

Latest Rev
April 6/74 *004*

IDENTIFICATION

PRODUCT CODE: MAINDEC-Ø8-DITCA-A-D
REPLACES MAINDEC-Ø8-D3BD
PRODUCT NAME: TC01*⁸ BASIC EXERCISER
DATE: MAY 31, 1972
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: JOHN VROBEL

TC Ø8 BE

COPYRIGHT © 1972
DIGITAL EQUIPMENT CORPORATION

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.

*This program can also be used on the TC08 DECTape control.
All tests are applicable, merely substitute "TC08" for "TC01" in text.

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

*CAUSE AC TO COUNT
FROM 0 TO 2701 (BLOCKS)
BACK AND FORTH.*

4.3 Program and/or Operator Action

- Place the select address for the routine desired in the SWITCH REGISTER and press LOAD ADDRESS.
- Set SWITCH REGISTER bits 0 to 7 to select drives. (Any configuration except all 0s is valid.)
- Press Start. The static register test will be run on status register A. and B. The processor should halt at address 0223 with bits 0 to 7 of the switch register displayed in the AC. For all error halts other than mentioned in 4.3 section D, consult the listing.

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

```
FWD or BKWD  
D O D O
```

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.

at start of tape

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

at end of Tape

```
"B"KWD
"W"AIT 0020
"F"WD
"W"AIT 0040
```

This program eventually moves the tape away from end zone. If BKWA wait is longer than FWD wait it will move back to the end zone.

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.) *which means if the wait is equal it will not move away from the end zone. (Wrong again)*

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 DECTape 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 typeout 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 Typeouts - The first three lines of the read-data typeouts are in the same format as the search typeouts. 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 typeouts 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 typeout is used by the Parity Generation Test. The last line of this typeout 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 typeout is also used by the Parity Generation Test and could follow the one above. The typeout indicates that a parity error should have been generated, but was not received. Again, the last line of the typeout indicates the data pattern written to test the parity circuitry. Notice the complement obverse relationship between the two data typeouts. 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 typeouts, 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 typeout 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
7700 CK SUM

(First word of block, as read from tape)
(As read from tape in READ ALL, in this case should equal 5700).

DRIVE 1
CKSUM ERROR
5757 DATA
5700 CK SUM

(As read from tape)
(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
7773 UNKNOWN
7000 ADDRS KNOWN

(Data written)
(Data read)
(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
END ZONE

NO BLOCKS
XXXX FIRST BLOCK
TYPE C FOR COMPARE
ALL OTHERS PRINT
```

```
(If an error status and then repeat the initial typeout)
(If the tape went into end zone before 129 blocks were
read)
(If no block numbers were read)
(First block number read)
```

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	4440
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 typeout is generated indicating PARITY ERROR EXPECTED, the contents of the LPB can be determined by the following procedure:

- a. The typeout includes the first data word of the block if read forward or the last word of the block if read backward (acutally 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 of 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 equaled 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.

9.10 Static Register Test

This test is run automatically prior to all the tests listed above 9.1-9.9. The static register test verifies the ability of status register A and status register B to accept various Data Patterns, IOT 766 to load AC to status register A, IOT 774 to load AC to status register B, IOT 764 to XOR AC to status register A, IOT 761 to "OR" information from status register A to the AC, IOT 772 to "OR" information from status register B to the AC, and IOTS 774, 764, and 766 to clear the AC after their execution.

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

7175 BUFFER2=BUFFER2+201
7376 BUFFER3=BUFFER2+201

0001 0001 *1 JMP 1 2 /FOR INTERRUPTS
0001 5402

0020 0020 *20 RECORD, 0 /BLOCK OPERATED ON
0021 0000 BLKFN, 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, BLKFN /FOR SRCH CA
0026 0020 RECRDK, RECORD /FOR TYPEOUTS

/ADDRESSES FOR INDIRECT TAD AND DCA

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

/SUBROUTINE ADDRESS

0035 0600 SRCHIT, SEARCH
0036 0400 REWIND, REPOSI
0037 0440 NEWDRV, CHNGDR
0040 0234 WAITI, WTINT
0041 0313 WTHALF, WT500
0042 5000 DATAO, CODATA
0043 4100 SAVPAD, DAPSAV
0044 1310 ERRSTP, ERSTP
0045 1511 DRIVTY, TYDRV
0046 1424 RDATY, 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 TYRTEX, TYTEXT

/TEMP STORAGE FOR DRIVE SELECTION

0057 0000 DRIVE, 0
 0060 0000 UNFUNC, 0
 0061 0000 MSBITS, 0
 0062 0000 COMBIT, 0
 /OTHER CONSTANTS

/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 SFAPK, 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
 0131 2525 K2525, 2525
 0132 0000 DTSAV, 0
 0133 0000 DTCNT, 0
 0134 7574 K7574, 7574
 0135 1077 XDTCHK, DTCHK
 0136 2744 XCHKGO, CHKGO

/DRIVE NUMBER AND
 /POSITIONED FOR STAT A
 /DRIVES SELECTED

0137 5331 XCHKB, CHKB
0140 0070 K0070, 0070
0141 7600 Z7600, 7600
0142 5000 K5000, 5000

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

*200
0200 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

CIPHER, 0 JMS I XDTCHK /STATIC REGISTER TEST ON STATUS A + B

0211 0000 OSR AND K7760
0212 4535 SEA +2
0213 7404 JMP HLTNS
0214 0063 DCA MSBITS
0215 7440 IOT 774
0216 5220
0217 5311
0220 3061
0221 6774

0222 1061 TAD MSBITS
0223 7402 HLT
0224 7200 CLA
0225 1211 TAD CIPHER
0226 1064 TAD K7777
0227 1277 TAD TSTBL
0230 3232 DCA +2
0231 4633 JMS I +2
0232 5700 JMP I TSTBL+1 /GO TO TEST SELECTED
0233 0420 RSPDRV

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

0234 5234 TAD WTJMP
0235 1272 DCA 1
0236 3001 TAD WTJMP+1
0237 1273 DCA 2
0240 3002 DCA WTJMP+2
0241 3274 TAD K4215
0242 1276 DCA WTJMP+3
0243 3275 ION
0244 6001 ISE WTJMP+2
0245 2274 JMP +1
0246 5245 ISE WTJMP+3
0247 2275

/WAIT A MAXIMUM
/OF 35 SECONDS
/FOR AN INTERRUPT

0250	5245	JMP .+3	
0251	6002	IOF	
0252	7402	HLT	
0253	5252	JMP .+1	
0254	6771	IOT 771	/DIF OR 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	JMP I 2	
0273	0254	IRECD	
0274	0000	0	
0275	0000	0	
0276	4215	K4215,	
0277	5700	TSTTBL, JMP I ;+1	

Handwritten notes:
 19151
 19151 R

/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	HLT	/TEST SELECTED	
0312	5212	JMP CIPHER +1	/NOT AVAILABLE	
0313	5313	JMP .		
0314	7200	CLA	/TIME OUT	
0315	3274	DCA WTIJMP+2		
0316	1325	TAD KM25		
0317	3275	DCA WTIJMP+3		
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,		

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

```

0400 *400
0400 REPOSI: JMS RSFDRV /RESET POINTRS TO FIRST DRIVE
0401 TAD K0624 /MOVE BACKWARDS
0402 TAD UNFUNG /POSITIONED UNIT NO
0403 IOT 762 /CLEAR STATUS A
0404 IOT 764 /XOR STATUS A
0405 JMS I WAITI /INDICATE EXPECT END
0406 S001 /NOT STATUS EXPECTED
0407 HLT
0410
0411 CLA CMA /INDICATE END ZONE
0412 DCA I POSITN
0413 CLA CMA
0414 DCA I DIRECT /INDICATE BACKWARDS
0415 JMS CHNGDR /SET UP NEXT DRIVE
0416 JMP REPOSI*2 /REWIND NEXT DRIVE
0417 JMP I REPOSI /GOT ALL DRIVES, EXIT

```

/RESET CURRENT DRIVE POINTERS TO
/FIRST DRIVE SELECTED

```

0420 RSFDRV, JMP . /SET INITIALLY TO 0
0421 CLA
0422 DCA CORIVE /THIS DRIVE SELECTED
0423 TAD K4000 /YES, SET POINTER
0424 DCA COMBIT
0425 TAD MSBITS
0426 AND COMBIT
0427 SEA CLA /MOVE COMPARE BIT
0430 JMP RSFOR1 /INCREMENT DRIVE NUM.
0431 TAD COMBIT
0432 CLL RAR
0433 DCA COMBIT
0434 ISZ CORIVE
0435 JMP RSFDRV*5

```

/HAVE FOUND FIRST DRIVE SELECTED
RSFOR1, JMS GNPTRS /GENERATE CONTROL POINTERS
JMP I RSFDRV /EXIT

```

0436 4262
0437 5620
0440 /SELECT NEXT DRIVE OR
0441 /RESET TO FIRST DRIVE AND SKIP
0442 CHNGDR, JMP .
0443 CLA
0444 TAD COMBIT /GET DRIVE COMPARE BIT
0445 CLL RAR /MOVE IT TO NEXT

```

```

0444 0067 AND K7767
0445 7440 /LAST DRIVE NUM 7
0446 5252 /NO
0447 4220 /RESET TO FIRST
0450 2240 /INCR, EXIT, SKIP
0451 5640 /EXIT
0452 3062 /THIS DRIVE SELECTED
0453 2057 /NO
0454 1062 /GENERATE DRIVE POINTERS
0455 0061
0456 7650
0457 5241
0460 4262
0461 5640

```

```

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

```

```

GNPTRS, JMP .
0462 5262 /DRIVE NUMBER
0463 1057 /POSITION TO BITS 0.i.i.2
0464 7112 /DRIVE NUMBER
0465 7012 /DRIVE NUMBER
0466 3060 /+ POS, PNTR ADDR:
0467 1057 /FOR INDIRECTS
0470 1301
0471 3022
0472 1057
0473 1312 /+ DIRECTION PNTR
0474 3023 /FOR INDIRECTS
0475 1323
0476 1057
0477 3024
0500 5662

```

