 1060         TAD UNFUNC
2235 7640         SZA CLA        /SAME DRIVE
2236 5243         JMP ,+5        /NOT SAME AS STAT A
2237 6761         IOT 761
2240 0075         AND K0200      /CLEAR IAC TO MOTION BIT
2241 6764         IOT 764        /CLEAR MOTION IF NOT READY
2242 5630         JMP I MVSTOP   /EXIT
2243 1060         TAD UNFUNC
2244 6766         IOT 766
2245 5630         JMP I MVSTOP
    
```



```

/ WAIT AND DO NOTHING FOR A NUMBER OF BLOCKS
2246 5246 MVWAIT,      JMP .
2247 3016      DCA 16
2250 1410      TAD I 10
2251 3017      DCA 17
2252 2016      ISZ 16
2253 5252      JMP , -1
2254 2017      ISZ 17
2255 5252      JMP , -3
2256 5646      JMP I MVWAIT

/ REPEAT X TIMES LOOP
2257 5257 MVRPT,      JMP .
2260 2011      ISZ 11      /RPT COUNT+1
2261 1011      TAD 11
2262 1410      TAD I 10      /* (-RPT CONSTANT)
2263 7640      SZA CLA      /DONE X TIMES
2264 5270      JMP , +4      /NO
2265 0410      AND I 10      /INC EXECUTE ADDRESS
2266 3011      DCA 11
2267 5657      JMP I MVRPT      /EXIT
2270 7040      CMA      /RESET EXECUTE ADDRESS
2271 1410      TAD I 10
2272 3010      DCA 10
2273 5657      JMP I MVRPT

/ CHANGE DRIVES AND REPEAT OR
/ IF BACK TO FIRST DRIVE CONTINUE
2274 5274 MVCHNG,      JMP .
2275 4437      JMS I NEWDRV
2276 5301      JMP , +3
2277 0410      AND I 10
2300 5674      JMP I MVCHNG
2301 7040      CMA
2302 1410      TAD I 10
2303 3010      DCA 10
2304 5674      JMP I MVCHNG

/ END OF ROUTINE TYPED IN
2305 5305 MVEND,      JMP .
2306 5707      JMP I , +1
2307 2000      MVTEST

```

```

/MAKE - CONSTANT OF 4 INPUTS
2310 5310 GETMIN, JMP ,
2311 7201 CLA IAC /*+1 WHEN L=1 GOT 4
2312 3013 DCA 13
2313 6031 KSF /*WAIT KEYBOARD
2314 5313 JMP ,-1
2315 6036 KRB /*READ KEYBOARD
2316 6046 TLS /*OUTPUT
2317 6041 TSF
2320 5317 JMP ,-1
2321 7040 CMA /*MAKE -
2322 0073 AND K0007 /*OCTAL
2323 3014 DCA 14
2324 1013 TAD 13 /*MOVE 1 CHARACTER
2325 7104 CLL RAL
2326 7006 RTL
2327 1014 TAD 14 /*ASSEMBLE
2330 3013 DCA 13
2331 7420 SNL /*SKIP = DONE 4
2332 5313 JMP GETMIN+3 /*NOT 4 YET
2333 1013 TAD 13
2334 7001 IAC /*MAKE 2'S COMPLEMENT
2335 7450 SNA
2336 7040 CMA /*MUST BE AT LEAST -1
2337 3410 DCA I 10 /*TO EXECUTE TABLE
2340 4456 JMS I TYPTEX
2341 7777 7777
2342 7700 7700
2343 5710 JMP I GETMIN
2344 0600 K0600, 600
2345 7000 K7000, 7000

```

/BASIC SEARCH ROUTINE 1
 /FORCE TAPE INTO END ZONE
 /FAR ENOUGH TO GUARANTEE BLOCK 0 FORWARD
 /VERIFY BLOCKS 0000 TO 2701 THEN REVERSE

```

*2400
2400 4436 SRCH1,      JMS I REWIND
2401 4436          JMS I REWIND      /INTO EZ TWICE
2402 3114          DCA DIRFLG
2403 3020          DCA RECORD      /RECORD 0
2404 4435          JMS I SRCHIT     /FIND IT
2405 5221          JMP SCH1ST      /YES
2406 4444          JMS I ERRSTP     /NO
2407 4452          JMS I SRCHTY
2410 1026          TAD RECRDK      /BLOCK LOOKED FOR
2411 4451          JMS I TYPCON
2412 1025          TAD IDCON      /BLOCK FOUND
2413 4451          JMS I TYPCON
2414 6772          IOT 772        /RD STAT B
2415 3021          DCA BLKFND
2416 1025          TAD IDCON
2417 4451          JMS I TYPCON     /TYPE STAT B
2420 5202          JMP SRCH1+2

2421 3422 SCH1ST,   DCA I POSITN     COND 0
2422 3423          DCA I DIRECT     /FORWARD
2423 2020          ISZ RECORD      /RECORD LOOKED FOR +1
2424 7040          CMA
2425 3427          DCA I WCLOC     /SET WC TO -1
2426 6764          IOT 764        /SET WC ENABLES
2427 4440          JMS I WAITI
2430 0001          1
2431 7410          SKP
2432 5244          JMP SCH1OK      /INTERRUPT OK
2433 6772 SCH1ER,   IOT 772
2434 0110          AND EZBIT
2435 7650          SMA CLA        /END ZONE
2436 5206          JMP SRCH1+6 /NO TYPE ERROR
2437 1020          TAD RECORD
2440 1077          TAD K5077
2441 7700          SMA CLA        /DONE ALL BLOCKS
2442 5261          JMP SCH1ND     /YES
2443 5206          JMP SRCH1+6 /NO, ERROR

2444 1021 SCH1OK,   TAD BLKFND      /GET BLOCK NUMBER READ
2445 7041          CMA IAC        /MAKE -
2446 1020          TAD RECORD
2447 7640          SZA CLA        /RIGHT BLOCK
2450 5253          JMP ,+3        /NO
2451 2422          ISZ I POSITN    /NEW POSITION
2452 5223          JMP SCH1ST+2    /TEST NEXT BLOCK
2453 4365          JMS SIERRO

```

```

2454 2422      ISZ I POSITN
2455 2020      ISZ RECORD
2456 4435      JMS I SRCHIT      /RESYNC ON NEXT BLOCK
2457 5244      JMP SCH10K          /FOUND OK
2460 5233      JMP SCH1ER        /TEST FOR END ZONE
2461 4437      SCH1ND, JMS I NEWDRV
2462 5202      JMP SRCH1+2    /RPT NEXT DRIVE
2463 7040      CMA
2464 3020      DCA RECORD
2465 1125      TAD K0204
2466 1060      TAD UNFUNC
2467 6766      IOT 766      /MOVE DRIVE INTO EZ
2470 4440      JMS I WAITI
2471 5001      5001
2472 7000      NOP
2473 4437      JMS I NEWDRV
2474 5265      JMP SCH1ND+4
2475 2020      ISZ RECORD      /DONE ALL TWICE
2476 7410      SKP          /YES

2477 5265      JMP SCH1ND+4    /MAKE 2ND MOVE INTO EZ
2500 1100      SRCH2, TAD K2701
2501 3020      DCA RECORD
2502 1101      TAD K0614      /SRCH BACKWARD
2503 1060      TAD UNFUNC      /* DRIVE NUMBER
2504 6766      IOT 766
2505 7040      CMA
2506 3427      DCA I WCLOC
2507 4440      JMS I WAITI
2510 0001      1
2511 7410      SKP
2512 5325      JMP SCH20K
2513 4365      SCH2ER, JMS SIERRO
2514 7240      CLA CMA
2515 1020      TAD RECORD
2516 3020      DCA RECORD
2517 1020      TAD RECORD
2520 7710      SPA CLA
2521 5356      JMP SIERRO-7
2522 7040      CMA
2523 3114      DCA DIRFLG
2524 4435      JMS I SRCHIT

```

```

2525 1020 SCH20K, TAD RECORD /BLOCK LOOKED FOR
2526 7041 CMA IAC /MAKE -
2527 1021 TAD BLKFND /* BLOCK FOJND
2530 7640 SZA CLA /RIGHT ONE
2531 5313 JMP SCH2ER /NO
2532 6764 IOT 764 /RESET WC ENA3LE
2533 7040 CMA
2534 3427 DCA I WCLOC
2535 1020 TAD RECORD /LAST BLOCK -1
2536 3422 DCA I POSITN
2537 7040 CMA
2540 1020 TAD RECORD
2541 3020 DCA RECORD
2542 4440 JMS I WAITI
2543 0001 1
2544 7410 SKP
2545 5325 JMP SCH20K /STATUS WAS NORMAL
2546 6772 IOT 772 /READ STATB
2547 0110 AND EZBIT
2550 7650 SNA CLA /EZ INT
2551 5313 JMP SCH2ER /NO, ERROR
2552 1020 TAD RECORD
2553 7040 CMA
2554 7640 SZA CLA /EXPECT END ZONE
2555 5313 JMP SCH2ER /NO, ERROR
2556 4437 JMS I NEWDRV /GET NEXT DRIVE
2557 5300 JMP SRCH2 /REPEAT
2560 7604 CLA OSR
2561 7010 RAR
2562 7620 SNL CLA /REPEAT TEST
2563 7402 HLT /NO, HALT
2564 5200 JMP SRCH1 /START OVER
2565 5365 SIERRO, JMP ,
2566 4444 JMS I ERRSTP
2567 4452 JMS I SRCHTY
2570 1026 TAD RECRDK
2571 4451 JMS I TYPCON
2572 1025 TAD IDCON
2573 4451 JMS I TYPCON
2574 4447 JMS I SBTYPE
2575 5765 JMP I SIERRO

```

/SEARCH ROUTINE 2
 /READ A SERIES OF 129 BLOCKS OR UNTIL END ZONE
 /COMPARE FOR INCREMENTING OR DEC.

/1ST KEY (F) SEARCH FORWARD
 /ALL OTHERS BACKWARD
 /2ND KEY (C) COMPARE BLOCKS
 /ALL OTHERS DUMP TO 129 ON PRINTER.

```

*2600
2600 4743  SERCH2,      JMS I DRVTYP  /TYPE DRIVE AND NUMBER
2601 4333      JMS TYINTX   /TYPE TYPE IN
2602 4456      JMS I TYPTX   /F FOR FWD
2603 0046      0046
2604 0046      0046
2605 5762      5762
2606 0046      0046
2607 6744      6744
2610 7700      7700
2611 4321      JMS OTHRTX   /TYPE ALL OTHERS
2612 4456      JMS I TYPTX
2613 4253      4253       /BACKWARD
2614 6744      6744
2615 0077      0077
2616 6031      KSF         /WAIT K
2617 5216      JMP , -1
2620 6036      KRB
2621 6046      TLS
2622 6041      TSF
2623 5222      JMP , -1
2624 6042      TCF
2625 1273      TAD K7472
2626 7640      SZA CLA      /#F
2627 1102      TAD K0400    /NO PREP BACKWARD
2630 1103      TAD K0214

2631 1060      TAD UNFUNC
2632 6766      IOT 766      /START SEARCH
2633 1032      TAD BF1LOC
2634 3010      DCA 10
2635 1064      TAD K7577
2636 3011      DCA 11
2637 7040      SER2ST,    CMA      /WC = -1
2640 3427      DCA I WCLOC
2641 1025      TAD IDCON    /TO BLKFND
2642 3430      DCA I CALOC
2643 4440      JMS I WAITI
2644 0001      1
2645 5255      JMP SEREZ      /NO NORMAL INT
  
```

2646	1021	TAD	BLKFND		
2647	3410	DCA	I 10		/STORE BLOCK NUMBER
2650	2011	ISZ	11		
2651	7410	SKP			
2652	5720	JMP	I SER2AI	/DONE 129	
2653	6764	IOT	764	/RESET WCE	
2654	5237	JMP	SER2ST	/REPT	
2655	6772	SEREZ,	IOT 772		/READ B
2656	0110	AND	EZBIT		
2657	7640	SZA	CLA	/END ZONE	
2660	5274	JMP	SER2NZ	/YES	
2661	6772	IOT	772		
2662	3020	DCA	RECORD		/TYPE STATUS B
2663	1026	TAD	RECRDK		
2664	4451	JMS	I TYPCON		
2665	4456	JMS	I TYPTX		
2666	0063		0063		
2667	6441		6441		
2670	6400		6400		
2671	4277		4277		
2672	5200	JMP	SERCH2	/START OVER	
2673	7472	K7472,	7472		