```

PNTRS, .+1

```

```

0501 0502 /TO GET LAST RECORD NUMBER
0502 0000 /FOR DRIVE 8
0503 0000 /1
0504 0000 /2
0505 0000 /3
0506 0000 /4
0507 0000 /5
0510 0000 /6
0511 0000 /7
0512 0513 /TO GET LAST DIRECTION
0513 0000 /DIRECTION - UNIT 8
0514 0000 /1
0515 0000 /2
0516 0000 /3

```


0636 3352 DCA TAPONT
 0637 6761 IOT 761
 0640 7040 CMA
 0641 0075 AND K0200 /IN CASE MOTION=0
 0642 1102 TAD K0400
 0643 6764 IOT 764 /CHANGE DIRECTION
 0644 7040 CMA
 0645 3350 DCA BLKFLG
 0646 4440 JMS I WAIT
 0647 0001 1
 0650 5747 JMP I SRCHER /HAS TO BE NORMAL
 0651 4256 JMS SRCNCK /OR ERROR
 0652 5600 JMP I SEARCH /FOUND BLOCK, EXIT
 0653 5747 JMP I SRCHER /WENT PAST, ERROR
 0654 6764 IOT 764 /NOT THERE YET
 0655 5246 JMP .07

SRCNCK, JMP .
 0656 5256 ISZ BLKFLG /FIRST BLOCK IN
 0657 2350 SKP /NO
 0660 7410 JMP SBCONS
 0661 5275 IOT 761
 0662 6761 AND K0400
 0663 0102 SZA CLA
 0664 7040 CMA
 0665 7040 SNA
 0666 7450 IAC /FORWARD IS
 0667 7001 TAD PREBLK /+1
 0670 1351 CMA IAC
 0671 7041 TAD BLKFND /BLOCKS SEQUENTIAL
 0672 1021 SZA CLA
 0673 7640 JMP I SRCHER /NO, ERROR
 0674 5747

SBCONS, TAD TAPONT
 0675 1352 CMA IAC
 0676 7041 TAD BLKFND
 0677 1021 SNA
 0700 7450 JMP I SRCNCK /FIND BLOCK YET
 0701 5656 ISZ SRCNCK /YES, TA OR EXIT
 0702 2256 CLL /STEP ADDR
 0703 7100 SPA CLA
 0704 7710 STL
 0705 7120 IOT 761
 0706 6761 AND K0400 /L=1 IS BLK FND LESS
 0707 0102 SZA CLA
 0710 7640 JMP .+6 /FORWARD
 0711 5317 SEL /NO BACKWARD
 0712 7430 ISZ SRCNCK /FORWARD AND BLKFND
 0713 2256 TAD BLKFND /LESS IS NOT THERE YET
 0714 1021 DCA PREBLK
 0715 3351 JMP I SRCNCK
 0716 5656 SNL /BACKWARD AND BLKFND
 0717 7420

0720 2296 ISZ SRCNCK /LESS IS GONE PAST
0721 5314 JMP .+5

0722 6772 SREZTS, IOT 772 /NOT END ZONE, ERROR
0723 0110 AND EZBIT /BLOCK 0 OR 1
0724 7650 SNA CLA /BLOCK 2700 OR 2701
0725 5747 JMP I SRCHER
0726 1392 TAD TAPONT
0727 7510 SPA

0730 5334 JMP .+4 /SET MOTION
0731 1126 TAD K5076 /WAIT FOR EZ AGAIN
0732 7710 SPA CLA /DO TURN AROUND
0733 5343 JMP .+10 /IF EZ WAS
0734 7200 CLA /FIRST INT WAS VALID
0735 1075 TAD K0200 /IF NOT FIRS IS INVALID
0736 6764 IOT 764
0737 4440 JMS I WAITI

0740 5000 5000
0741 5747 JMP I SRCHER
0742 5235 JMP SRTAFN
0743 1390 TAD BLKFLG
0744 7700 SMA CLA
0745 5747 JMP I SRCHER
0746 5235 JMP SRTAFN
0747 1000 SRCHER, SRHERR
0750 0000 BLKFLG: 0
0751 0000 PREBLK: 0
0752 0000 TAPONT: 0

/SEARCH ERROR TYPEOUT

1000 *1000 /TYPE BLOCK SEARCHED
4444 SRHERR, JMS I ERRSTP
4452 JMS I SRCHTY
1026 TAD RECRDK
4451 JMS I TYPCON
4456 JMS I TYPTEX
1005 0042 42
5457 5457
4353 4353
67 67
4156 4156
6445 6445
4400 4400
7700 7700
TAD DIRFLG
SNA CLA
JMS I BACKTY
JMS I BACKTY
JMS I FORDTY
TAD IDCON
1023 1023

1024 4451 JMS I TYPCON /TYPE BLOCK FOUND
1025 4456 JMS I TYPTX
1026 0042 42
1027 5457
1030 4353

1031 0046 46
1032 5765
1033 5644
1034 7700
1035 7040 CMA
1036 1671 TAD I SEKONS
1037 7710 SPA CLA
1040 5252 JMP ,+12
1041 1272 TAD SEKONS+1
1042 4451 JMS I TYPCON
1043 4456 JMS I TYPTX
1044 0054 54
1045 4163 4163
1046 6400 6400
1047 4254 4254
1050 5743 5743
1051 5377 5377
1052 2671 ISE I SEKONS
1053 7000 NOP
1054 1271 TAD SEKONS
1055 4451 JMS I TYPCON
1056 4456 JMS I TYPTX
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 \$BTYPE
1067 5670 JMP I ,+1
1070 0601 SEARCH+1
1071 0750
1072 0751
1073 1074
1074 0000
1075 1545
1076 1555

/BLKFLG /MORE THAN 1 BLOCK /NO
/PREBLK /TYPE LAST BLOCK /FOUND

/TYPE NUMBER OF /BLKS NUM READ

SEKONS, BLKFLG
PREBLK ,+1
0

BACKTY, TYBKW
FORDTY, TYFWD

/ROUTINE TO DO STATIC CHECK ON
/STATUS REGISTER A BITS 1-3 AND 5-9
/AND STATUS B BITS 6-8

DTCHK, 0

1077 0000
1100 7300
1101 3133
1102 7340
1103 4322

CLA CLL /ZERO PASS COUNTER

DCA DTCNT
CLA CLL CMA
JMS PATCHK

/CHECK ALL ONES PATTERN

1104 4322 JMS PATCHK
 1105 1131 TAD K2525
 1106 4322 JMS PATCHK
 1107 1131 TAD K2525
 1110 7040 CMA
 1111 4322 JMS PATCHK
 1112 2133 ISZ DTENT
 1113 5302 JMP DTCHK +3
 1114 1133 TAD DTENT
 1115 4322 JMS PATCHK
 1116 2133 ISZ DTENT
 1117 5314 JMP .+3
 1120 4537 JMS I XCHKB
 1121 5677 JMP I DTCHK

/PATTERN CHECKER
 /PATCHK. 0

1122 0000 AND K7574
 1123 0134 DCA DTSAV
 1124 1132 TAD DTSAV
 1125 1132 IOT 766
 1126 6766 SEZ
 1127 7440 HLT 761
 1130 7402 CIA
 1131 6761 TAD DTSAV
 1132 7041 SEZ
 1133 1132 HLT
 1134 7440 TAD DTSAV
 1135 7402 HLT
 1136 1132 CMA
 1137 7040 IOT 761
 1140 6761 CMA
 1141 7040 SEZ
 1142 7440 HLT
 1143 7402 IOT 764
 1144 6764 IOT 761
 1145 6761 CIA
 1146 7041 TAD DTSAV
 1147 1132 SEZ
 1150 7440 HLT
 1151 7402 TAD DTSAV
 1152 1132 IOT 764
 1153 6764 SEZ
 1154 7440 HLT
 1155 7402 CLA CMA
 1156 7240 IOT 761
 1157 6761 CMA
 1160 7040 SEZ
 1161 7440 HLT
 1162 7402 TAD DTSAV
 1163 1132 IOT 766
 1164 6766 IOT 762
 1165 6762 IOT 761
 1166 6761 SEZ
 1167 7440

11 111 111 111
 010 000 000 100

/BITS 4,10, AND 11 NOT USED
 /STORE PATTERN
 /LOAD STATUS A
 /IOT 766 SHOULD CLEAR AC
 /IOT 766 DID NOT CLEAR AC
 /"OR" STATUS A TO AC
 /THIS IS THE GOOD STATUS ANSWER
 /WAS DTSAV EQUAL TO STATUS A
 /STATUS A OR AC FAILED
 /"OR" STATUS A TO AC
 /AC SHOULD NOW BE ZEROS
 /DID STATUS A "OR" TO ALL ONES
 /STATUS A OR AC FAILED
 /"XOR" AC TO STATUS A
 /"OR" STATUS A TO AC
 /DID STATUS CHANGE
 /STATUS A OR AC FAILED
 /CLEAR STATUS A WITH XOR
 /IOT 764 SHOULD CLEAR AC
 /AC OR STATUS A FAILED
 /OR STATUS A TO AC
 /WAS STATUS A ZERO
 /STATUS A OR AC FAILED
 /LOAD STATUS A
 /CLEAR STATUS A
 /GET STATUS A
 /DID IOT 766 CLEAR STATUS A

111 111 111 111
 000 000 000 100

/TYPE TEXT ROUTINE

```

1200 *1200
1201 TYTEXT, JMP .
1202 CLA
1203 TAD I TYTEXT
1204 CMA
1205 DCA TXSTOR
1206 ISZ TYTEXT
1207 TAD TXSTOR
1208 SZA
1209 JMP .+3
1210
1211 JMS CRLFLF
1212 JMP TYTEXT+1
1213 AND K7700
1214 SNA
1215 JMP I TYTEXT
1216 RTR
1217 RTR
1218 RTR
1219 JMS TYCHAR
1220 TAD TXSTOR
1221 AND K0077
1222 SNA
1223 JMP I TYTEXT
1224 JMS TYCHAR
1225 JMP TYTEXT+1
1226
1227 TYCHAR, JMP .
1228 CMA
1229 AND K0077
1230 TAD K0240
1231 TLS
1232 TSF
1233 JMP .-1
1234 CLA
1235 TCF
1236 JMP I TYCHAR
1237
1238 CRLFLF, JMP .
1239 TAD K0215
1240 TLS
1241 TSF
1242 JMP .-1
1243 CLA
1244 TAD K0212
1245 TLS
1200 /GET NEXT 2 CHARACTERS
1201 /MAKE =
1202
1203 /CARRIAGE RETURN = LINE FEED
1204 /NO
1205
1206 /CR LF
1207 /GET NEXT
1208 /CLEAR TO UPR CHAR
1209 /END OF MESSAGE
1210 /YES
1211 /MOVE
1212 /OVER
1213 /6 PLACES
1214 /OUTPUT
1215
1216 /END OF MESSAGE
1217 /YES EXIT
1218 /OUTPUT
1219 /GET NEXT 2
1220
1221 /MAKE + AGAIN
1222 /CLEAR TO LOWER 6
1223 /MAKE ASCII
1224 /OUTPUT
1225 /WAIT FLAG
1226
1227 /CLEAR FLAG
1228 /DO NEXT
1229
1230 /CARRIAGE RETURN
1231
1232 /LINE FEED
  
```

1252 6041 TSF
 1253 5252 JMP .-1
 1254 6042 TCF
 1255 7200 CLA
 1256 5642 JMP I CRLF LF
 1257 0215 K0215:
 1260 0212 K0212:

/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 CRLF LF /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
 1305 4230 JMS TYCHAR /MAKE = FOR
 1306 5673 JMP I TYCOVR /OUTPUT
 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 TYPCON
 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 .01

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

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

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

/TYPE READ ALL DIRECTION AND MODE
TYRALL, JMP
1433 JMS TYDRV
1434 JMS TYREAD
1435 JMS TYALL
1436 JMS TYDIR
1437 JMS TYMODE
1440 JMP I TYRALL
1441 5633

/TYPE WRITE DATA DIRECTION AND MODE
TYWDAT, JMP
1442 JMS TYDRV
1443 JMS TYWRIT
1444 4266

1445 4275 JMS TYDATA
1446 4336 JMS TYDIR
1447 4363 JMS TYMODE
1450 5642 JMP I TYWDAT

/TYPE WRITE ALL DIRECTION AND MODE

TYWALL, JMP
1451 5251 JMS TYDRV
1452 4311 JMS TYWRIT
1453 4266 JMS TYALL
1454 4303 JMS TYDIR
1455 4336 JMS TYMODE
1456 4363 JMP I TYWALL
1457 5651

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

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

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

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

/TYPE DRIVE AND NUMBER
TYDRV, JMP I TYPTX
1511 5311
1512 4456

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 .01
1530	7200	CLA
1531	4456	JMS I TYPTX
1532	7777	7777
1533	7700	7700
1534	5711	JMP I TYDRV
1535	0260	K260,

/TYPE FORWARDS OR BACKWARD

1536	5336	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	JMP I TYPTX
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	JMP I TYDIR
1556	4456	JMS I TYPTX
1557	4667	4667
1560	4400	4400
1561	7700	7700
1562	5755	JMP I TYFWD

/TYPE CONTINUOUS IF NOT NORMAL MODE

1563	5363	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
 1573 4445
 1574 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(0)

2000	*2000		
2001	MVTEST,	JMS I TYPTEX	
2002	7777	7777	/CR LF
2003	7777	7777	/CR LF
2004	7700		
2005	1032	TAD BFILOC	/SA OF EXECUTE TABLE
2006	3010	DCA 10	/FOR INDIRECTS
2007	1010	TAD 10	
2008	7001	IAC	
2009	3011	DCA 11	/FOR RESET ON RPT LOOP
2010	7040	CHA	
2011	3013	DCA 13	/FOR 1ST D
2012	1011	TAD 11	
2013	3012	DCA 12	/FOR RESET ON CHNG DRIVES
2014	4240	JMS WAITIN	/WAIT FOR INPUT
2015	4247	JMS MVEGUL	
2016	0304	304	/1ST IN = D
2017	3013	DCA 13	/0 TO CHAR COUNTER
2018	4247	JMS MVEGUL	
2019	0306	306	/= F FORWARD
2020	4247	JMS MVEGUL	
2021	0302	302	/= B BACKWARD
2022	4247	JMS MVEGUL	
2023	0323	323	/= S STOP
2024	4247	JMS MVEGUL	
2025	0303	303	/= C CHANGE
2026	4247	JMS MVEGUL	
2027	0327	327	/= H WAIT
2028	4247	JMS MVEGUL	
2029	0322	322	/= R REPEAT
2030	4247	JMS MVEGUL	
2031	0304	304	/= D DO LOOP
2032	4247	JMP MVTEST	
2033	5200		

/WAIT KEYBOARD INPUT
 WAITIN, JMP .
 2040 5240
 2041 6031
 2042 5241
 2043 6036
 2044 7041
 2045 3014
 CHA IAC
 KRB
 KSF
 JMP ,=1
 /14 = 2'S COMPLEMENT OF IN

2046 5640 JMP I WAITIN

```

/TEST FOR WHICH CHARACTER IN
MVEGUL. JMP .
TAD I 14
TAD I MVEGUL
ISZ MVEGUL
SNA CLA
JMP .+4
ISZ 13
NOP
JMP I MVEGUL
TAD 13
CMA CLA
SNA CLA
JMP I DOTHEM
TAD 13
CLL RAL
TAD MVRTBL
DCA I 14
TAD I 14
DCA I 10
JMP I 14

/RIGHT CHAR
/FOUND WHICH CHAR
/INC POINTER
/TEST NEXT CHAR
/FIRST IN # D
/FIRST # D DO LAST
/NUMBER TIMES 2
/FORM ADDRESS
/FOR INDIRECTS
/GET ROUTINE ADDRESS
/TO EXECUTE TABLE
/GO TO TYPEOUT
    
```

```

MVRTBL.
MV FWD
JMP MVG FWD
MVBKWD
JMP MVBKWK
MVSTOP
JMP MVBGSTP
MVCHNG
JMP MVBGCHG
MVWAIT
JMP MVBGWAT
MVRPT
JMP MVBGRPT
MVEND
JMP I .+1
DOTHEM, DOLOOP
/START TAPE FORWARD
/START BACKWARD
/STOP TAPE
/CHANGE DRIVES
/WAIT, MARK TIME
/REPEAT X TIMES
/END OF DO LOOP
/TO GET TO EXECUTE LOOP
    
```

```

/TYPE FORWARD
MVG FWD. 7777
4667
4477
JMS WAITIN
JMP MVRST
    
```

```

/TYPE BACKWARD
MVBKWK. 7777
    
```

2047 5247
 2050 1014
 2051 1647
 2052 2247
 2053 7650
 2054 5200
 2055 2013
 2056 7000
 2057 5647
 2060 1013
 2061 7040
 2062 7650
 2063 5712
 2064 1013
 2065 7104
 2066 1273
 2067 3014
 2070 1414
 2071 3410
 2072 5414

2073 2073
 2074 2216
 2075 5313
 2076 2223
 2077 5321
 2100 2230
 2101 5327
 2102 2274
 2103 5335
 2104 2246
 2105 5347
 2106 2257
 2107 5356
 2110 2305
 2111 5712
 2112 2200

2113 4456
 2114 7777
 2115 4667
 2116 4477
 2117 4240
 2120 5220

2121 4456
 2122 7777

2123 4253
2124 6744
2125 7700
2126 5317
JMP MVGFWD+4

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

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

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

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

2172 2310
GET4IN, GETMIN /TO GET 4 CHARACTERS

/DO LOOP, EXECUTE SELECTED SEQUENCE

```

2200 *2200
2201 DOLOOP, JMS I TYPTEX /TYPE DO
2202 7777
2203 4457
2204 7777
2205 7700 /SET ROUTINE POINTER TO START
2206 TAD BFILOC
2207 DCA 10
2208 DCA 11
2209 TAD I 10 /TO COUNT RPTS
2210 DCA I+3 /GET ROUTINE ADDRESS
2211 JMS I /FOR JMS I
2212 JMP I+2 /EXECUTE ROUTINE
2213 .
2214 JMP .
2215

```

```

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

/START FORWARD MOTION
MVFWD, JMP
2216 TAD K0200 /MOVE TAPE FORWARD
2217 1060 /+ DRIVE NUMBER
2220 IOT 766
2221 JMP I MVBKWD
2222

```

0075 = 0200
0060 = 0000
JMS MVWAIT = 4246

```

/START BACKWARD MOTION
MVBKWD, JMP
2223 TAD K0600 /MOVE BACKWARD
2224 1344 /+ DRIVE NUMBER
2225 1060
2226 IOT 766
2227 JMP I MVBKWD

```

2344 = 0600
2375 = 2000
JMS MVWAIT = 4246

```

/STOP TAPE
MVSTOP, JMP
2230 IOT 761 /READ STAT A
2231 6761 /CLEAR TO DRIVE NUM
2232 0345
2233 7041
2234 1060 /SAME DRIVE
2235 7640 /NOT SAME AS STAT A
2236 5243
2237 6761 IOT 761
2240 0075 AND K0200 /CLEAR AC TO MOTION BIT
2241 6764 IOT 764
2242 5630 JMP I MVSTOP
2243 1060 TAD UNFUNC
2244 6766 IOT 766
2245 5630 JMP I MVBKWD

```

7402

/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 L=1
 2254 2017 ISZ 17
 2255 5252 JMP L=3
 2256 5646 JMP I MVWAIT

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

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

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

2310 5310 /MAKE = CONSTANT OF 4 INPUTS
 2311 7201 GETMIN, JMP L=+1 WHEN L=1 GOT 4
 2312 3013 CLA IAC
 2313 6031 DCA 13
 2314 5313 KSF /WAIT KEYBOARD
 2315 6036 JMP L=1 /READ KEYBOARD
 2316 6046 KRB /OUTPUT
 2317 6041 TLS
 TSF

ATC01 BASIC EXERCISER TAPE 1

PAL10 V141