```

/TYPE END ZONE
2674 4456 SER2NZ,      JMS I TYPTX
2675 7777          7777
2676 4556          4556
2677 4400          4400
2700 7257          7257
2701 5645          5645
2702 7700          7700
2703 1011          TAD 11
2704 7041          CMA IAC
2705 1064          TAD K7577
2706 7640          SZA CLA      /READ ANY AT ALL
2707 5720          JMP I SER2AI  /YES
2710 4456          JMS I TYPTX  /TYPE NO BLOCKS
2711 7777          7777
2712 5657          5657
2713 0042          0042
2714 5457          5457
2715 4353          4353
2716 6377          6377
2717 5200          JMP SERCH2
2720 3000 SER2AI,   SER2A      /TO GET TO REST OF PROGRAM

```

```

/TYPE ALL OTHERS
2721 5321 OTHRTX,    JMP .
2722 4456      JMS I TYPTX
2723 7777      7777
2724 4154      4154
2725 5400      5400
2726 5764      5764
2727 5045      5045
2730 6263      6263
2731 0077      0077
2732 5721      JMP I OTHRTX

```

```

/TYPE TYPE IN
2733 5333 TYINTX,    JMP .
2734 4456      JMS I TYPTX
2735 7777      7777
2736 6471      6471
2737 6045      6045
2740 0051      0051
2741 5677      5677
2742 5733      JMP I TYINTX
2743 1511 DRV TYP,   TYDRV

```


/REST OF SEARCH ROUTINE 2

```

*3000
SER2A,      IOT 761
3000  6761
3001  0075      AND K0200
3002  6764      IOT 764          /STOP TAPE
3003  1011      TAD 11
3004  7041      CMA IAC
3005  1064      TAD K7577
3006  3011      DCA 11          /NUMBER OF BLOCKS READ
3007  1032      TAD BF1LOC
3010  3010      DCA 10          /TO GET FIRST BLOCK
3011  1010      TAD 10
3012  7001      IAC
3013  3012      DCA 12          /TO GET 2ND BLOCK
3014  1012      TAD 12
3015  4451      JMS I TYPCON    /TYPE FIRST
3016  4456      JMS I TYPTX     /BLOCK NO READ
3017  0046      0046
3020  5162      5162
3021  6364      6364
3022  7700      7700
3023  4625      JMS I ,+2       /TYPE TYPE IN
3024  7410      SKP
3025  2733      TYINTX
3026  4456      JMS I TYPTX     /C FOR COMPARE
3027  0043      0043
3030  0046      0046
3031  5762      5762
3032  0043      0043
3033  5755      5755
3034  6041      6041
3035  6245      6245
3036  7700      7700
3037  4641      JMS I ,+2       /ALL OTHERS
3040  74410     SKP
3041  2721      OTHRTX
3042  4456      JMS I TYPTX     /PRINT
3043  6062      6062
3044  5156      5156
3045  6400      6400
3046  7700      7700
3047  6031      KSF            /WAIT KEY
3050  5247      JMP , -1
3051  6036      KRB
3052  6046      TLS
3053  6041      TSF
3054  5253      JMP , -1
3055  6042      TCF
3056  1331      TAD K7475
3057  7640      SZA CLA        /=C
3060  5323      JMP SER2TY     /NO, PRINT ALL

```

```

3061 6761 IOT 761
3062 0102 AND K0400
3063 7640 SZA CLA /FORWARDS
3064 7040 CMA /NO MAKE -1
3065 7450 SNA /BACKWARDS
3066 7001 IAC /NO MAKE +1
3067 3014 DCA 14

/COMPARE BLOCKS FOR INCREMENTING JR DEC
3070 2011 SERCMP, ISZ 11 /COMPARED ALL
3071 7410 SKP /NO
3072 5311 JMP SER2LS /TYPE LAST BLOCK
3073 1014 TAD 14 /* OR -
3074 1410 TAD I 10 /* FIRST BLOCK
3075 7041 CMA IAC /MAKE -
3076 1412 TAD I 12 /* NEXT BLOCK
3077 7650 SNA CLA /SHOULD BE 0
3100 5270 JMP SERCMP /DO NEXT
3101 4456 JMS I TYPTX
3102 7777 7777
3103 7700 7700
3104 1010 TAD 10
3105 4451 JMS I TYPON /TYPE 1ST
3106 1012 TAD 12
3107 4451 JMS I TYPON /TYPE 2ND
3110 5270 JMP SERCMP /DO NEXT

/TYPE OUT LAST BLOCK READ
3111 1012 SER2LS, TAD 12
3112 4451 JMS I TYPON /TYPE BLOCK NUMBER
3113 4456 JMS I TYPTX
3114 0054 0054 /LAST
3115 4163 4163
3116 6477 6477
3117 4437 JMS I NEWDRV
3120 7000 NOP
3121 5722 JMP I ,+1
3122 2600 SERCH2 /REPEAT FOR NEXT DRIVE

/PRINT ALL BLOCKS READ
3123 1012 SER2TY, TAD 12 /ADDRESS
3124 4451 JMS I TYPON /TYPE BLOCK NUMBER
3125 2012 ISZ 12
3126 2011 ISZ 11 /DONE ALL
3127 5323 JMP SER2TY /NO
3130 5313 JMP SER2LS+2 /YES, DO NEXT DRV
3131 7475 <7475, 7475

```

PAUSE

```

/TC01 BASIC EXERCISER TAPE 3
/START STOP TURN AROUND TEST
/1 TO 8 DRIVES IN ANY COMBINATION
/TESTS TA ON BLOCK 0 BOTH DIRECTIONS
/S/S/TA LENGTH OF TAPE AND TA ON BLOCK 2701
/MOVE ALL DRIVES INTO REVERSE END ZONE

```

*3200

```

3200 4436 SSTRNA,      JMS I REWIND
3201 4436      JMS I REWIND      /MOVE INTO EZ 2
3202 4441      JMS I WTHALF     /WAIT HALF SECOND
3203 4443      JMS I SAVPAD

      /TEST TURN AROUND ON BLOCK 0 FIND 1 FWD
3204 7201      CLA IAC
3205 3020      DCA RECORD
3206 4454      JMS I SSTFWD     /FIND BLOCK 1 FWD
3207 5214      JMP GBKW1
3210 4366      SSTER1,        JMS EZERR
3211 7000      NOP
3212 4453      JMS I ERSSTA
3213 5204      JMP , -7

      /TURN AROUND FIND 0 BACKWARDS
3214 3020      GBKW1,         DCA RECORD
3215 7001      IAC
3216 3422      DCA I POSITN
3217 3423      DCA I DIRECT
3220 4455      JMS I SSTBKW
3221 5225      JMP , +4
3222 5210      JMP SSTER1
3223 5211      JMP SSTER1+1

      /WAIT FOR EZ TA FIND 0 FWD
3224 5211      JMP SSTER1+1
3225 3422      DCA I POSITN
3226 7040      CMA
3227 3423      DCA I DIRECT
3230 1065      TAD K0604
3231 1060      TAD UNFUNC
3232 6766      IOT 766
3233 4440      JMS I WAITI
3234 5001      5001
3235 5210      JMP SSTER1
3236 7040      CMA
3237 3422      DCA I POSITN
3240 4454      JMS I SSTFWD
3241 5245      JMP , +4
3242 5210      JMP SSTER1
3243 7000      NOP
3244 5211      JMP SSTER1+1

```

```

3245 3422 DCA I POSITN
3246 3423 DCA I DIRECT
3247 1075 TAD K0200
3250 6764 IOT 764
3251 4437 JMS I NEWDRV
3252 5204 JMP SSTRNA+4
3253 4441 JMS I WTHALF

```

```

/TEST FORWARD START AFTER FORWARD

```

```

3254 4443 SSTAN1, JMS I SAVPAD
3255 1422 TAD I POSITN
3256 1104 TAD SFAFK
3257 3020 DCA RECORD
3260 4454 JMS I SSTFWD
3261 5267 JMP SSTA1A
3262 5666 JMP I .+4
3263 7000 NOP
3264 4453 JMS I ERSSTA
3265 5254 JMP SSTAN1
3266 3437 SSTAEZ

```

```

3267 1020 SSTA1A, TAD RECORD
3270 3422 DCA I POSITN
3271 3423 DCA I DIRECT
3272 1075 TAD K0200
3273 6764 IOT 764
3274 4437 JMS I NEWDRV
3275 5254 JMP SSTAN1
3276 4441 JMS I WTHALF

```

```

/TEST BACKWARD START AFTER FORWARD

```

```

3277 4443 SSTAN2, JMS I SAVPAD
3300 1422 TAD I POSITN
3301 1107 TAD SBAFK
3302 3020 DCA RECORD
3303 4455 JMS I SSTBKW
3304 5311 JMP SSTA2A
3305 4366 JMS EZERR
3306 7000 NOP
3307 4453 JMS I ERSSTA
3310 5277 JMP SSTAN2
3311 1020 SSTA2A, TAD RECORD
3312 3422 DCA I POSITN
3313 7040 CMA
3314 3423 DCA I DIRECT
3315 1075 TAD K0200
3316 6764 IOT 764
3317 4437 JMS I NEWDRV
3320 5277 JMP SSTAN2
3321 4441 JMS I WTHALF

```

```

/TEST START FORWARD AFTER BACKWARD)
3322 4443 SSTAN3,      JMS I SAVPAD
3323 1422      TAD I POSITN
3324 1105      TAD SFABK
3325 3020      DCA RECORD
3326 4454      JMS I SSTFWD
3327 5334      JMP SSTA3A
3330 4366      JMS EZERR
3331 7000      NOP

```

```

/THEN TEST FORWARD TO BACKWARD TURN AROUND
3332 4453      JMS I ERSSTA
3333 5322      JMP SSTAN3
3334 1020      SSTA3A,      TAD RECORD
3335 3422      DCA I POSITN
3336 3423      DCA I DIRECT
3337 7040      CMA
3340 3427      DCA I WCLOC
3341 6764      IOT 764
3342 4440      JMS I WAITI
3343 5001      5001
3344 5331      JMP SSTAN3+7
3345 2422      ISZ I POSITN
3346 4455      JMS I SSTBKW
3347 5353      JMP SSTA3B
3350 5330      JMP SSTAN3+6
3351 7000      NOP
3352 5331      JMP SSTAN3+7
3353 1020      SSTA3B,      TAD RECORD
3354 3422      DCA I POSITN
3355 7040      CMA
3356 3423      DCA I DIRECT
3357 1075      TAD K0200
3360 6764      IOT 764
3361 4437      JMS I NEWDRV
3362 5322      JMP SSTAN3
3363 4441      JMS I WTHALF
3364 5765      JMP I ,+1
3365 3400      SSTAN4
3366 5366      EZERR,      JMP
3367 7240      CMA CLA
3370 3021      DCA BLKFND
3371 5766      JMP I EZERR

```

*3400

```

/TEST BACKWARD START AFTER BACKWARDS
3400 4443 SSTAN4,      JMS I SAVPAD
3401 1422      TAD I POSITN
3402 1106      TAD SBABK
3403 3020      DCA RECORD
3404 4455      JMS I SSTBKW
3405 5212      JMP SSTA4A
3406 4636      JMS I EZERR
3407 7000      NOP

```