```

2320 5317
2321 7040
2322 0073
2323 3014
2324 1013
2325 7104
2326 7006
2327 1014
2330 3013
2331 7420
2332 5313
2333 1013
2334 7001
2335 7450
2336 7040
2337 3410
2340 4456
2341 7777
2342 7700
2343 5710
2344 0600
2345 7000

JMP =1
CMA K0007
AND K0007
DCA 14
TAD 13
CLL RAL
RTL
TAD 14
DCA 13
SNL GETMIN+3
JMP GETMIN+3
TAD 13
IAC
SNA
CMA I 10
DCA I TYPTEX
JMS I TYPTEX
7777
7700
JMP I GETMIN
600
K0600: 7000
K7000:

```

```

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

```

```

2400
2401 4436
2402 3114
2403 3020
2404 4435
2405 5221
2406 4444
2407 4452
2410 1026
2411 4451
2412 1025
2413 4451
2414 6772
2415 3021
2416 1025
2417 4451
2420 5202
2421 3422
2422 3423
2423 2020
2424 7040
2425 3427
2426 6764
2427 4440

*2400
SRCH1:
JMS I REWIND
JMS I REWIND
DCA DIRFLG
DCA RECORD
JMS I SRCHIT
JMP SRCH1
JMS I ERRSTP
JMS I SRCHTY
TAD RECRD
JMS I TYPCON
TAD IDCON
JMS I TYPCON
IOT 772
DCA BLKFND
TAD IDCON
JMS I TYPCON
JMP SRCH1+2

SCH1ST:
DCA I POSITN
DCA I DIRECT
ISZ RECORD
CMA
DCA I WCLOG
IOT 764
JMS I WAITI

/INTO EZ TWICE
/RECORD 0
/FIND IT
/YES /NO
/BLOCK LOOKED FOR
/BLOCK FOUND
/RD STAT B
/TYPE STAT B
/FOUND 0
/FORWARD
/RECORD LOOKED FOR +1
/SET WC TO -1
/SET WC ENABLES

```

```

2430 0001
2431 7410
2432 5244
2433 6772
2434 0110
2435 7650
2436 5206
2437 1020
2440 1077
2441 7700
2442 5262
2443 5206

1
SKP SCH10K /INTERRUPT OK
JMP IOT 772
SCH1ER, AND EZBIT
SNA CLA /END ZONE
JMP SRCH1+6 /NO TYPE ERROR
TAD RECORD
TAD K507X
SMA CLA /DONE ALL BLOCKS
JMP SCH1ND /YES
JMP SRCH1+6 /NO, ERROR

```

```

2444 1021
2445 7041
2446 1020
2447 7640
2450 5254
2451 4536
2452 2422
2453 5233
2454 4366

SCH10K, TAD BLKFN
CMA IAC /MAKE
TAD RECORD /RIGHT BLOCK
SEA CLA /NO
JMP I+4 /CHECK GO BIT IF BLOCK 200-2000
JMS I XCHKGO /NEW POSITION
ISE I POSI+2 /TEST NEXT BLOCK
JMP SCH1ST+2
JMS SIERRA

```

```

2455 2422
2456 2020
2457 4435
2460 5244
2461 5233
2462 4437
2463 5202
2464 7040
2465 3020
2466 1125
2467 1060
2470 6766
2471 4440
2472 5001
2473 7000
2474 4437
2475 5266
2476 2020
2477 7410

ISE I POSI+2 /RESYNC ON NEXT BLOCK
ISE RECORD /FOUND OK
JMS I SRCHIT /TEST FOR END ZONE
JMP SCH10K
JMP SCH1ER /RPT NEXT DRIVE
JMS I NEWDRV
JMP SRCH1+2
CMA
DCA RECORD /MOVE DRIVE INTO EZ
TAD K0204
TAD UNFUNC
IOT 766
JMS I WAITI
5001
NOP
JMS I NEWDRV
JMP SCHIND+4 /DONE ALL TWICE
ISE RECORD /YES
SKP /MAKE 2ND MOVE INTO EZ

```

```

2500 5266
2501 1100
2502 3020
2503 1101
2504 1060
2505 6766
2506 7040
2507 3427
2510 4440
2511 0001

SRCH2,
TAD K2701
DCA RECORD /SRCH BACKWARD
TAD K0614 /+ DRIVE NUMBER
TAD UNFUNC
IOT 766
CMA
DCA I WCL0G
JMS I WAITI
1

```

2512 7410 SKP
 2513 5326 JMP SCH20K
 2514 4366 JMS SIERRO
 2515 7240 CLA CMA
 2516 1020 TAD RECORD
 2517 3020 DCA RECORD
 2520 1020 TAD RECORD
 2521 7710 SPA CLA
 2522 5357 JMP SIERRO-7
 2523 7040 CMA
 2524 3114 DCA DIRFLG
 2525 4435 JMS I SRCHIT

SCH2ER,
 SCH20K,
 SIERRO,
 SRCH1

2526 1020 TAD RECORD
 2527 7041 CMA IAC
 2530 1021 TAD BLKFN
 2531 7640 SZA CLA
 2532 5314 JMP SCH2ER
 2533 6764 IOT 764
 2534 7040 CMA
 2535 3427 DCA I WCLOC
 2536 1020 TAD RECORD
 2537 3422 DCA I POSITN
 2540 7040 CMA
 2541 1020 TAD RECORD
 2542 3020 DCA RECORD
 2543 4440 JMS I WAITI
 2544 0001 1
 2545 7410 SKP

SCH20K,
 SIERRO,
 SRCH1

2546 5326 JMP SCH20K
 2547 6772 IOT 772
 2550 0110 AND E2BIT
 2551 7650 SNA CLA
 2552 5314 JMP SCH2ER
 2553 1020 TAD RECORD
 2554 7040 CMA
 2555 7640 SZA CLA
 2556 5314 JMP SCH2ER
 2557 4437 JMS I NEWDRV
 2560 5301 JMP SRCH2
 2561 7004 CLA OSR
 2562 7010 RAR
 2563 7620 SNL CLA
 2564 7402 HLT
 2565 5200 JMP SRCH1
 2566 5366 JMP
 2567 4444 JMS I ERRSTP
 2570 4452 JMS I SRCHTY
 2571 1026 TAD RECRDK
 2572 4451 JMS I TYPCON
 2573 1025 TAD IDCON
 2574 4451 JMS I TYPCON
 2575 4447 JMS I SBTYPE
 2576 5766 JMP I SIERRO

SIERRO,
 SRCH1

2576 5766 JMP I SIERRO

SIERRO,
 SRCH1

/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
 SERCH2. JMS I DRVTYP /TYPE DRIVE AND NUMBER
 JMS TYINTX /TYPE TYPE IN
 JMS I TYPTX
 0046 /F FOR FWD
 0046
 5762
 0046
 6744
 7700

JMS OTHRTX /TYPE ALL OTHERS
 JMS I TYPTX /BACKWARD
 4253
 6744
 0077 /WAIT K
 KSF
 JMP .-1
 KRB
 TLS
 TSF
 JMP .-1
 TCF
 TAD K7472
 SEA CLA
 TAD K0400
 TAD K0214

/NF /NO PREP BACKWARD

TAD UNFUNC /START SEARCH
 IOT 766
 TAD BF1LOC
 DCA 10
 TAD K7577
 DCA 11

SER2ST. CMA /HC = -1
 DCA I WCLOC /TO BLKFND
 TAD IDCON
 DCA I CALOC
 JMS I WAITI
 1
 JMP SEREZ /NO NORMAL INT

2600
 4743
 4333
 4456
 0046
 0046
 5762
 0046
 6744
 7700
 4321
 4456
 4253
 6744
 0077
 6031
 5216
 6036
 6046
 6041
 5222
 6042
 1273
 7640
 1102
 1103

2631 1060
 2632 6766
 2633 1032
 2634 3010
 2635 1064
 2636 3011
 2637 7040
 2640 3427
 2641 1025
 2642 3430
 2643 4440
 2644 0001
 2645 5255

2646 1021 /STORE BLOCK NUMBER
 2647 3410

```

2650 2011
2651 7410
2652 5720
2653 6764
2654 5237

2655 6772
2656 0110
2657 7640
2660 5274
2661 6772
2662 3020
2663 1026
2664 4451
2665 4456
2666 0063
2667 6441
2670 6400
2671 4277
2672 5200
2673 7472

SEREZ,
IOT 772
AND EZBIT
SEA CLA
JMP SER2NZ
IOT 772
DCA RECORD
TAD RECRDK
JMS I TYPCON
JMS I TYPTX
0063
6441
6400
4277
JMP SERCH2
K7472, 7472

```

```

/DONE 129
/RESET WCE
/REPT
/READ B
/END ZONE
/YES

```

/TYPE STATUS B

/START OVER

```

/TYPE END ZONE
SER2NZ, JMS I TYPTX

```

```

2674 4456
2675 7777
2676 4556
2677 4400
2700 7257
2701 5645
2702 7700
2703 1011
2704 7041
2705 1064
2706 7640
2707 5720
2710 4456
2711 7777
2712 5657
2713 0042
2714 5457
2715 4353
2716 6377
2717 5200
2720 3000

```

```

/READ ANY AT ALL
/YES
/TYPE NO BLOCKS

```

/TO GET TO REST OF PROGRAM

```

SER2AI, SER2A

```

```

/TYPE ALL OTHERS
OTHRTX, JMS I TYPTX

```

```

2721 5321
2722 4456
2723 7777
2724 4154
2725 5400
2726 5764
2727 5045

```

2730 6263
2731 0077
2732 5721
JMP I 0THR1X

2733 5333
2734 4456
2735 7777
2736 6471
2737 6045
2740 0051
2741 5677
2742 5733
JMP I TYIN1X
JMS I TYPTEX
JMP I TYIN1X

2743 1511
DRVTYP, TYDRV
/ROUTINE TO CHECK GO BIT

2744 0000
2745 7300
2746 1021
2747 1142
2750 7630
2751 5744
2752 1021
2753 1141
2754 7620
2755 5744
2756 6761
2757 0075
2760 7650
2761 7402
2762 5744
CHKGO, 0
CLA CLL
TAD BLKFND
TAD K5000
SZL CLA
JMP I CHKGO
TAD BLKFND
TAD 27600
SNL CLA
JMP I CHKGO
IOT 761
AND K0200
SNA CLA
HLT
JMP I CHKGO
/GET BLOCK READ
/WAS IT GREATER THAN BLOCK 2000
/YES EXIT
/GET BLOCK READ
/WAS IT LESS THAN 0200
/YES EXIT
/GET GO BIT
/MASK GO BIT
/WAS GO BIT SET
/STATUS A OR GO BIT FAILED
/EXIT GO BIT O.K.

/REST OF SEARCH ROUTINE 2

3000 6761
3001 0075
3002 6764
3003 1011
3004 7041
3005 1064
3006 3011
3007 1032
3010 3010
3011 1010
3012 7001
3013 3012
3014 1012
3015 4451
3016 4456
*3000
SER2A, IOT 761
AND K0200
IOT 764
TAD 11
CMA IAC
TAD K7577
DCA 11
TAD BF1LOC
DCA 10
TAD 10
IAC
DCA 12
TAD 12
JMS I TYPCON
JMS I TYPTEX
/STOP TAPE
/NUMBER OF BLOCKS READ
/TO GET FIRST BLOCK
/TO GET 2ND BLOCK
/TYPE FIRST
/BLOCK NO READ

3017	0046		
3020	5162		
3021	6364		
3022	7700		
3023	4625	JMS I .+2	/TYPE TYPE IN
3024	7410	SKP	
3025	2733	TYINTX	
3026	4456	JMS I TYPTEX	/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	7410	SKP	
3041	2721	OTHRTX	
3042	4456	JMS I TYPTEX	/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	5293	JMP .=1	
3055	6042	TCF	
3056	1331	TAD K7475	
3057	7640	SEA CLA	/#C
3060	5323	JMP SER2TY	/NO, PRINT ALL
3061	6761	IOT 761	
3062	0102	AND K0400	
3063	7640	SEA CLA	/FORWARDS
3064	7040	CMA	/NO MAKE =1
3065	7450	SNA	/BACKWARDS
3066	7001	IAC	/NO MAKE +1
3067	3014	DCA 14	
3070	2011	/COMPARE BLOCKS FOR INCREMENTING OR DEC	
3071	7410	SERCMP, ISZ 11	/COMPARED ALL
3072	5311	SKP SER2LS	/NO
3073	1014	JMP SER2LS	/TYPE LAST BLOCK
3074	1410	TAD 14	/+ OR = /+ FIRST BLOCK
3075	7041	TAD I 10	/MAKE = /+ NEXT BLOCK
3076	1412	CMA IAC	/SHOULD BE 0
3077	7650	TAD I 12	/DO NEXT
3100	5270	SNA CLA	
3101	4456	JMP SERCMP	
3102	7777	JMS I TYPTEX	

3103 7700
 3104 1010
 3105 4451 /TYPE 1ST
 3106 1012 /TYPE 2ND
 3107 4451 /DO NEXT
 3110 5270

/TYPE OUT LAST BLOCK READ
 SER2LS, TAD 12 /TYPE BLOCK NUMBER
 JMS I TYPCON /LAST
 JMS I TYPTEX
 0054
 4163
 6477
 JMS I NEWDRV
 NOP
 JMP I ,+1
 SERCH2 /REPEAT FOR NEXT DRIVE

/PRINT ALL BLOCKS READ
 SER2TY, TAD 12 /ADDRESS
 JMS I TYPCON /TYPE BLOCK NUMBER
 ISZ 12 /DONE ALL
 ISZ 11 /NO
 JMP SER2TY /YES, DO NEXT DRV
 JMP SER2LS+2
 K7475, 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/T/L LENGTH OF TAPE AND TA ON BLOCK 2701
 /MOVE ALL DRIVES INTO REVERSE END ZONE

*3200
 SSTRNA, JMS I REWIND /MOVE INTO EZ 2
 JMS I REWIND /WAIT HALF SECOND
 JMS I WTHALF
 JMS I SAVPAD

/TEST TURN AROUND ON BLOCK 0 FIND 1 FWD
 CLA IAC
 DCA RECORD /FIND BLOCK 1 FWD
 JMS I SSTFWD
 JMP GBKH1
 SSTER1, JMS EEERR

3200 4436
 3201 4436
 3202 4441
 3203 4443
 3204 7201
 3205 3020
 3206 4454
 3207 5214
 3210 4366

3211 7000 NOP
3212 4453 JMS I ERSSTA
3213 5204 JMP .+7

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

3224 5211 /WAIT FOR E2 TA FIND 0 FWD
3225 3422 JMP SSTER1+1
3226 7040 DCA I POSITN
3227 3423 CMA
3230 1065 DCA I DIRECT
3231 1060 TAD K0604
3232 6766 TAD UNFUNC
3233 4440 IOT 764
3234 5001 JMS I WAITI
3235 5210 5001
3236 7040 JMP SSTER1
3237 3422 CMA
3240 4454 DCA I POSITN
3241 5245 JMS I SSTFWD
3242 5210 JMP .+4
3243 7000 JMP SSTER1
3244 5211 NOP
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

3254 4443 /TEST FORWARD START AFTER FORWARD
3255 1422 SSTAN1, JMS I SAVPAD
3256 1104 TAD I POSITN
3257 3020 TAD SFAPK
3260 4454 DCA RECORD
3261 5267 JMS I SSTFWD
3262 5666 JMP SSTA1A
3263 7000 JMP I .+4
3264 4453 NOP
JMS I ERSSTA

3265 5254 JMP SSTAN1
 3266 3437 SSTAEZ

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

/TEST BACKWARD START AFTER FORWARD

SSTAN2, JMS I SAVPAD
 3277 4443 TAD I POSITN
 3300 1422 TAD SBABK
 3301 1107 DCA RECORD
 3302 3020 JMS I 6STBRW
 3303 4455 JMP SSTA2A
 3304 5311 JMS EZERR
 3305 4366 NOP
 3306 7000 JMS I ERSSTA
 3307 4453 JMP SSTAN2
 3310 5277 TAD RECORD
 3311 1020 DCA I POSITN

SSTA2A, CMA I DIRECT
 3312 3422 TAD K0200
 3313 7040 IOT 764
 3314 3423 JMS I NEWDRV
 3315 1075 JMP SSTAN2
 3316 6764 JMS I WTHALF
 3317 4437
 3320 5277
 3321 4441

/TEST START FORWARD AFTER BACKWARD

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

/THEN TEST FORWARD TO BACKWARD TURN AROUND

JMS I ERSSTA
 3332 4453 JMP SSTAN3
 3333 5322 TAD RECORD
 3334 1020 DCA I POSITN
 3335 3422 DCA I DIRECT
 3336 3423 CMA I WCLC0
 3337 7040 IOT 764
 3340 3427 JMS I WAITI
 3341 6764
 3342 4440

```

3343 5001
3344 5331 JMP SSTAN3+7
3345 2422 ISZ I POSITN
3346 4455 JMS I SSTBRW
3347 5353 JMP SSTA3B
3350 5330 JMP SSTAN3+6
3351 7000 NOP
3352 5331 JMP SSTAN3+7
3353 1020 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 JMP
3367 7240 CMA CLA
3370 3021 DCA BLKFND
3371 5766 JMP I EZERR

```

SSTA3B.

EZERR.

*3400

```

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

```

/THEN TEST BACKWARD TO FORWARD TURN AROUND

```

3410 4453 JMS I ERSSIA
3411 5200 JMP SSTAN4
3412 1020 TAD RECORD
3413 3422 DCA I POSITN
3414 7040 CMA
3415 3423 DCA I DIRECT
3416 2020 ISZ RECORD
3417 4454 JMS I STFWD
3420 5224 JMP SSTA4B

```

SSTA4A.

SSTA4B.