```
      /THEN TEST BACKWARD TO FORWARD TURN AROUND
3410  4453      JMS I ERSSTA
3411  5200      JMP SSTAN4
3412  1020      SSTA4A,      TAD RECORD
3413  3422      DCA I POSITN
3414  7040      CMA
3415  3423      DCA I DIRECT
3416  2020      ISZ RECORD
3417  4454      JMS I SSTFWD
3420  5224      JMP SSTA4B

3421  5206      JMP SSTAN4+6
3422  7000      NOP
3423  5207      JMP SSTAN4+7
3424  1020      SSTA4B,      TAD RECORD
3425  3422      DCA I POSITN
3426  3423      DCA I DIRECT
3427  1075      TAD K0200
3430  6764      IOT 764
3431  4437      JMS I NEWDRV
3432  5200      JMP SSTAN4
3433  4441      JMS I WTHALF
3434  5635      JMP I ,+1
3435  3254      SSTAN1

3436  3366      EZERRA,      EZERR
```

/END ZONE HAS BEEN REACHED FWD
/TEST TURN AROUND ON 2701

```

3437 1020 SSTAEZ, TAD RECORD
3440 1077 TAD K5077
3441 7500 SMA /DONE TO END ZONE
3442 5245 JMP ,+3 /YES
3443 5644 JMP I ,+1 /FALSE END ZONE
3444 3264 SSTAN1+10
3445 4437 JMS I NEWDRV /RESET TO FIRST DRV
3446 5245 JMP , -1

```

/MOVE ALL DRIVES INTO END ZONE

```

3447 7040 CMA
3450 3020 DCA RECORD /TO COUNT TWO EZ PASSES
3451 1125 TAD K0204
3452 1060 TAD UNFUNC
3453 6766 IOT 766
3454 4440 JMS I WAITI
3455 5001 5001
3456 7000 NOP /IGNORE OTHER INTERRUPTS
3457 7040 CMA
3460 3422 DCA I POSITN
3461 3423 DCA I DIRECT
3462 4437 JMS I NEWDRV
3463 5254 JMP , -7
3464 2020 ISZ RECORD /2 PASSES
3465 7410 SKP /YES
3466 5263 JMP , -3
3467 4441 JMS I WTHALF
3470 1340 SSTEZA, TAD K2700
3471 3020 DCA RECORD
3472 4443 JMS I SAVPAD
3473 4455 JMS I SSTBKW /GO BACKWARD TO 2700

```

```

3474 5301 JMP SSTEZ1 /REACHED 2700 OK
3475 7000 NOP
3476 7000 NOP
3477 4453 JMS I ERSSTA
3500 5270 JMP SSTEZA
3501 1020 SSTEZ1, TAD RECORD
3502 3422 DCA I POSITN /BLOCK 2700
3503 7040 CMA /BACKWARD
3504 3423 DCA I DIRECT
3505 2020 ISZ RECORD /2700 TO 2701
3506 4454 JMS I SSTFWD /FIND 2701 FORWARD
3507 5313 JMP ,+4 /OK

```

3510	4636	JMS I EZERRA	/EZ INT. ERROR
3511	7000	NOP	
3512	5274	JMP SSTEZA+4	
3513	1125	TAD K0204	
3514	1060	TAD UNFUNC	
3515	6766	IOT 766	
3516	4440	JMS I WAITI	/WAIT FOR EZ
3517	5001	5001	
3520	7000	NOP	
3521	4455	JMS I SSTBKW	/FIND 2701 BACKWARD
3522	5326	JMP ,+4	/OK
3523	7000	NOP	
3524	7000	NOP	
3525	5274	JMP SSTEZA+4	
3526	1075	TAD K0200	
3527	6764	IOT 764	
3530	4437	JMS I NEWDRV	/TESTED ALL DRIVES
3531	5335	JMP ,+4	/NO
3532	7604	CLA OSR	
3533	7006	RTL	
3534	7420	SNL	/DELETE END OF TEST HALT
3535	7402	HLT	/HLT END OF TEST
3536	5737	JMP I ,+1	/REPEAT TEST
3537	3201	SSTRNA+1	
3540	2700	K2700, 2700	

/START STOP TURN AROUND TEST
/SEARCH FORWARD ROUTINE

```

*3600
3600 5200 SSTA FW,      JMP .
3601 7240      CLA CMA
3602 3370      DCA BLOCKK
3603 1103      TAD K0214
3604 1060      TAD UNFUNC
3605 6766      IOT 766          /START TAPE FORWARD
3606 7040      CMA
3607 3427      DCA I WCLOC        /SET WC = -1
3610 1025      TAD IDCON
3611 3430      DCA I CALOC
3612 4440      JMS I WAITI
3613 0001      1
3614 5232      JMP SSTFSE          /STATUS B ERROR
3615 2370      ISZ BLOCKK
3616 5242      JMP SSTFBE+2
3617 1021      SSTFR,      TAD BLKFND
3620 3371      DCA BLOCKK+1
3621 1021      TAD BLKFND
3622 7041      CMA IAC
3623 1020      TAD RECORD
3624 7650      SNA CLA          /RIGHT BLOCK
3625 5600      JMP I SSTA FW      /YES, EXIT
3626 7710      SPA CLA          /BLOCK FOUND LESSER
3627 5240      JMP SSTFBE          /NO, ERROR
3630 6764      IOT 764          /RESET WC ENABLE
3631 5206      JMP SSTA FW+6

3632 6772      SSTFBE,      IOT 772
3633 0110      AND EZBIT
3634 7650      SNA CLA
3635 2200      ISZ SSTA FW      /END ZONE EXIT
3636 2200      ISZ SSTA FW      /ONLY 1 ISZ
3637 5600      JMP I SSTA FW
3640 2200      SSTFBE,      ISZ SSTA FW
3641 5235      JMP , -4
3642 1371      TAD BLOCKK+1
3643 7040      CMA
3644 1021      TAD BLKFND
3645 7650      SNA CLA
3646 5217      JMP SSTFR
3647 4322      JMS BNOTCN
3650 5201      JMP SSTA FW+1
    
```

/START STOP TURN AROUND TEST
/SEARCH BACKWARD ROUTINE

3651	5251	SSTABW,	JMP .	
3652	7240	CLA	CMA	
3653	3370	DCA	BLOCKK	
3654	1101	TAD	K0614	
3655	1060	TAD	UNFUNC	
3656	6766	IOT	766	
3657	7040	CMA		
3660	3427	DCA	I WCLOC	
3661	1025	TAD	IDCON	
3662	3430	DCA	I CALOC	
3663	4440	JMS	I WAITI	
3664	0001	1		
3665	5303	JMP	SSTBSE	
3666	2370	ISZ	BLOCKK	
3667	5313	JMP	SSTBBE+2	
3670	1021	SSTBR,	TAD BLKFND	
3671	3371	DCA	BLOCKK+1	
3672	1020	TAD	RECORD	
3673	7041	CMA	IAC	
3674	1021	TAD	BLKFND	
3675	7450	SNA		
3676	5651	JMP	I SSTABW	/FOUND BLOCK EXIT
3677	7710	SPA	CLA	
3700	5311	JMP	SSTBBE	
3701	6764	IOT	764	
3702	5257	JMP	SSTABW+6	
3703	6772	SSTBSE,	IOT 772	
3704	0110	AND	EZBIT	
3705	7650	SNA	CLA	
3706	2251	ISZ	SSTABW	
3707	2251	ISZ	SSTABW	
3710	5651	JMP	I SSTABW	
3711	2251	SSTBBE,	ISZ SSTABW	
3712	5306	JMP	,-4	
3713	1021	TAD	BLKFND	
3714	7040	CMA		
3715	1371	TAD	BLOCKK+1	
3716	7650	SNA	CLA	
3717	5270	JMP	SSTBR	
3720	4322	JMS	BNOTCN	
3721	5252	JMP	SSTABW+1	

/BLOCK NUMBERS ARE NOT
/CONSECUTIVE ON START UP OR TURV AROUND

3722	5322	BNOTCN, JMP ,	
3723	4444	JMS I ERRSTP	
3724	4452	JMS I SRCHTY	
3725	1026	TAD RECRDK	
3726	4451	JMS I TYPCON	
3727	4456	JMS I TYPTX	
3730	7777	7777	/TYPE BLOCK NJMBER ERR.
3731	4254	4254	
3732	5300	5300	
3733	0300	0300	
3734	4562	4562	
3735	6277	6277	
3736	1372	TAD BLOCKK+2	
3737	4451	JMS I TYPCON	
3740	4456	JMS I TYPTX	/TYPE LAST
3741	0054	0054	
3742	4163	4163	
3743	6477	6477	
3744	1025	TAD IDCON	
3745	4451	JMS I TYPCON	/TYPE CURRENT
3746	4456	JMS I TYPTX	/BLOCK NUMBER
3747	0064	64	
3750	5051	5051	
3751	6377	6377	
3752	1370	TAD BLOCKK	
3753	7001	IAC	
3754	3370	DCA BLOCKK	
3755	1373	TAD BLOCKK+3	
3756	4451	JMS I TYPCON	/TYPE OUT BLOCK
3757	4456	JMS I TYPTX	/COUNTER
3760	0043	43	
3761	5664	5664	
3762	6277	6277	
3763	7604	CLA OSR	
3764	7012	RTR	
3765	7620	SNL CLA	/STOP ON ERROR
3766	7402	HLT	/YES
3767	5722	JMP I BNOTCN	
3770	0000	BLOCKK, 0	
3771	0000	0	
3772	3771	.-1	
3773	3770	.-3	

/START STOP TURN AROUND TEST
 /ERROR TYPE OUT AND RESYNC ROUTINE
 *4000

```

4000 5200 SSTAER,      JMP ,
4001 4444      JMS I ERRSTP      /STOP TAPE LB STATB
4002 4452      JMS I SRCHTY      /TYPE SEARCH
4003 1422      TAD I POSITN
4004 7040      CMA
4005 7640      SZA CLA          /TAPE WAS WHERE
4006 5211      JMP ,+3          /NOT END ZONE
4007 4267      JMS EZTYPE          /TYPE END ZONE
4010 5213      JMP ,+3
4011 1022      TAD POSITN
4012 4451      JMS I TYP CON      /TYPE LAST BLOCK
4013 1423      TAD I DIRECT
4014 7640      SZA CLA          /DIRECTION WAS:
4015 5220      JMP ,+3          /BACKWARD
4016 4711      JMS I FWD TYP      /TYPE FORWARD
4017 5221      JMP ,+2
4020 4710      JMS I BKW TYP      /TYPE BACKWARD
4021 4456      JMS I TYPT EX      /TYPE (LAST PGS)
4022 0054      0054
4023 4163      4163
4024 6400      6400
4025 6057      6057
4026 6377      6377
4027 1026      TAD RECRDK
4030 4451      JMS I TYP CON      /TYPE BLOCK1 LOOKED FOR
4031 4456      JMS I TYPT EX      /TYPE (SEARCHED)
4032 0063      0063
4033 4541      4541
4034 6243      6243
4035 5045      5045
4036 4477      4477
4037 1021      TAD BLKFND
4040 7040      CMA
4041 7640      SZA CLA          /WAS A BLOCK NUMBER
4042 5245      JMP ,+3          /YES
4043 4267      JMS EZTYPE          /TYPE END ZONE
4044 5247      JMP ,+3
4045 1025      TAD IDCON          /TYPE BLOCK NUMBER
4046 4451      JMS I TYP CON
4047 4456      JMS I TYPT EX      /TYPE (FOUND)
4050 0046      0046
4051 5765      5765
4052 5644      5644
4053 7700      7700
4054 4447      JMS I SBTYPE
4055 7604      CLA OSR
4056 7006      RTL
4057 7630      SZA CLA          /DELETE STOP AFTER ERROR
4060 5263      JMP ,+3          /NO ERROR STOP
  
```

4061	1200	TAD SSTAER	
4062	7402	HLT	/DISPLAY ADDRESS
4063	7200	CLA	
4064	4707	JMS I SYNCRE	
4065	4441	JMS I WTHALF	
4066	5600	JMP I SSTAER	
/TYPE (END ZONE)			
4067	5267	EZTYPE,	JMP ,
4070	4456	JMS I TYPTX	
4071	7777	7777	
4072	4556	4556	
4073	4400	4400	
4074	7257	7257	
4075	5645	5645	
4076	7700	7700	
4077	5667	JMP I EZTYPE	
/SAVE POSITION AND DIRECTION POINTERS			
4100	5300	DAPSAV,	JMP ,
4101	7200	CLA	
4102	1422	TAD I POSITN	
4103	3111	DCA POSSAV	
4104	1423	TAD I DIRECT	
4105	3112	DCA DIRSAV	
4106	5700	JMP I DAPSAV	
4107	4200	SYNCRE,	RESYNC
4110	1545	BKWTYP,	TYBKW
4111	1555	FWDTP,	TYFWD

/PUT TAPE BACK TO LAST KNOWN POSITION

*4200

4200	5200	RESYNC,	JMP ,	
4201	1111	TAD	POSSAV	
4202	7040	CMA		
4203	7650	SNA	CLA	
4204	5354	JMP	RESYEZ	
4205	1112	TAD	DIRSAV	
4206	7640	SZA	CLA	
4207	5276	JMP	RESBKW	
4210	1101	TAD	K0614	/TAPE GOES BACKWARD
4211	1060	TAD	UNFUNC	/FIRST TO RESYNC
4212	6766	IOT	766	/FORWARD
4213	7040	RESFWD,	CMA	
4214	3427	DCA	I WCLOC	/-1 TO WC
4215	4440	JMS	I WAITI	
4216	0001	1		
4217	5260	JMP	RESFEZ	
4220	1021	TAD	BLKFND	
4221	7041	CMA	IAC	
4222	1111	TAD	POSSAV	
4223	1374	TAD	K7772	
4224	7700	SMA	CLA	
4225	5231	JMP	,+4	
4226	7200	CLA		
4227	6764	IOT	764	
4230	5213	JMP	RESFWD	
4231	1102	TAD	K0400	
4232	6764	RESFWF,	IOT 764	
4233	7040	CMA		
4234	3427	DCA	I WCLOC	
4235	4440	JMS	I WAITI	
4236	0001	1		
4237	5201	JMP	RESYNC+1	
4240	1111	TAD	POSSAV	
4241	7041	CMA	IAC	
4242	1021	TAD	BLKFND	
4243	7450	SNA		
4244	5250	JMP	,+4	
4245	7710	SPA	CLA	
4246	5232	JMP	RESFWF	
4247	5201	JMP	RESYNC+1	

```

4250 1075 RESXIT,      TAD K0200                /STOP TAPE
4251 6764      IOT 764
4252 1111      TAD POSSAV      /RESET POSITION
4253 3422      DCA I POSITN    /AND DIRECTION
4254 1112      TAD DIRSAV      /POINTERS
4255 3423      DCA I DIRECT
4256 4441      JMS I WTHALF
4257 5600      JMP I RESYNC

4260 6772 RESFEZ,      IOT 772
4261 0110      AND EZBIT
4262 7650      SNA CLA          /END ZONE
4263 5201      JMP RESYNC+1    /NO, SOME OTHER, RESYNC
4264 1065      TAD K0604
4265 1060      TAD UNFUNC
4266 6766      IOT 766          /MOVE FARTHER
4267 4440      JMS I WAITI     /INTO EZ
4270 5001      5001
4271 5264      JMP ,+5
4272 1103      TAD K0214
4273 1060      TAD UNFUNC
4274 6766      IOT 766          /NOW START FORWARD
4275 5233      JMP RESFWF+1

4276 1103 RESBKW,      TAD K0214                /TO RESYNC BKWD
4277 1060      TAD UNFUNC          /TAPE MUST FIRST
4300 6766      IOT 766            /GO FORWARD
4301 7040      CMA
4302 3427      DCA I WCLOC
4303 4440      JMS I WAITI
4304 0001      1
4305 5336      JMP RESBEZ        /TEST FOR END ZONE
4306 1373      TAD K0006
4307 1111      TAD POSSAV
4310 7041      CMA IAC
4311 1021      TAD BLKFND
4312 7700      SMA CLA          /REACH POSITION +6 YET
4313 5317      JMP ,+4          /YES
4314 7200      CLA
4315 6764      IOT 764          /GO FORWARD MORE
4316 5301      JMP RESBKW+3
4317 1102      TAD K0400        /CHANGE TO BACKWARD
4320 6764 RESBKB,      IOT 764
4321 7040      CMA
4322 3427      DCA I WCLOC
4323 4440      JMS I WAITI
4324 0001      1
4325 5201      JMP RESYNC+1     /NOT NORMAL STAT, TRY AGAIN

```

```

4326 1021 TAD BLKFND
4327 7041 CMA IAC
4330 1111 TAD POSSAV
4331 7450 SNA /IN POSITION YET
4332 5250 JMP RESXIT /YES
4333 7700 CLA SMA /GO PAST AGAIN
4334 5201 JMP RESYNC+1 /YES, TRY AGAIN
4335 5320 JMP RESBKB /NO, WAIT FOR NEXT BLOCK
    
```

```

4336 6772 RESBEZ, IOT 772
4337 0110 AND EZBIT
4340 7650 SNA CLA /END ZONE
4341 5201 JMP RESYNC+1 /NO, TRY AGAIN
4342 1125 TAD K0204
4343 1060 TAD UNFUNC
4344 6766 IOT 766 /MOVE INTO EZ AGAIN
4345 4440 JMS I WAITI
4346 5001 5001
4347 5201 JMP RESYNC+1
4350 1101 TAD K0614
4351 1060 TAD UNFUNC /NOW START BACKWARDS
4352 6766 IOT 766
4353 5321 JMP RESBKB+1
    
```

/PUT TAPE BACK INTO END ZONE

/LEZ OR TEZ

```

4354 1112 RESYEZ, TAD DIRSAV
4355 7640 SZA CLA /BACKWARD = NO SKIP
4356 1102 TAD K0400 /YES BACKWARD
4357 1125 TAD K0204
4360 1060 TAD UNFUNC
4361 6766 IOT 766
4362 4440 JMS I WAITI
4363 5001 5001
4364 5354 JMP RESYEZ
4365 1075 TAD K0200
4366 6764 IOT 764 /MOVE INTO EZ TWICE
4367 4440 JMS I WAITI
4370 5001 5001
4371 5365 JMP ,-4
4372 5251 JMP RESXIT+1
4373 0006 K0006, 6
4374 7772 K7772, 7772
    
```

PAUSE

/TC01 BASIC EXERCISER - TAPE 3A

/WRITE BASIC DATA PATTERNS

/READ VERIFY WRITE FORWARD

/READ BACKWARD, FORWARD, WRITE BACKWARD

/READ FORWARD, BACKWARD

*4400

```

4400 7604 WRTTST, LAS /GET SWITCHES
4401 0073 AND K0007 /MASK PATTERN NUM
4402 3370 DCA PATNUM
4403 1370 TAD PATNUM
4404 1365 TAD PATTBL
    
```


4405	3363	DCA TEMP1	/ADDRESS TO GET
4406	1763	TAD I TEMP1	/ROUTINE ADDRESS
4407	3363	DCA TEMP1	
4410	4763	JMS I TEMP1	/GENERATE PATTERN
4411	7040	CMA	
4412	3424	DCA I LSTBLK	
4413	3422	DCA I POSITN	
4414	4437	JMS I NEWDRV	
4415	5211	JMP , -4	
4416	3114	DCA DIRFLG	
4417	7001	IAC	
4420	3113	DCA BLKINC	
4421	1113	WRTLP1, TAD BLKINC	
4422	7510	SPA	
4423	7200	CLA	
4424	1424	TAD I LSTBLK	
4425	3020	DCA RECORD	
4426	4435	JMS I SRCHIT	
4427	1115	TAD K0050	
4430	6764	IOT 764	
4431	1032	TAD BF1LOC	
4432	3430	DCA I CALOC	
4433	1064	TAD K7577	
4434	3427	DCA I WCLOC	
4435	4761	JMS I WRTSLP	/CHECK 1,32 OR 2701 OPTIONS
4436	5230	JMP WRTLP1+7	/RETURN, NOT DONE ALL
4437	4437	JMS I NEWDRV	/RETURN, DONE ALL
4440	5221	JMP WRTLP1	
4441	1114	TAD DIRFLG	/SAVE WRITE DIR
4442	3771	DCA I PATNUM+1	/FOR ERROR TYPEOUTS
4443	1114	TAD DIRFLG	/MAKE 1ST RD PASS
4444	7040	CMA	/GO OTHER DIRECTION
4445	3114	DCA DIRFLG	
4446	1113	TAD BLKINC	/MAKE BLOCK
4447	7041	CMA IAC	/INCREMENTER
4450	3113	DCA BLKINC	/COMPLIMENT

```

/TEST READ COMPARE OPTION
/EXAMINE SWITCHES
RDCOMP,      LAS
4451  7604
4452  0121      AND K0040
4453  7640      SZA CLA
4454  5344      JMP RDCEND+13
4455  7040      CMA
4456  3364      DCA RDCPAS
4457  1422      TAD I POSITN
4460  3020      DCA RECORD
4461  4435      JMS I SRCHIT      /FIND BLOCK
4462  1116      TAD K0030
4463  6764      IOT 764      /CHANGE TO READ DATA
4464  1033      TAD BF2LOC
4465  3430      DCA I CALOC
4466  1064      TAD K7577
4467  3427      DCA I WCLOC
4470  4762      RDCLP1,      JMS I RDSWLP      /WAIT FOR READ INTERRUPT
4471  5315      JMP RDCEND
4472  6764      IOT 764      /RESET ENABLES:
4473  1034      TAD BF3LOC
4474  3430      DCA I CALOC
4475  1064      TAD K7577
4476  3427      DCA I WCLOC
4477  4442      JMS I DATACO      /VERIFY DATA PATTERN
4500  7175      BUFFER2
4501  4767      JMS I RERFLG

4502  4762      JMS I RDSWLP
4503  5315      JMP RDCEND      /DONE ALL
4504  6764      IOT 764      /RESET ENABLES: AGAIN
4505  1033      TAD BF2LOC
4506  3430      DCA I CALOC
4507  1064      TAD K7577
4510  3427      DCA I WCLOC
4511  4442      JMS I DATACO      /VERIFY DATA READ
4512  7376      BUFFER3
4513  4767      JMS I RERFLG
4514  5270      JMP RDCLP1

```

```

4515 6772  RDCEND,      IOT 772
4516 0110      AND EZBIT
4517 7640      SZA CLA
4520 5331      JMP RSEND      /END ZONE INTERRUPT
4521 1427      TAD I WCLOC
4522 7640      SZA CLA
4523 5331      JMP RSEND
4524 1430      TAD I CALOC
4525 1366      TAD K7600      /FORM BUFFER ADDRESS
4526 3330      DCA .+2
4527 4442      JMS I DATACO
4530 7175      BUFFER2      /OR BUFFER3
4531 4437  RSEND,      JMS I NEWDRV
4532 5257      JMP RDCOMP+6
4533 1114      TAD DIRFLG
4534 7040      CMA
4535 3114      DCA DIRFLG      /GO OTHER DIRECTION

4536 1113      TAD BLKINC      /MAKE BLOCK INCREMENTER
4537 7041      CMA IAC      /OTHER DIRECTION
4540 3113      DCA BLKINC
4541 2364      ISZ RDCPAS      /READ BOTH DIRECTIONS
4542 7410      SKP      /YES
4543 5257      JMP RDCOMP+6      /READ OTHER DIRECTION
4544 7604      CLA OSR
4545 0123      AND K0100
4546 7640      SZA CLA
4547 5255      JMP RDCOMP+4
4550 1424      TAD I LSTBLK
4551 1077      TAD K5077
4552 7640      SZA CLA      /WRITTEN 2701 YET
4553 5221      JMP WRTLP1      /WRITE NEXT SET
4554 1114      TAD DIRFLG
4555 7640      SZA CLA
4556 5221      JMP WRTLP1
4557 5760      JMP I .+1
4560 5161      WRRDND