```

3421 5206 JMP SSTAN4+6
3422 7000 NOP
3423 5207 JMP SSTAN4+7
3424 1020 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 SSTA4
3433 4441 JMS I WTHALF
3434 5635 JMP I .+1
3435 3254 SSTAN1

3436 3366 EZERR, EZERR

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

3437 1020 SSTA EZ, 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
3457 7040 CMA /IGNORE OTHER INTERRUPTS
3460 3422 DCA I POSI7N
3461 3423 DCA I DIRECT
3462 4437 JMS I NEWDRV
3463 5254 JMP .+7
3464 2020 ISE RECORD /2 PASSES
3465 7410 SKP /YES
3466 5263 JMP .+3
3467 4441 JMS I WTHALF

SSTA EZ, TAD K2700
JMS I SAVPAD /GO BACKWARD TO 2700
JMS I SSTRW

3474 5301 JMP SSTEZ1 /REACHED 2700 OK
3475 7000 NOP
3476 7000 NOP
3477 4453 JMS I ERSSYA
3500 5270 JMP SSTEZA
3501 1020 TAD RECORD
3502 3422 DCA I POSI7N /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
3537 3201 SSTRNA+1 /REPEAT TEST
3540 2700 K2700, 2700

/START STOP TURN AROUND TEST
/SEARCH FORWARD ROUTINE

3600 *3600
3601 5200 SSTAFW, JMP
3602 7240 CLA CMA
3603 3370 DCA BLOCKK
3604 1103 TAD K0214
3605 1060 TAD UNFUNC
3606 6766 IOT 766
3607 7040 CMA
3610 3427 DCA I WCLOC /SET WC = -1
3611 1025 TAD IDCON
3612 3430 DCA I CALOC
3613 4440 JMS I WAITI
3614 0001 1
3615 5232 JMP SSTFSE /STATUS B ERROR
3616 2370 ISZ BLOCKK
3617 5242 JMP SSTFBE+2
3620 1021 TAD BLKFND
3621 3371 DCA BLOCKK+1

Handwritten notes: 25 (7), 19151, 12, 11

```

3621 1021 TAD BLKFND
3622 7041 CMA IAC
3623 1020 TAD RECORD
3624 7450 SNA
3625 5600 JMP I SSTAFW
3626 7710 SPA CLA
3627 5240 JMP SSTFBE
3630 6764 IOT 764
3631 5206 JMP SSTAFW+6

```

```

/RIGHT BLOCK
/YES, EXIT
/BLOCK FOUND LESSER
/NO, ERROR
/RESET WC ENABLE

```

```

3632 6772 SSTFSE, IOT 772
3633 0110 AND EZBIT
3634 7650 SNA CLA
3635 2200 ISZ SSTAFW
3636 2200 ISZ SSTAFW
3637 5600 JMP I SSTAFW
3640 2200 ISZ SSTAFW
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 SSTAFW+1

```

```

/END ZONE EXIT
/ONLY 1 ISZ

```

```

/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 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
3677 7710 SPA CLA
3700 5311 JMP SSTBBE
3701 6764 IOT 764

```

```

/FOUND BLOCK EXIT

```

3702 5257 JMP SSTABW*6
 3703 6772 SSTBSE, IOT 772
 3704 0110 AND E2BIT
 3705 7650 SNA CLA
 3706 2251 ISZ SSTABW
 3707 2251 ISZ SSTABW
 3710 5651 JMP I SSTABW
 3711 2251 ISZ SSTABW
 3712 5306 SSTBSE, ISZ SSTABW
 3713 1021 JMP .*4
 3714 7040 TAD BLKFND
 3715 1371 CMA
 3716 7650 TAD BLOCKK*1
 3717 5270 SNA CLA
 3720 4322 JMP SSTBR
 3721 5252 JMS BNOTCN
 JMP SSTABW*1

/BLOCK NUMBERS ARE NOT
 /CONSECUTIVE ON START UP OR TURN 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 TYPTEX
 3730 7777 7777
 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 TYPTEX
 3741 0054 0054

/TYPE BLOCK NUMBER ERR;

/TYPE LAST

3742 4163 4163
 3743 6477 6477
 3744 1025 TAD IDCON
 3745 4451 JMS I TYPCON
 3746 4456 JMS I TYPTEX
 3747 0064 64
 3750 5051 5051
 3751 6377 6377

/TYPE CURRENT
 /BLOCK NUMBER

3752 1370 TAD BLOCKK
 3753 7001 IAC
 3754 3370 DCA BLOCKK
 3755 1373 TAD BLOCKK*3
 3756 4451 JMS I TYPCON
 3757 4456 JMS I TYPTEX
 3760 0043 43

/TYPE OUT BLOCK
 /COUNTER

```

3761 5664
3762 6277
3763 7604
3764 7012
3765 7620
3766 7402
3767 5722
3770 0000
3771 0000
3772 3771
3773 3770

```

/STOP ON ERROR
/YES
BLOCKK, 0
0
.01
.03

```

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

```

```

4000 4000
4001 4444
4002 4452
4003 1422
4004 7040
4005 7640
4006 5211
4007 4267
4010 5213
4011 1022
4012 4451
4013 1423
4014 7640
4015 5220
4016 4711
4017 5221
4020 4710
4021 4456
4022 0054
4023 4163
4024 6400
4025 6057
4026 6377
4027 1026
4030 4451
4031 4456
4032 0063
4033 4541
4034 6243
4035 5045
4036 4477
4037 1021
4040 7040
4041 7640
4042 5245
4043 4267
4044 5247

```

SSTAER, JMP .
JMS I ERRSTP /STOP TAPE LB STATB
JMS I SRCHTY /TYPE SEARCH
TAD I POSITN
CMA CLA
SEA CLA /TAPE WAS WHERE
JMP .03 /NOT END ZONE
JMS EZTYPE /TYPE END ZONE
JMP .03
TAD POSITN
JMS I TYPCON /TYPE LAST BLOCK
TAD I DIRECT
SEA CLA /DIRECTION WAS
JMP .03 /BACKWARD
JMS I FWDTP /TYPE FORWARD
JMP .02
JMS I BKWTYP /TYPE BACKWARD
JMS I TYPTEX /TYPE (LAST POS)
0054
4163
6400
6057
6377
TAD RECRDK
JMS I TYPCON /TYPE BLOCK1 LOOKED FOR
JMS I TYPTEX /TYPE (SEARCHED)
0063
4541
6243
5045
4477
TAD BLKFND
CMA
SEA CLA /WAS A BLOCK NUMBER
JMP .03 /YES
JMS EZTYPE
JMP .03 /TYPE END ZONE

4045	1025	TAD IDCON	/TYPE BLOCK NUMBER
4046	4451	JMS I TYPCON	
4047	4456	JMS I TYPTEX	/TYPE (FOUND)
4050	0046		
4051	5765		
4052	5644		
4053	7700		
4054	4447	JMS I SBTYPE	
4055	7604	CLA OSR	
4056	7006	RTL	
4057	7630	SEL CLA	/DELETE STOP AFTER ERROR
4060	5263	JMP .+3	/NO ERROR STOP

4061	1200	TAD SSTAER	/DISPLAY ADDRESS
4062	7402	HLT	
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 TYPTEX	
4071	7777		
4072	4556		
4073	4400		
4074	7257		
4075	5645		
4076	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 CLA	
4203	7650	SNA CLA	
4204	5354	JMP RESYEEZ	
4205	1112	TAD DIRSAV	
4206	7640	SEA CLA	

4207 5276 JMP RESBKW
 4210 1101 TAD K0614
 4211 1060 TAD UNFUNC
 4212 6766 IOT 766
 4213 7040 CMA
 4214 3427 DCA I WCLOC
 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 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

/TAPE GOES BACKWARD
 /FIRST TO RESYNC
 /FORWARD
 /=1 TO WC
 RESFWD,
 RESFWF,
 RESXIT,
 RESFEZ,
 /STOP TAPE
 /RESET POSITION
 /AND DIRECTION
 /POINTERS
 /END ZONE
 /NO, SOME OTHER, RESYNC
 /MOVE FARTHER
 /INTO EZ

4250 1075
 4251 6764
 4252 1111
 4253 3422
 4254 1112
 4255 3423
 4256 4441
 4257 5600
 4260 6772
 4261 0110
 4262 7650
 4263 5201
 4264 1065
 4265 1060
 4266 6766
 4267 4440
 4270 5001
 4271 5264
 4272 1103

```

4273 1060 TAD UNFUNC
4274 6766 IOT 766
4275 5233 JMP RESFWF+1

RESBKW,
4276 1103 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
4306 1373 TAD K0006
4307 1111 TAD POSSAV
4310 7041 CMA IAC
4311 1021 TAD BLKFND
4312 7700 SMA CLA
4313 5317 JMP .+4
4314 7200 CLA
4315 6764 IOT 764
4316 5301 JMP RESBKW+3
4317 1102 TAD K0400
4320 6764 IOT 764
4321 7040 CMA
4322 3427 DCA I WCLOC
4323 4440 JMS I WAITI
4324 0001 1
4325 5201 JMP RESYNC+1

RESBKW,
4326 1021 TAD BLKFND
4327 7041 CMA IAC
4330 1111 TAD POSSAV
4331 7450 SNA
4332 5250 JMP RESXIT
4333 7700 CLA SMA
4334 5201 JMP RESYNC+1
4335 5320 JMP RESBKW

RESBEZ,
4336 6772 IOT 772
4337 0110 AND EZBIT
4340 7650 SNA CLA
4341 5201 JMP RESYNC+1
4342 1125 TAD K0204
4343 1060 TAD UNFUNC
4344 6766 IOT 766
4345 4440 JMS I WAITI
4346 5001 5001
4347 5201 JMP RESYNC+1
4350 1101 TAD K0614
4351 1060 TAD UNFUNC
4352 6766 IOT 766
4353 5321 JMP RESBKW+1

/NOT NORMAL STAT, TRY AGAIN

/IN POSITION YET
/YES
/GO PAST AGAIN
/YES, TRY AGAIN
/NO, WAIT FOR NEXT BLOCK

/END ZONE
/NO, TRY AGAIN

/MOVE INTO EZ AGAIN

/NOW START BACKWARDS

```

4354 1112
4355 7640
4356 1102
4357 1125
4360 1060
4361 6766
4362 4440
4363 5001
4364 5354
4365 1075
4366 6764
4367 4440
4370 5001
4371 5365
4372 5251
4373 0006
4374 7772

/PUT TAPE BACK INTO END ZONE
/LEZ OR TEZ
RESYEZ. TAD DIRSAV
SZA CLA /BACKWARD # NO SKIP
TAD K0400 /YES BACKWARD
TAD K0204
TAD UNFUNG
IOT 766
JMS I WAITI
5001
JMP RESYEZ
TAD K0200
IOT 764
JMS I WAITI
5001
JMP .04
JMP RESXIT+1
K0006, 6
K7772, 7772

/MOVE INTO EZ TWICE

PAUSE

/TC01 BASIC EXERCISER - TAPE 3A
/WRITE BASIC DATA PATTERNS
/READ VERIFY WRITE FORWARD
/READ BACKWARD, FORWARD, WRITE BACKWARD
/READ FORWARD, BACKWARD