4561 5200  WRTSLP,      WRTSWS      /TO TEST SWITCHES FOR WRITE
4562 5400  RDSWLP,      RDSWS      /TO TEST SWITCHES FOR READ
4563 0000  TEMP1,      0      /READ PASS SWITCH
4564 0000  RDCPAS,      0      /-1 PASS 1 0 PASS 2

/Routine ADDRESS FOR PATTERNS
4565 4724  PATTBL,      PTABLE
4566 7600  K7600,      7600
4567 4734  RERFLG,      REFLGS
4570 0000  PATNUM,      0
4571 5157      WRDIR

```

/PATTERN GENERATION FOR
/INITIAL WRITE TEST

```
*4600
4600 0000 GNPAT0,      0
4601 7300      CLA CLL
4602 4251      JMS GNSTRA
4603 5600      JMP I GNPAT0

4604 0000 GNPAT1,      0
4605 7340      CLA CMA CLL
4606 4251      JMS GNSTRA
4607 5604      JMP I GNPAT1

4610 0000 GNPAT2,      0
4611 7320      CLA STL
4612 4251      JMS GNSTRA
4613 5610      JMP I GNPAT2

4614 0000 GNPAT3,      0
4615 1221      TAD ,+4
4616 7100      CLL
4617 4251      JMS GNSTRA
4620 5614      JMP I GNPAT3
4621 2525      2525

4622 0000 GNPAT4,      0
4623 1227      TAD ,+4
4624 7100      CLL
4625 4251      JMS GNSTRA
4626 5622      JMP I GNPAT4
4627 5252      5252

4630 0000 GNPAT5,      0
4631 1235      TAD ,+4
4632 7100      CLL
4633 4251      JMS GNSTRA
4634 5630      JMP I GNPAT5
4635 0707      0707

4636 0000 GNPAT6,      0
4637 1243      TAD ,+4
4640 7100      CLL
4641 4251      JMS GNSTRA
4642 5630      JMP I GNPAT5
4643 7070      7070

4644 0000 GNPAT7,      0
4645 1221      TAD GNPAT4-1
4646 7120      STL
4647 4251      JMS GNSTRA
4650 5644      JMP I GNPAT7
```

/STORE AC CONTENTS IN BF1LOC
/OR IF L=1 COMPLIMENT EVERY OTHER

4651	0000	GNSTRA,	0
4652	3010	DCA	10
4653	1032	TAD	BF1LOC
4654	3011	DCA	11
4655	1064	TAD	K7577
4656	3012	DCA	12
4657	1010	TAD	10
4660	3411	DCA	I 11

4661	2012	ISZ	12
4662	7410	SKP	
4663	5651	JMP	I GNSTRA
4664	7420	SNL	
4665	5257	JMP	GNSTRA+6
4666	1010	TAD	10
4667	7040	CMA	
4670	3010	DCA	10
4671	5257	JMP	GNSTRA+6

4672	0100	PARTAB,	0100
4673	0200	0200	
4674	0400	0400	
4675	1000	1000	
4676	2000	2000	
4677	4000	4000	
4700	0101	0101	
4701	0202	0202	
4702	0404	0404	
4703	1010	1010	
4704	2020	2020	
4705	4040	4040	
4706	7600	7600	
4707	7500	7500	
4710	7300	7300	
4711	6700	6700	
4712	5700	5700	
4713	3700	3700	
4714	7700	7700	
4715	7676	7676	
4716	7575	7575	
4717	7373	7373	
4720	6767	6767	
4721	5757	5757	
4722	3737	3737	
4723	7777	7777	

4724	4600	P	TABLE,	GNPAT0
4725	4604		GNPAT1	
4726	4610		GNPAT2	
4727	4614		GNPAT3	
4730	4622		GNPAT4	
4731	4630		GNPAT5	
4732	4636		GNPAT6	
4733	4644		GNPAT7	

/TEST READ ERRORS

/AND RESYNC NEXT BLOCK

4734	5334	REFLGS,	JMP .
4735	2742	ISZ I	.,+5
4736	5744	JMP I	.,+6
4737	2743	ISZ I	.,+4
4740	5744	JMP I	.,+4
4741	5734	JMP I	REFLGS
4742	5150	COFLAG	
4743	5526	NOSERR	
4744	4461	RDCOMP	*10

/COMPARE DATA SUBROUTINE FOR
/WRITE / READ BASIC DATA PATTERNS

```

*5000
5000 5200 CODATA,      JMP .
5001 7200      CLA
5002 1031      TAD BF1WD1          /GET KNOWN DATA
5003 3345      DCA KNDATA          /ADDRESS
5004 1600      TAD I CODATA        /UNKNOWN DATA
5005 3346      DCA UKDATA          /ADDRESS
5006 2200      ISZ CODATA
5007 7040      CMA
5010 3350      DCA COFLAG          /SET ERR COJNT FLAG
5011 1064      TAD K7577
5012 3347      DCA NUMWRD          /NUMBER OF WORDS

/TST FOR DELETE COMPARE
5013 7604      LAS
5014 0102      AND K0400
5015 7640      SZA CLA
5016 5600      JMP I CODATA
5017 1745      COLOOP,      TAD I KNDATA
5020 7041      CMA IAC
5021 1746      TAD I UKDATA
5022 7640      SZA CLA          /WORDS =
5023 5231      JMP COCOMP          /NO TEST COMPLIMENT

5024 2345      COINCR,      ISZ KNDATA
5025 2346      ISZ UKDATA
5026 2347      ISZ NUMWRD          /DONE ALL
5027 5217      JMP COLOOP
5030 5600      JMP I CODATA          /EXIT

5031 1745      COCOMP,      TAD I KNDATA
5032 7450      SNA          /WORD = 0'S
5033 5237      JMP ,+4          /YES, TRY COMPLIMENT
5034 7040      CMA
5035 7640      SZA CLA          /WORD=1'S
5036 5251      JMP COERRO          /NO DATA ERROR
5037 1745      TAD I KNDATA
5040 7001      IAC          /MAKE 2'S COMPI

5041 1746      TAD I UKDATA
5042 7640      SZA CLA          /COMPLIMENTS =
5043 5251      JMP COERRO          /NO ERROR
5044 2345      ISZ KNDATA
5045 2346      ISZ UKDATA
5046 2347      ISZ NUMWRD
5047 5237      JMP COCOMP+6          /STAY IN TEST COMP
5050 5600      JMP I CODATA

```

/DATA ERROR TYPEOUT

5051	2350	COERRO,	ISZ COFLAG	/FIRST ERROR
5052	5307	JMP COERR1		/NO HDR ALREADY TYPED
5053	4444	JMS I ERRSTP		/STOP TAPE
5054	4446	JMS I RDATTY		
5055	4456	JMS I TYPTX		
5056	0044	44		
5057	4164	4164		
5060	4100	4100		
5061	4562	4562		
5062	6257	6257		
5063	6277	6277		
5064	1022	TAD POSITN		
5065	4451	JMS I TYPCON		
5066	4456	JMS I TYPTX	/TYPE BLOCK	
5067	0042	42		
5070	5457	5457		
5071	4353	4353		
5072	0067	67		
5073	6251	6251		
5074	6464	6464		
5075	4556	4556		
5076	0077	0077		
5077	1357	TAD WRDIR	/TYPE DIRECTION	
5100	7650	SNA CLA	/BLOCK WAS WRITTEN	
5101	5306	JMP ,+5		
5102	4704	JMS I ,+2		
5103	5307	JMP COERR1		
5104	1545	TYBKW		
5105	1555	TYFWD		
5106	4705	JMS I .-1		

5107	4456	COERR1,	JMS I TYPTX	/LINE FEED TO
5110	7777	7777		/SEPARATE ERRORS
5111	7700	7700		
5112	1345	TAD KNDATA		
5113	4451	JMS I TYPCON		/TYPE GOOD DATA
5114	4456	JMS I TYPTX		
5115	0053	53		
5116	5657	5657		
5117	6756	6756		
5120	7700	7700		
5121	1346	TAD UKDATA		/TYPE DATA READ
5122	4451	JMS I TYPCON		
5123	4456	JMS I TYPTX		
5124	0065	65		
5125	5653	5653		
5126	5657	5657		
5127	6756	6756		
5130	7700	7700		
5131	1344	TAD KNDATA-1		
5132	4451	JMS I TYPCON		/TYPE ADDRESS OF
5133	4456	JMS I TYPTX		/KNOW DATA
5134	0041	41		
5135	4444	4444		
5136	6263	6263		
5137	0053	53		
5140	5657	5657		
5141	6756	6756		
5142	7700	7700		
5143	5224	JMP COINCR		
5144	5145	,+1		
5145	0000	KNDATA,	0	
5146	0000	JKDATA,	0	
5147	0000	NUMWRD,	0	
5150	0000	COFLAG,	0	
5151	5351	WAETYP,	JMP ,	
5152	4444	JMS I ERRSTP		
5153	4756	JMS I ,+3		
5154	4447	JMS I SBTYPE		
5155	5751	JMP I WAETYP		
5156	1451	TYWALL		
5157	0000	NRDIR,	0	
		STPERR=ERRSTP		

5160	4570	PATNUM	
5161	1360	WRRDND,	TAD ,-1
5162	4451	JMS I TYPCON	/TYPE PATTERN NUMBER
5163	4456	JMS I TYPTX	
5164	0045	45	
5165	5644	5644	
5166	7700	7700	
5167	7604	CLA OSR	
5170	0075	AND K0200	
5171	7650	SNA CLA	/DO NEXT PATTERN
5172	5777	JMP I ,+5	/NO USE SWS
5173	1760	TAD I WRRDND=1	
5174	7001	IAC	/PATNUM+1
5175	5776	JMP I ,+1	
5176	4401	WRTTST+1	
5177	4400	WRTTST	

/WAIT FOR WRITE INTERRUPT
/AND TEST SWITCHES FOR NUM BLOCKS

*5200

5200	5200	WRTSWS,	JMP	.	
5201	4440	JMS I WAITI			/WAIT NORMA_ INT
5202	0001	1			
5203	5235	JMP WRTEZT			/TEST FOR END ZONE
5204	1427	TAD I WCLOC			
5205	7640	SZA CLA			/WC GO TO 0
5206	5241	JMP WRTEZT+4			/NO ERROR
5207	7604	CLA OSR			
5210	0117	AND BLKBTS			
5211	7450	SNA			/INDICATE 1 OR 32 BLOCKS
5212	5221	JMP .+7			/NO DO ALL
5213	0120	AND BLKBIT			
5214	7650	SNA CLA			/BLOCK
5215	5312	JMP WRT32			/NO TEST 32 BLOCKS
5216	2200	ISZ WRTSWS			/INC EXIT ADDRESS
5217	1075	TAD K0200			
5220	6764	IOT 764			/STOP TAPE
5221	1020	TAD RECORD			
5222	3422	DCA I POSITN			
5223	1020	TAD RECORD			
5224	1113	TAD BLKINC			
5225	3020	DCA RECORD			
5226	1114	WRTSWA,	TAD DIRFLG		
5227	7640	SZA CLA			/GOING FORWARD
5230	5600	JMP I WRTSWS			/BACKWARD EXIT
5231	1424	TAD I LSTBLK			
5232	7001	IAC			/INCREMENT _AST
5233	3424	DCA I LSTBLK			/BLOCK WRITTEN
5234	5600	JMP I WRTSWS			
5235	6772	WRTEZT,	IOT 772		
5236	0110	AND EZBIT			
5237	7640	SZA CLA			
5240	5301	JMP WRTEZA			
5241	6761	IOT 761			
5242	0075	AND K0200			/IF TAPE NOT STOPPED
5243	7650	SNA CLA			
5244	5251	JMP .+5			/IF SW3=1
5245	7604	LAS			
5246	0102	AND K0400			
5247	7640	SZA CLA			
5250	5207	JMP WRTSWS+7			
5251	4444	JMS I ERRSTP			
5252	4450	JMS I WDATTY			
5253	1026	TAD RECRDK			
5254	4451	JMS I TYPCON			

5255	4456	JMS I	TYPTX	
5256	0042	42		
5257	5457	5457		
5260	4353	4353		
5261	7700	7700		
5262	4447	JMS I	SBTYPE	
5263	1027	TAD	WCLOC	
5264	4451	JMS I	TYPCON	/TYPE WHATS LEFT OF WC
5265	4456	JMS I	TYPTX	
5266	0067	67		
5267	1643	1643		
5270	1677	1677		
5271	4435	JMS I	SRCHIT	
5272	1115	TAD	K0050	
5273	6764	IOT	764	
5274	1032	TAD	BF1LOC	
5275	3430	DCA I	CALOC	
5276	1064	TAD	K7577	
5277	3427	DCA I	WCLOC	
5300	5201	JMP	WRTSWS+1	
5301	1020	WRTEZA,	TAD RECORD	
5302	1126	TAD	K5076	
5303	7650	SNA	CLA	
5304	5310	JMP	,+4	
5305	1020	TAD	RECORD	
5306	7700	SMA	CLA	
5307	5241	JMP	WRTEZT+4	
5310	2200	ISZ	WRTSWS	
5311	5600	JMP	I WRTSWS	
5312	1020	WRT32,	TAD RECORD	
5313	3422	DCA I	POSITN	
5314	1020	TAD	RECORD	
5315	1113	TAD	BLKINC	
5316	3020	DCA	RECORD	
5317	1114	TAD	DIRFLG	
5320	7041	CMA	IAC	
5321	1020	TAD	RECORD	
5322	0127	AND	K0037	
5323	7640	SZA	CLA	
5324	5226	JMP	WRTSWA	
5325	1075	TAD	K0200	
5326	6764	IOT	764	
5327	2200	ISZ	WRTSWS	
5330	5226	JMP	WRTSWA	

/WAIT FOR READ INTERRUPT AND
/TEST SWITCHES FOR NUMBER OF BLOCKS

```

*5400
5400 5200 RDSWS,      JMP .
5401 4440      JMS I WAITI      /WAIT NORMAL INTERRUPT
5402 0001      1
5403 5252      JMP TSRDEZ          /TEST FOR END ZONE
5404 1427      TAD I WCLOC
5405 7640      SZA CLA
5406 5302      JMP RDSERR
5407 7040      CMA
5410 3326      DCA NOSERR
5411 7604      CLA OSR
5412 0117      AND BLKBTS
5413 7450      SNA
5414 5225      JMP .+11      /NO, ALL /1 OR 32 BLOCKS

5415 0120      AND BLKBIT
5416 7650      SNA CLA      /1 BLOCK
5417 5234      JMP RDSW32      /NO, 32 BLOCKS:
5420 1075      TAD K0200
5421 6764      IOT 764      /1 BLOCK, STOP TAPE
5422 1020      TAD RECORD
5423 3422      DCA I POSITN
5424 5600      JMP I RDSWS
5425 1020      TAD RECORD
5426 3422      DCA I POSITN      /NEW POSITION
5427 1020      TAD RECORD
5430 1113      TAD BLKINC      /NEXT BLOCK
5431 3020      DCA RECORD
5432 2200      ISZ RDSWS
5433 5600      JMP I RDSWS

5434 1020      RDSW32,      TAD RECORD
5435 3422      DCA I POSITN
5436 1020      TAD RECORD
5437 1113      TAD BLKINC      /+ OR - 1
5440 3020      DCA RECORD
5441 1114      TAD DIRFLG
5442 7041      CMA IAC
5443 1020      TAD RECORD
5444 0127      AND K0037
5445 7640      SZA CLA      /DONE AN INC OF 32
5446 5232      JMP RDSW32-2
5447 1075      TAD K0200
5450 6764      IOT 764      /32 BLOCKS, STOP TAPE
5451 5600      JMP I RDSWS
    
```

```

5452 6772  TSRDEZ,      IOT 772
5453 0110      AND EZBIT
5454 7650      SNA CLA      /END ZONE INT
5455 5302      JMP RDSERR      /NO, ERROR
5456 1020      TAD RECORD
5457 1126      TAD K5076      /BLOCK 2702
5460 7650      SNA CLA
5461 5265      JMP ,+4      /2702 DOESN'T EXIST
5462 1020      TAD RECORD
5463 7700      SMA CLA      /BLOCK-1
5464 5302      JMP RDSERR      /NO, EZ STAT IN ERROR
5465 5600      JMP I RDSWS      /EXIT

5466 5266      PARRSE,      JMP ,      /STOP TAPE
5467 4444      JMS I ERRSTP
5470 4446      JMS I RDATTY      /READ AND DIRECTION
5471 1026      TAD RECRDK
5472 4451      JMS I TYPCON      /TYPE BLOCK
5473 4456      JMS I TYPTX
5474 0042      42
5475 5457      5457
5476 4353      4353
5477 7700      7700
5500 4447      JMS I SBTYPE
5501 5666      JMP I PARRSE
5502 6761      RDSERR,      IOT 761
5503 0075      AND K0200
5504 7650      SNA CLA
5505 5312      JMP ,+5
5506 7604      LAS
5507 0102      AND K0400
5510 7640      SZA CLA
5511 5207      JMP RDSWS+7
5512 4266      JMS PARRSE
5513 3326      DCA NOSERR

5514 1427      TAD I WCLOC
5515 7650      SNA CLA      /WC GO TO 0
5516 5211      JMP RDSWS+11      /YES
5517 1027      TAD WCLOC
5520 4451      JMS I TYPCON
5521 4456      JMS I TYPTX      /TYPE WORD COJNT
5522 0067      67
5523 1643      1643
5524 1677      1677
5525 5211      JMP RDSWS+11
5526 0000      NOSERR,      0

```

/WRITE DATA SCOPE LOOP
 /NO ERROR CHECKING BOUNCES OFF EZ
 /PATTERN SELECTION BITS 9, 10, 11