4400
4401 0073
4402 3370
4403 1370
4404 1365
4405 3363
4406 1763
4407 3363
4410 4763

4411 7040
4412 3424
4413 3422
4414 4437
4415 5211
4416 3114
4417 7001
4420 3113
4421 1113
4422 7510
4423 7200
4424 1424
4425 3020
4426 4435
4427 1115

*4400
WRTTST, LAS /GET SWITCHES
AND K0007 /MASK PATTERN NUM
DCA PATNUM
TAD PATNUM
TAD PATTBL
DCA TEMP1 /ADDRESS TO GET
TAD I TEMP1 /ROUTINE ADDRESS
DCA TEMP1 /GENERATE PATTERN
JMS I TEMP1

CMA
DCA I LSTBLK
DCA I POSITN
JMS I NEWDRV
JMP .04
DCA DIRFLG
IAC
DCA BLKING
TAD BLKING
SPA
CLA I LSTBLK
DCA RECORD
JMS I SRCHIT
TAD K0050

WRTLPL1,

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 270I OPTIONS
4436	5230	JMP WRTLPI+7	/RETURN, NOT DONE ALL
4437	4437	JMS I NEWDRV	/RETURN, DONE ALL
4440	5221	JMP WRTLPI	
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 DIRFLG	/GO OTHER DIRECTION
4445	3114	DCA DIRFLG	
4446	1113	TAD BLKING	/MAKE BLOCK
4447	7041	CMA IAC	/INCREMENTER
4450	3113	DCA BLKING	/COMPLIMENT

/TEST READ COMPARE OPTION

4451	7604	/EXAMINE SWITCHES	
4452	0121	RDCOMP. LAS	
4453	7640	AND K0040	
4454	5344	SEA CLA	
4455	7040	JMP RSEND+13	
4456	3364	CMA RDCRAS	
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	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 DATAGO	/VERIFY DATA PATTERN
4500	7175	BUFFR2	
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 DATAGO	/VERIFY DATA READ

4512 7376
4513 4767
4514 5270

4515 6772
4516 0110
4517 7640
4520 5331
4521 1427
4522 7640
4523 5331
4524 1430
4525 1366
4526 3330
4527 4442
4530 7175
4531 4437
4532 5257
4533 1114
4534 7040
4535 3114

ROSEND, JOT 772
AND EZBIT
SEA CLA
JMP ROSEND
TAD I WCLOC
SEA CLA
JMP ROSEND
TAD I CALOC
TAD K7600
DCA I+2
JMS I DATA0
BUFFR2
ROSEND, JMS I NEWDRV
JMP RDCOMP+6
TAD DIRFLG
CMA
DCA DIRFLG

/END ZONE INTERRUPT
/FORM BUFFER ADDRESS
/OR BUFFR3
/GO OTHER DIRECTION
/MAKE BLOCK INCREMENTER
/OTHER DIRECTION
/READ BOTH DIRECTIONS
/YES
/READ OTHER DIRECTION
/WRITTEN 2701 YET
/WRITE NEXT SET
/TO TEST SWITCHES FOR WRITE
/TO TEST SWITCHES FOR READ
/READ PASS SWITCH
/P1 PASS 1 0 PASS 2

4536 1113
4537 7041
4540 3113
4541 2364
4542 7410
4543 5257
4544 7604
4545 0123
4546 7640
4547 5255
4550 1424
4551 1077
4552 7640
4553 5221
4554 1114
4555 7640
4556 5221
4557 5760
4560 5161

TAD BLKING
CMA IAC
DCA BLKING
ISZ RDCPAS
SKP
JMP RDCOMP+6
CLA OSR
AND K0100
SEA CLA
JMP RDCOMP+4
TAD I LSTBLK
TAD K5077
SEA CLA
JMP WRTLP1
TAD DIRFLG
SEA CLA
JMP WRTLP1
JMP I.+1
WRRDND

WRTSLP, WRTSWS
RDSWLP, ROSWS
TEMP1, 0
RDCPAS, 0
/ROUTINE ADDRESS FOR PATTERNS
PATTBL, PTABLE
K7600, 7600
RERFLG, REFLGS
PATNUM, 0
WRDIR

4561 5200
4562 5400
4563 0000
4564 0000
4565 4724
4566 7600
4567 4734
4570 0000
4571 5157

/PATTERN GENERATION FOR
/INITIAL WRITE TEST

4600	**4600
4600	GNPAT0, 0
4601	CLA CLL
4602	JMS GNSTRA
4603	JMP I GNPAT0
4604	GNPAT1, 0
4605	CLA CMA CLL
4606	JMS GNSTRA
4607	JMP I GNPAT1
4610	GNPAT2, 0
4611	CLA STL
4612	JMS GNSTRA
4613	JMP I GNPAT2
4614	GNPAT3, 0
4615	TAD ,+4
4616	CLL
4617	JMS GNSTRA
4620	JMP I GNPAT3
4621	2525
4622	GNPAT4, 0
4623	TAD ,+4
4624	CLL
4625	JMS GNSTRA
4626	JMP I GNPAT4
4627	5252
4630	GNPAT5, 0
4631	TAD ,+4
4632	CLL
4633	JMS GNSTRA
4634	JMP I GNPAT5
4635	0707
4636	GNPAT6, 0
4637	TAD ,+4
4640	CLL
4641	JMS GNSTRA
4642	JMP I GNPAT6
4643	7070
4644	GNPAT7, 0
4645	TAD GNPAT4=1
4646	STL
4647	JMS GNSTRA
4650	JMP I GNPAT7

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

4651	0000	GNSTRA, 0
4652	3010	DCA 10
4653	1032	TAD BFILOC
4654	3011	DCA 11
4655	1064	TAD K7977
4656	3012	DCA 12
4657	1010	TAD 10
4660	3411	DCA 1 11
4661	2012	ISZ 12
4662	7410	SKP
4663	5651	JMP 1 GNSTRA
4664	7420	SNL
4665	5297	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	PTABLE, GNPAT0
4725	4604	GNPAT1
4726	4610	GNPAT2
4727	4614	GNPAT3
4730	4622	GNPAT4
4731	4630	GNPAT5

4732 4636
4733 4644

GNPAT6
GNPAT7

/TEST READ ERRORS
/AND RESYNC NEXT BLOCK

4734 5334
4735 2742
4736 5744
4737 2743
4740 5744
4741 5734
4742 5150
4743 5526
4744 4461

REFLGS, JMP .
ISZ I ,+5
JMP I ,+6
ISZ I ,+4
JMP I ,+4
JMP I REFLGS
COFLAG
NOSERR
RDCOMP+10

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

5000
5001 7200
5002 1031
5003 3345
5004 1600
5005 3346
5006 2200
5007 7040
5010 3350
5011 1064
5012 3347

*5000
CODATA, JMP .
CLA
TAD BF1WD1 /GET KNOWN DATA
DCA KNDA /ADDRESS
TAD I CODA /UNKNOWN DATA
DCA UKDA /ADDRESS
ISZ CODATA
CMA COFLAG /SET ERR COUNT FLAG
TAD K7577 /NUMBER OF WORDS
DCA NUMWRD
/TST FOR DELETE COMPARE

5013 7604
5014 0102
5015 7640
5016 5600
5017 1745
5020 7041
5021 1746
5022 7640
5023 5231

LAS
AND K0400
SZA CLA
JMP I CODA
TAD I KNDA
CMA IAC
TAD I UKDA
SZA CLA
JMP COCOMP /WORDS =
/NO TEST COMPLIMENT

5024 2345
5025 2346
5026 2347
5027 5217
5030 5600

COINCR, ISZ KNDA
ISZ UKDA
ISZ NUMWRD /DONE ALL
JMP COLOOP
JMP I CODA /EXIT

5031 1745
5032 7450
5033 5237
5034 7040
5035 7640
5036 5251

COCOMP, TAD I KNDA
SNA
JMP ,+4 /WORD = 0/S
CMA /YES, TRY COMPLIMENT
SZA CLA /WORD=1/S
JMP COERRO /NO DATA ERROR

5037 1745 PAL10 V141
 5040 7001 TAD I KNDATA
 IAC
 5041 1746 TAD I UKDATA
 5042 7640 SZA CLA
 5043 5251 JMP COERRO
 5044 2345 ISZ KNDATA
 5045 2346 ISZ UKDATA
 5046 2347 ISZ NUHWRD
 5047 5237 JMP COCOMP+6
 5050 5600 JMP I CODATA

/MAKE 2'S COMP
 /COMPLIMENTS *
 /NO ERROR,
 /STAY IN TEST COMP

/DATA ERROR TYPEOUT

5051 2350 COERRO, ISZ COFLAG
 5052 5307 JMP COERR1
 5053 4444 JMS I ERRSTP
 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
 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
 5100 7650 SNA CLA
 5101 5306 JMP +5
 5102 4704 JMS I +2
 5103 5307 JMP COERR1
 5104 1545 TYBKW
 5105 1555 TYFWD
 5106 4705 JMS I -1

/FIRST ERROR
 /NO HDR ALREADY TYPED
 /STOP TAPE

/TYPE BLOCK

/TYPE DIRECTION
 /BLOCK WAS WRITTEN

5107 4456 COERR1, JMS I TYPTX
 5110 7777 7777
 5111 7700 7700
 5112 1345 TAD KNDATA
 5113 4451 JMS I TYPCON
 5114 4456 JMS I TYPTX
 5115 0053 53

/LINE FEED TO
 /SEPARATE ERRORS
 /TYPE GOOD DATA

5116	5657		
5117	6756		
5120	7700		
5121	1346	TAD UKDATA	
5122	4451	JMS I TYPCON	/TYPE DATA READ
5123	4456	JMS I TYPTEX	
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 TYPTEX	/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	UKDATA: 0	
5147	0000	NUMWRD: 0	
5150	0000	COFLAG: 0	
5151	5351	WAETYP: JMP	
5152	4444	JMS I ERRSIP	
5153	4756	JMS I .+3	
5154	4447	JMS I SBTYPE	
5155	5751	JMP I WAETYP	
5156	1451	TYWALL	
5157	0000	WRDIR: 0	
5157	0044	STPERRERRSTP	