```

*1600
1600 7604 *RSCOP, LAS
1601 0073 AND K0007
1602 1371 TAD PATTBA /PAT NUM + TABLE AORS
1603 3370 DCA TEMPY
1604 1770 TAD I TEMPY
1605 3370 DCA TEMPY
1606 4770 JMS I TEMPY /GENERATE PATTERN
1607 1103 TAD K0214 /SEARCH +
1610 1060 TAD UNFUNC /DRIVE
1611 6766 IOT 766
1612 7440 SZA
1613 7402 HLT /IOT 766 OR 754 DID NOT CLEAR AC
1614 1025 TAD IDCON
1615 3430 DCA I CALOC
1616 6773 IOT 773 /WAIT FOR FLAG
1617 5216 JMP .-1
1620 7710 SPA CLA
1621 5255 JMP WRTSEZ /ERROR STATUS
1622 1115 TAD K0050
1623 6764 IOT 764 /CHANGE TO WRITE
1624 7440 SZA
1625 7402 HLT /IOT 764 DID NOT CLEAR AC
1626 1032 TAD BF1LOC
1627 3430 DCA I CALOC
1630 1064 TAD K7577
1631 3427 DCA I WCLOC
1632 6761 IOT 761
1633 0075 AND K0200
1634 7650 SNA CLA
1635 5243 JMP WRSCO1
1636 1427 TAD I WCLOC
1637 7650 SNA CLA
1640 5243 JMP .+3
1641 6771 IOT 771 /WAIT FOR FLAG
1642 5232 JMP .-10
1643 1375 *RSCO1, TAD K7730
1644 3370 DCA TEMPY
1645 6771 IOT 771
1646 7410 SKP
1647 5252 JMP .+3
1650 2370 ISZ TEMPY
1651 5245 JMP .-4
1652 1376 TAD K0052 /LEAVE ERROR FLAGS SET
1653 6764 IOT 764
1654 5212 JMP WRSCOP+12
1655 6772 *WRTSEZ, IOT 772
1656 0374 AND K1000
1657 7650 SNA CLA /END ZONE
1660 5265 JMP .+5 /NO START SEARCH AGAIN
1661 6761 IOT 761 /END ZONE SET
1662 7040 CMA /CHANGE DIRECTION
1663 0102 AND K0400
1664 5207 JMP WRSCOP+7
1665 6761 IOT 761

```

MAINDEC-08-D3BB-D

79

1666 5263

JMP .-3

/READ DATA SCOPE LOOP
 /IGNORES ALL ERRORS
 /BOUNCES OFF END ZONE

1667	1372	RDSCOP,	TAD K0220	/READ DATA
1670	1060	TAD UNFUNC		/+ DRIVE
1671	6766	IOT 766		
1672	7440	SZA		
1673	7402	HLT		/IOT 764 DID NOT CLEAR AC
1674	1033	TAD BF2LOC		
1675	3430	DCA I CALOC		
1676	1064	TAD K7577		
1677	3427	DCA I WCLOC		
1700	6761	IOT 761		/MONITOR MOTION
1701	0075	AND K0200		/BIT IN CASE IT=0
1702	7650	SNA CLA		
1703	5311	JMP .+6		
1704	1427	TAD I WCLOC		/MONITOR WORD K
1705	7650	SNA CLA		
1706	5311	JMP .+3		
1707	6771	IOT 771		/AND FLAGS
1710	5304	JMP .-4		
1711	1375	TAD K7730		
1712	3370	DCA TEMPY		
1713	6773	IOT 773		/MONITOR FLAG
1714	7410	SKP		/FOR 200 MICRO SEC.
1715	5320	JMP .+3		
1716	2370	ISZ TEMPY		
1717	5313	JMP .-4		
1720	0374	AND K1000		
1721	7640	SZA CLA		/END ZONE SET
1722	5330	JMP .+6		/YES, REVERSE
1723	6761	RDSC01,	IOT 761	
1724	7040	CMA		
1725	0075	AND K0200		/IN CASE GO=0
1726	6764	IOT 764		/RESET ENABLES
1727	5272	JMP RDSCOP+3		
1730	6761	IOT 761		
1731	7040	CMA		
1732	0102	AND K0400		/CHANGE DIRECTION
1733	5267	JMP RDSCOP		

/SEARCH SCOPE LOOP IGNORES ERRORS
 /BOUNCES OFF END ZONES
 /DISPLAYS LAST BLOCK IN AC

1734	1103	SRSCOP,	TAD K0214	/SEARCH
1735	1060	TAD UNFUNC		/+ DRIVE
1736	6766	IOT 766		
1737	7440	SZA		
1740	7402	HLT	/IOT 764 DID NOT CLEAR AC	
1741	1377	TAD K3500		
1742	3010	DCA 10		
1743	1031	TAD BF1WD1		
1744	3430	DCA I CALOC		
1745	1431	TAD I BF1WD1	/DISPLAY BLOCK	
1746	2010	ISZ 10	/WAIT 1 BLOCK	
1747	5346	JMP .-1	/APPROX	
1750	7200	CLA		
1751	6772	IOT 772		
1752	0374	AND K1000	/READ B	
1753	7640	SZA CLA	/END ZONE	
1754	5362	JMP .+6	/YES REVERSE	
1755	6761	IOT 761		
1756	7040	CMA	/IN CASE GO=0	
1757	0075	AND K0200		
1760	6764	IOT 764		
1761	5337	JMP SRSCOP+3		
1762	6761	IOT 761		
1763	0075	AND K0200		
1764	7640	SZA CLA		
1765	7402	HLT	/EZ DID NOT ZERO MOTION	
1766	1373	TAD K0600A		
1767	5360	JMP .-7	/COMPLEMENT DIRECTION	
1770	0000	TEMPY,	0	
1771	4724	PATTRA,	PTABLE	
1772	0220	<0220,	220	
1773	0600	<0600A,	600	
1774	1000	<1000,	1000	
1775	7730	<7730,	7730	
1776	0052	<0052,	52	
1777	3500	<3500,	3500	

PAUSE

/TAPE 4 OF TC01 BASIC EXERCISER
 /PARITY GENERATION TEST
 /IS CORRECT PARITY GENERATED
 /BEGIN BY WRITING REV CKSUMS TO 0
 /BACKWARD IS 77#00 FWD
 *5600

5600	1100	PARTST,	TAD K2701	
5601	3020	DCA RECORD	/FIND 2701	
5602	7040	CMA		
5603	3114	DCA DIRFLG	/BACKWARDS	
5604	4435	JMS I SRCHIT		
5605	1116	TAD K0030	/SEARCH TO READ DATA	
5606	6764	IOT 764		
5607	1032	TAD BF1LOC		
5610	3430	DCA I CALOC		
5611	1357	TAD K7600B	/DUMMY INPUT	
5612	3427	DCA I WCLOC		
5613	1427	TAD I WCLOC		
5614	7650	SNA CLA	/WAIT FOR WCTD=-1	
5615	5221	JMP ,+4		
5616	6771	IOT 771	/FLAG SET	
5617	5213	JMP , -4	/COULD BE END ZONE	
5620	5242	JMP PAREZ1		
5621	7001	IAC		
5622	7040	CMA		
5623	3427	DCA I WCLOC	/2 WORDS	
5624	1032	TAD BF1LOC		
5625	3430	DCA I CALOC		
5626	1122	TAD K0170	/TO WRITE ALL CONTINUOUS	
5627	6764	IOT 764		
5630	7040	CMA		
5631	3431	DCA I BF1WD1		
5632	4440	JMS I WAITI		
5633	0001	1		
5634	5237	JMP ,+3		
5635	1122	TAD K0170	/BACK TO READ DATA	
5636	5206	JMP PARTST+6		
5637	4641	JMS I ,+2		
5640	5200	JMP PARTST		
5641	5151	WAETYP		
5642	6772	PAREZ1, IOT 772		
5643	0110	AND EZBIT	/END ZONE INT.	
5644	7640	SZA CLA		
5645	5251	JMP ,+4		
5646	4650	JMS I ,+2		
5647	5200	JMP PARTST		
5650	5466	PARRSE		
5651	4437	JMS I NEWDRV		
5652	5200	JMP PARTST	/PREPARE NEXT DRIVE	

5653	1356	PARWL1,	TAD PARLOC	
5654	3010		DCA 10	
5655	3114		DCA DIRFLG	
5656	3020		DCA RECORD	
5657	3422		DCA I POSITN	
5660	4435		JMS I SRCHIT	/FIND BLOCK 0 FWD
5661	1115		TAD K0050	
5662	6764		IOT 764	/WRITE DATA
5663	1410		TAD I 10	
5664	3431		DCA I BF1WD1	
5665	1032		TAD BF1LOC	
5666	3430		DCA I CALOC	/ONLY WRITE 1 WD
5667	7040		CMA	
5670	3427		DCA I WCLOC	/REST OF BLOCKS SHOULD
5671	4440		JMS I WAITI	/GO TO ZERO'S
5672	0001		1	
5673	5344		JMP PARWER	
5674	1431		TAD I BF1WD1	
5675	7040		CMA	
5676	7650		SNA CLA	/DO ALL ONES YET?
5677	5303		JMP ,+4	/YES
5700	2020		ISZ RECORD	
5701	2422		ISZ I POSITN	/COUNT BLOCKS
5702	5262		JMP PARWL1+7	

/INCREMENTING PARITY PATTERNS
/0100 TO 7700 AND 0101 TO 7777

5703	3431	PARWL2,	DCA I BF1WD1	
5704	6764		IOT 764	
5705	7040		CMA	
5706	3427		DCA I WCLOC	
5707	1032		TAD BF1LOC	
5710	3430		DCA I CALOC	
5711	1431		TAD I BF1WD1	/UPPER +1
5712	1123		TAD K0100	
5713	3431		DCA I BF1WD1	
5714	1431		TAD I BF1WD1	
5715	7450		SNA	/UPPER GONE TO 0
5716	5323		JMP ,+5	/YES, SET WORD TO 0101
5717	0071		AND K0077	
5720	7640		SZA CLA	/INCREMENTING LWR
5721	2431		ISZ I BF1WD1	/ADD 1 TO LOWER
5722	5325		JMP ,+3	
5723	1124		TAD K0101	/DONE UPPER TO 7700
5724	3431		DCA I BF1WD1	/START BOTH EQUAL
5725	4440		JMS I WAITI	
5726	0001		1	
5727	5344		JMP PARWER	/SHOULD GET NO ERROR STATUS
5730	2422		ISZ I POSITN	
5731	2020		ISZ RECORD	
5732	1431		TAD I BF1WD1	
5733	7040		CMA	
5734	7640		SZA CLA	
5735	5304		JMP PARWL2+1	

5736	1075	TAD	K0200	
5737	6764	IOT	764	
5740	4437	JMS	I NEWDRV	
5741	5253	JMP	PARWL1	
5742	5743	JMP	I .+1	/READ AND CHECK
5743	6200	PARTS1		/FOR CORRECT CKSUMS
5744	4444	PARWER,	JMS I ERRSTP	
5745	4450	JMS	I WDATTY	
5746	1026	TAD	RECRDK	
5747	4451	JMS	I TYPCON	
5750	0042	42		
5751	5457	5457		
5752	4353	4353		
5753	7700	7700		
5754	4447	JMS	I SBTYPE	
5755	5253	JMP	PARWL1	
5756	4671	PARLOC,	PARTAB-1	
5757	7600	K7600B,	7600	

/WRITE BLOCKS TO WRONG
 /PARITY AND VERIFY PARTITY ERRORS
 /GENERATED (GOING BACKWARD REWRITE|REV, CKSUM)
 *6000

6000	3020	PARTS4,	DCA RECORD	
6001	3114	DCA	DIRFLG	
6002	4435	JMS	I SRCHIT	/FIND LAST BLOCK BKWD
6003	1355	TAD	K0140	/CHNG TO WRITE ALL
6004	6764	IOT	764	
6005	1122	TAD	K0170	/TO READ DATA
6006	6764	IOT	764	
6007	1032	TAD	BF1LOC	
6010	3430	DCA	I CALOC	
6011	1375	TAD	K7600A	
6012	3427	DCA	I WCLOC	
6013	1427	TAD	I WCLOC	
6014	7650	SNA	CLA	/WAIT FOR LAST
6015	5221	JMP	,+4	/WORD IN
6016	6771	IOT	771	
6017	5213	JMP	,+4	
6020	5356	JMP	PARRE3	/NO FLAGS FOR READ DATA
6021	1032	TAD	BF1LOC	/WRITE CHECKSUM TO FIRST WORD
6022	3430	DCA	I CALOC	
6023	7001	IAC		
6024	7040	CMA		
6025	3427	DCA	I WCLOC	
6026	1122	TAD	K0170	
6027	6764	IOT	764	/WRITE ALL
6030	1431	TAD	I BF1WD1	
6031	7040	CMA		
6032	7650	SNA	CLA	
6033	3431	DCA	I BF1WD1	
6034	4440	JMS	I WAITI	/WRITE ALL CONTINUOUS
6035	0001	1		
6036	5311	JMP	PRWAE	/WRITE ALL STATUS ERR
6037	4361	JMS	PR4INC	
6040	5205	JMP	PARTS4+5	
6041	1075	TAD	K0200	
6042	6764	IOT	764	
6043	4437	JMS	I NEWDRV	
6044	5200	JMP	PARTS4	

```

/READ BLOCKS FORWARD AND
/EXPECT PARITY ERRORS THEN BACKWARDS
PARTS5,      DCA RECORD
6045  3020
6046  3114      DCA DIRFLG
6047  4435      JMS I SRCHIT      /FIND 0 FWD OR LAST
6050  1116      TAD K0030
6051  6764      IOT 764      /READ DATA
6052  1032      TAD BF1LOC
6053  3430      DCA I CALOC

6054  1064      TAD K7577
6055  3427      DCA I WCLOC
6056  4440      JMS I WAITI      /EXPECT PARITY
6057  4201      4201      /ERROR
6060  5277      JMP PRT5EZ
6061  6772      IOT 772
6062  0075      AND K0200
6063  7650      SNA CLA      /PARITY ERROR SET

6064  5314      JMP PARRE4      /NO
6065  4361      JMS PR4INC
6066  5251      JMP PARTS5+4

6067  1075      TAD K0200
6070  6764      IOT 764      /STOP TAPE
6071  4437      JMS I NEWDRV      /CHANGE DRIVES
6072  5245      JMP PARTS5

```

```

/READ BLOCKS BACKWARDS AND EXPECT
/PARITY ERRORS
6073 1422 PARTS6, TAD I POSITN /LAST BLOCK
6074 3020 DCA RECORD
6075 7040 CMA /BACKWARDS
6076 5246 JMP PARTS5+1

6077 1114 PRTSEZ, TAD DIRFLG
6100 7700 SMA CLA /GOING BACKWARD
6101 5314 JMP PARRE4 /NO, ERROR
6102 1020 TAD RECORD
6103 7700 SMA CLA /DONE BLOCK 0
6104 5314 JMP PARRE4 /NO, ERROR
6105 4437 JMS I NEWDRV
6106 5273 JMP PARTS6
6107 5710 JMP I ,+1
6110 5600 PARTST

6111 4713 PRWAE, JMS I ,+2
6112 5202 JMP PARTS4+2
6113 5151 WAETYP

6114 4444 PARRE4, JMS I ERRSTP
6115 4446 JMS I RDATTY
6116 1026 TAD RECRDK
6117 4451 JMS I TYPCON
6120 4456 JMS I TYPTX

6121 0042 42
6122 5457 5457
6123 4353 4353
6124 7777 7777
6125 6041 6041
6126 6251 6251
6127 6471 6471
6130 0045 45
6131 6262 6262
6132 5762 5762
6133 0045 45
6134 7060 7060
6135 4543 4543
6136 6445 6445
6137 4477 4477

6140 4447 JMS I SBTYPE
    
```


6141	1114	TAD DIRFLG	
6142	7640	SZA CLA	
6143	1075	TAD K0200	
6144	1031	TAD BF1WD1	
6145	4451	JMS I TYPCON	
6146	4361	JMS PR4INC	
6147	7410	SKP	
6150	5271	JMP PARTS6-2	
6151	1020	TAD RECORD	
6152	7710	SPA CLA	
6153	5305	JMP PRWAE-4	
6154	5247	JMP PARTS5+2	
6155	0140	K0140,	140
6156	4760	PARRSE,	JMS I ,+2
6157	5202	JMP PARTS4+2	
6160	5466	PARRSE	
6161	5361	PR4INC,	JMP ,
6162	1114	TAD DIRFLG	
6163	7450	SNA	
6164	2020	ISZ RECORD	
6165	1020	TAD RECORD	
6166	3020	DCA RECORD	
6167	1422	TAD I POSITN	
6170	7040	CMA	
6171	1020	TAD RECORD	
6172	7650	SNA CLA	
6173	2361	ISZ PR4INC	
6174	5761	JMP I PR4INC	
6175	7600	K7600A,	7600

/READ THE GENERATED CKSUMS BACK
 /AND VERIFY THAT THEY ARE CORRECT
 *6200

6200	3020	PARTS1,	DCA RECORD	
6201	3114	DCA	DIRFLG	
6202	4435	JMS	I SRCHIT	/FIND BLOCK 0
6203	7040	CMA		/CA = 7777 IN CASE
6204	3430	DCA	I CALOC	/R ALL BREAKS BEFORE DATA
6205	1074	TAD	K0020	
6206	6764	IOT	764	
6207	1130	TAD	K0010	
6210	6764	IOT	764	/READ ALL TO RD DATA
6211	1032	TAD	BF1LOC	
6212	3430	DCA	I CALOC	
6213	1064	TAD	K7577	
6214	3427	DCA	I WCLOC	
6215	1427	TAD	I WCLOC	
6216	7650	SNA	CLA	/WAIT FOR WC TO =0
6217	5223	JMP	,+4	
6220	6771	IOT	771	/IN CASE READ ERROR
6221	5215	JMP	,+4	
6222	5342	JMP	PARRE1	
6223	7040	CMA		
6224	3427	DCA	I WCLOC	
6225	1130	TAD	K0010	
6226	6764	IOT	764	/RD DATA TO RD ALL
6227	4440	JMS	I WAITI	
6230	0001	1		
6231	5342	JMP	PARRE1	/WAIT FOR CKSUM IN
6232	1431	TAD	I BF1WD1	
6233	0071	AND	K0077	/TEST FOR 2 CHAR
6234	7640	SZA	CLA	/IN FIRST WORD
6235	5240	JMP	,+3	/CKSUM SHOULD =7700
6236	1431	TAD	I BF1WD1	
6237	0070	AND	K7700	
6240	1756	TAD	I CKSLOC	
6241	1123	TAD	K0100	
6242	7640	SZA	CLA	/CKSUM CORRECT
6243	5310	JMP	CKSERR	/NO
6244	4755	JMS	I PR2INC	
6245	5207	JMP	PARTS1+7	/NO
6246	1075	TAD	K0200	
6247	6764	IOT	764	/STOP TAPE
6250	4437	JMS	I NEWDRV	
6251	5200	JMP	PARTS1	/DO NEXT DRIVE

```

/READ BLOCKS BKWD FOR NO PARITY
/ERRORS
6252 1422 PARTS2,      TAD I POSITN
6253 3020      DCA RECORD
6254 7040      CMA
6255 3114      DCA DIRFLG
6256 4435      JMS I SRCHIT      /FIND LAST BLOCK BKWDS
6257 1116      TAD K0030
6260 6764      IOT 764
6261 1032      TAD BF1LOC

6262 3430      DCA I CALOC
6263 1064      TAD K7577
6264 3427      DCA I WCLOC
6265 4440      JMS I WAITI
6266 0001      1
6267 5273      JMP PARREZ      /TEST FOR END ZONE
6270 4755      JMS I PR2INC
6271 5260      JMP PARTS2+6
6272 5302      JMP PARTS3+2
6273 1020      PARREZ,      TAD RECORD
6274 7700      SMA CLA      /DONE BLOCK 0
6275 5345      JMP PARRE2      /NO, ERROR STATUS
6276 4437      JMS I NEWDRV      /DONE ALL DRIVES
6277 5252      JMP PARTS2      /DO NEXT

/READ BLOCKS FORWARD FOR NO
/PARITY ERRORS
6300 3020      PARTS3,      DCA RECORD
6301 5255      JMP PARTS2+3      /BLOCK 0 FWD
    
```

6302	1075	TAD K0200	
6303	6764	IOT 764 /STOP TAPE	
6304	4437	JMS I NEWDRV	/ALL DRIVES
6305	5300	JMP PARTS3	/NOPE
6306	5707	JMP I ,+1	/WRITE TO WRONG
6307	6000	PARTS4	/PARITY AND TEST ERROR
6310	4444	CKSERR,	JMS I ERRSTP
6311	4445	JMS I DRIVTY	
6312	4456	JMS I TYPTX	
6313	0043	43	
6314	5363	5363	
6315	6555	6555	
6316	0045	45	
6317	6262	6262	
6320	5762	5762	
6321	7700	7700	
6322	1031	TAD BF1WD1	
6323	4451	JMS I TYPCON	
6324	4456	JMS I TYPTX	
6325	0044	44	
6326	4164	4164	
6327	4177	4177	
6330	1356	TAD CKSLOC	
6331	4451	JMS I TYPCON	
6332	4456	JMS I TYPTX	
6333	0043	43	
6334	5300	5300	
6335	6365	6365	
6336	5577	5577	
6337	4755	JMS I PR2INC	
6340	5202	JMP PARTS1+2	
6341	5250	JMP PARTS2+2	
6342	4744	PARRE1,	JMS I ,+2
6343	5202	JMP PARTS1+2	
6344	5466	PARRSE	
6345	4744	PARRE2,	JMS I , -1
6346	4755	JMS I PR2INC	
6347	7410	SKP	
6350	5304	JMP PARTS3+4	
6351	1020	TAD RECORD	
6352	7710	SPA CLA	
6353	5276	JMP PARTS3-2	
6354	5256	JMP PARTS2+4	
6355	6161	PR2INC,	PR4INC
6356	7175	CKSLOC,	BUFFRS+201

BACKTY	1375	BNFLS	2461	BTAD	2325
BEILOC	3732	BNSTPA	4651	BTCBL	2247
BEIYDI	2231	BLTNS	2311	BVFWD	2216
BEFLC	2233	IDCON	2225	BVBRKW	2121
BEFLC	2234	LOT	6222	BVCHG	2135
BKNTYP	4117	IRECD	2254	BVCFWD	2113
BLKBIT	2122	KM25	2325	BVCRPT	2156
BLKATS	2117	KNDATA	5145	BVCRSTP	2127
BLKFLG	2747	K2223	2276	BVCRAT	2147
BLKFND	2221	K2226	4373	BVCRST	2222
BLKINC	2113	K2227	2273	BVCRPT	2257
BLOCKK	3772	K2212	2132	BVCRBL	2273
BNOTCN	3722	K2222	2274	BVSTOP	2232
BUFFRS	6774	K2232	2116	BVTEST	2222
BUFFR2	7175	K2237	2127	BVWAIT	2246
BUFFR3	7376	K2242	2121	NEWDRV	2237
CALOC	2232	K2252	2115	NOSE22	5526
CDRIVE	2257	K2252	1776	NUMWRD	5147
CHNGDR	2437	K2277	2271	OTHRIX	2721
CIPHER	2211	K2122	2123	PARZ1	5642
CKSERR	6312	K2121	2124	PARLOC	5756
CKSLOC	6356	K2142	2155	PARREZ	6273
COCOMP	5231	K2172	2122	PARRE1	6342
CODATA	5222	K2222	2275	PARRE2	6345
COERRO	5251	K2224	2125	PARRE3	6156
COERRI	5127	K2212	1262	PARRE4	6114
COFLAG	5152	K2214	2123	PARRE5	5466
COINCR	5224	K2215	1257	PARTAB	4672
COLOOP	5217	K2222	1772	PARTST	5622
COMBIT	2262	K2242	2272	PARTS1	6222
CRLFLE	1242	K2422	2122	PARTS2	6252
DAPSAV	4122	K2622	2344	PARTS3	6322
DATA22	2242	K2622A	1773	PARTS4	6222
DIRECT	2223	K2624	2265	PARTS5	6245
DIRFLG	2114	K2614	2121	PARTS6	6273
DIRSAV	2112	K1222	1774	PARWER	5744
DOLOOP	2222	K262	1535	PARWL1	5653
DOTHEM	2112	K2722	3542	PARWL2	5723
DRIVTY	2245	K2721	2122	PATNUM	4572
DRVTYP	2743	K3522	1777	PATTBA	1771
ERRSTP	2244	K4222	2266	PATTBL	4565
ERSSTA	2253	K4215	2276	PNTPS	2222
ERSTP	1312	K5276	2126	POSITN	2222
EZBIT	2112	K5277	2277	POSSAV	2111
EZERR	3366	K7222	2345	PREBLK	2752
EZERRA	3436	K7472	2673	PRT5EZ	6277
EZTYPE	4267	K7475	3131	PRWAF	6111
FORDTY	1276	K7577	2264	PR2INC	6355
FWDTYP	4111	K7622	4566	PR4INC	6161
GBKWI	3214	K7622A	6175	PTABLE	4724
GETMIN	2312	K7622B	5757	RDATTY	2246
GET4IN	2172	K7722	2272	RDCEND	4515
GNPAT2	4622	K7732	1775	RDCLEP1	4472
GNPAT1	4624	K7762	2263	RDCOMP	4451
GNPAT2	4612	K7767	2267	RDCPAS	4564
GNPAT3	4614	K7772	4374	RDCSOP	1667
GNPAT4	4622	LSTPLK	2224	RDCO1	1723
GNPAT5	4632	MSBITS	2261	RDCEND	4531
GNPAT6	4636	MVBKWD	2223	RDSERR	5522
GNPAT7	4644	MVCHNG	2274	RDSWLP	4562

RDSWS	5400	SRCHER	0746	TY3KW	1545
RDSW32	5434	SRCHIT	0035	TYCHAR	1230
RECORD	0020	SRCHTY	0052	TYCONT	1261
RECRDK	0026	SRCHI	2400	TYCOVR	1273
REFLGS	4734	SRCH2	2500	TYDATA	1475
REPOSI	0400	SRCNCK	0655	TYDIR	1536
RERFLG	4567	SREZTS	0721	TYDRV	1511
RESBEZ	4336	SRHER	1000	TYFWD	1555
RESBK3	4320	SRSCOP	1734	TYINTX	2733
RESBKW	4276	SRTAFN	0634	TYMODE	1563
RESFEZ	4260	SSTA3W	3651	TYMOVE	1400
RESFWD	4213	SSTAER	4000	TYPCON	0051
RESFWF	4232	SSTA EZ	3437	TYPT EX	0056
RESXIT	4250	SSTA FW	3600	TYRALL	1433
RESYEZ	4354	SSTANI	3254	TYRDAT	1424
RESYNC	4200	SSTAN2	3277	TYREAD	1460
REWIND	0036	SSTAN3	3322	TYSRCH	1411
RSFDRV	0417	SSTAN4	3400	TYTEXT	1200
RSFDR1	0435	SSTAIA	3267	TYWALL	1451
SAVPAD	0043	SSTA2A	3311	TYWDAT	1440
SABK	0106	SSTA3A	3334	TYWRIT	1466
SBAFK	0107	SSTA3B	3353	UKDATA	5146
SBCONS	0674	SSTA4A	3412	UNFUNC	0063
S3RECV	1331	SSTA4B	3424	WAETYP	5151
S3TYPE	0047	SSTBBE	3711	WAITI	0040
SCHIER	2433	SSTBKW	0055	WAITIN	2040
SCHIND	2461	SSTBR	3670	WCLOC	0007
SCH1OK	2444	SSTBSE	3703	WDATTY	0000
SCH1ST	2421	SSTER1	3210	WRDIR	5157
SCH2ER	2513	SSTEZA	3470	WRDND	5161
SCH2OK	2525	SSTEZ1	3501	WRSCOP	1600
SEARCH	0600	SSTFBE	3640	WRSCOI	1643
SEKONS	1071	SSTFR	3617	WRITEA	5301
SERCH2	2600	SSTFSE	3632	WRTEZT	5235
SERCMP	3070	SSTFWD	0054	WRTLPI	4421
SEREZ	2655	SSTRNA	3200	WRITSEZ	1655
SER2A	3000	STPERR	0044	WRTSLP	4561
SER2AI	2720	SYNCRE	4107	WRTSWA	5226
SER2LS	3111	TAPONT	0751	WRTSWS	5200
SER2NZ	2674	TEMPY	1770	WRTTST	4400
SER2ST	2637	TEMPI	4563	WRT32	5312
SER2TY	3123	TSRDEZ	5452	WTHALF	0041
SFABK	0105	TSTATB	1316	WTIJMP	0272
SFAFK	0104	TSTTBL	0277	WTINT	0234
SIERRO	2565	TXSTOR	1307	WT500	0313
		TYALL	1503		