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

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

```

5200 *5200
5200 WRTSWS, JMP ;
5201 JMS I WAITI /WAIT NORMAL INT
5202 I /TEST FOR END ZONE
5203 JMP WRTEZT /WC GO TO 0
5204 TAD I WCLOC /NO ERROR
5205 SZA CLA /INDICATE 1 OR 32 BLOCKS
5206 JMP WRTEZT+4 /NO DO ALL
5207 CLA OSR /BLOCK
5208 AND BLKBT /NO TEST 32 BLOCKS
5209 SNA +7 /INC EXIT ADDRESS
5210 JMP AND BLKBIT /STOP TAPE
5211 AND BLKBIT
5212 SNA CLA
5213 JMP WRT32
5214 JMS WRTSWS
5215 ISZ WRTSWS
5216 TAD K0200
5217 IOT 764
5220 TAD RECORD
5221 DCA I POSIFN
5222 TAD RECORD
5223 TAD BLKING
5224 DCA RECORD
5225 TAD DIRFLG
5226 SZA CLA /GOING FORWARD
5227 JMP I WRTSWS /BACKWARD EXIT
5230 TAD I LSTBLK /INCREMENT LAST
5231 IAC /BLOCK WRITTEN
5232 DCA I LSTBLK
5233 JMP I WRTSWS
5234

WRTSWS,
5235 IOT 772
5236 AND EZBIT
5237 SZA CLA
5240 JMP WRTEZA
5241 IOT 761
5242 AND K0200
5243 SNA CLA
5244 JMP +5 /IF SW3=1
5245 LAS
5246 AND K0400
5247 SZA CLA
5250 JMP WRTSWS+7
5251 JMS I ERRSTP
5252 JMS I WDATTY
5253 TAD RECROK
5254 JMS I TYPCON

```

/IF TAPE NOT STOPPED

5255 4456 JMS I TYPTEX
 5256 0042 42
 5257 5457
 5260 4353
 5261 7700
 5262 4447 JMS I SBTYP
 5263 1027 TAD WCLOC
 5264 4451 JMS I TYPCON
 5265 4456 JMS I TYPTEX
 5266 0067 67
 5267 1643
 5270 1677
 5271 4435 JMS I SRCHIT
 5272 1115 TAD K0050
 5273 6764 IOT 764

/TYPE WHATS LEFT OF WC

5274 1032 TAD BFLOC
 5275 3430 DCA I CALOC
 5276 1064 TAD K7577
 5277 3427 DCA I WCLOC
 5300 5201 JMP WRTSWS+1

WRTEZA, TAD RECORD
 5301 1020 TAD K5076
 5302 1126 SNA CLA
 5303 7650
 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

WRT32, TAD RECORD
 5312 1020 DCA I POSITN
 5313 3422 TAD RECORD
 5314 1020 TAD BLKING
 5315 1113 DCA RECORD
 5316 3020 TAD DIRFLG
 5317 1114 CMA IAC
 5320 7041 TAD RECORD
 5321 1020 AND K0037
 5322 0127 SEA CLA
 5323 7640 JMP WRTSWA
 5324 5226 TAD K0200
 5325 1075 IOT 764
 5326 6764 ISZ WRTSWS
 5327 2200 JMP WRTSWA
 5330 5226

/ROUTINE TO CHECK STATUS B BITS 6-8

CHKB, 0
 5331 0000 CLA CLL CMA
 5332 7340 JMS GOB
 5333 4352 JMS GOB
 5334 4352 TAD K2525
 5335 1131

/CHECK PATTERN 7777
/CHECK PATTERN 0000

5336 4352 JMS GOB
 5337 1131 TAD K2525
 5340 7040 CMA
 5341 4352 JMS GOB
 5342 2133 ISZ DTCNT
 5343 5332 JMP CHKB +1
 5344 1133 TAD DTCNT
 5345 4352 JMS GOB
 5346 2133 ISZ DTCNT
 5347 5344 JMP .+3
 5350 7300 CLA CLL
 5351 5731 JMP I CHKB

5352 0000 / GOB,
 5353 0140 AND K0070
 5354 3132 DCA DTSV
 5355 1132 TAD DTSV
 5356 6774 IOT 774
 5357 7440 SZA
 5360 7402 HLT
 5361 6772 IOT 772
 5362 0140 AND K0070
 5363 7041 CIA
 5364 1132 TAD DTSV
 5365 7440 SZA
 5366 7402 HLT
 5367 7240 CLA CMA
 5370 6772 IOT 772
 5371 7040 CMA
 5372 7440 SZA
 5373 7402 HLT
 5374 5752 JMP I GOB

5392 0000 /
 5393 0140 AND K0070
 5394 3132 DCA DTSV
 5395 1132 TAD DTSV
 5396 6774 IOT 774
 5397 7440 SZA
 5398 7402 HLT
 5399 6772 IOT 772
 5400 7040 CMA
 5401 7440 SZA
 5402 7402 HLT
 5403 5752 JMP I GOB

5400 5400 *5400
 5401 5200 RDSWS,
 5402 4440 JMP I WAITI
 5403 0001 1
 5404 5252 JMP TSRDEZ
 5405 1427 TAD I WCLOC
 5406 7640 SZA CLA
 5407 5302 JMP RDSERR
 5408 7040 CMA
 5409 3326 DCA NOSERR
 5410 7604 CLA OSR
 5411 0117 AND BLK8TS
 5412 7450 SNA
 5413 5225 JMP .+11 /NO,ALL
 5414 0120 AND BLKBIT
 5415 7650 SNA CLA /1 BLOCK

5400 5400 /WAIT FOR READ INTERRUPT AND
 5401 5200 /TEST SWITCHES FOR NUMBER OF BLOCKS
 5402 4440 /WAIT NORMAL INTERRUPT
 5403 0001 /TEST FOR END ZONE
 5404 5252
 5405 1427
 5406 7640
 5407 5302
 5408 7040
 5409 3326
 5410 7604
 5411 0117
 5412 7450
 5413 5225
 5414 0120
 5415 7650

5400 5400 /LOAD STATUS B
 5401 5200 /DID IOT 774 CLEAR AC
 5402 4440 /IOT 774 DID NOT CLEAR AC
 5403 0001 /GET STATUS B
 5404 5252 /MASK 6-9
 5405 1427 /THIS IS THE GOOD STATUS WORD
 5406 7640 /WAS STATUS B O.K.
 5407 5302 /STATUS B OR AC FAILED
 5408 7040 /GET STATUS B
 5409 3326 /DID OR WORK
 5410 7604 /STATUS B OR AC FAILED
 5411 0117 /EXIT PATTERN O.K.
 5412 7450
 5413 5225
 5414 0120
 5415 7650

5417 5234 JMP RDSW32
 5420 1075 TAD K0200
 5421 6764 IOT 764
 5422 1020 TAD RECORD
 5423 3422 DCA I POSITN
 5424 5600 JMP I RDSWS
 5425 1020 TAD RECORD
 5426 3422 DCA I POSITN
 5427 1020 TAD RECORD
 5430 1113 TAD BLKINC
 5431 3020 DCA RECORD
 5432 2200 ISZ RDSWS
 5433 5600 JMP I RDSWS

RDSW32.
 5434 1020 TAD RECORD
 5435 3422 DCA I POSITN
 5436 1020 TAD RECORD
 5437 1113 TAD BLKINC
 5440 3020 DCA RECORD
 5441 1114 TAD DIRFLG
 5442 7041 CMA IAC
 5443 1020 TAD RECORD
 5444 0127 AND K0037
 5445 7640 SZA CLA
 5446 5232 JMP RDSW32=2
 5447 1075 TAD K0200
 5450 6764 IOT 764
 5451 5600 JMP I RDSWS

/NO, 32 BLOCKS
 /1 BLOCK, STOP TAPE
 /NEW POSITION
 /NEXT BLOCK
 /+ OR = 1
 /DONE AN INC OF 32
 /32 BLOCKS, STOP TAPE

5452 6772 TSRDEZ. IOT 772
 5453 0110 AND EZBIT
 5454 7650 SNA CLA
 5455 5302 JMP RDSERR
 5456 1020 TAD RECORD
 5457 1126 TAD K5076
 5460 7650 SNA CLA
 5461 5265 JMP .+4
 5462 1020 TAD RECORD
 5463 7700 SMA CLA
 5464 5302 JMP RDSERR
 5465 5600 JMP I RDSWS

PARRSE.
 5466 3266 JMP
 5467 4444 JMS I ERRSIP
 5470 4446 JMS I RDATAIY
 5471 1026 TAD RECROK
 5472 4451 JMS I TYPCON
 5473 4456 JMS I TYPTEX
 5474 0042 42
 5475 5457 5457
 5476 4353 4353
 5477 7700 7700
 5500 JMS I SBTYPE
 5501 JMP I PARRSE

/END ZONE INT
 /NO, ERROR
 /BLOCK 2702
 /2702 DOESN'T EXIST
 /BLOCK=1
 /NO, EZ STAT IN ERROR
 /EXIT
 /STOP TAPE
 /READ AND DIRECTION
 /TYPE BLOCK

```

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 WCL0C      /WC GO TO 0
5515 7650      SNA CLA      /YES
5516 5211      JMP RDSWS+11
5517 1027      TAD WCL0C
5520 4451      JMS I TYP0CN
5521 4456      JMS I TYPTEX
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
1604 WRSCOP, LAS
1601 0073 AND K0007
1602 1371 TAD PATTBA
1603 3370 DCA TEMPY
1604 1770 TAD I TEMPY
1605 3370 DCA TEMPY
1606 4770 JMS I TEMPY
1607 1103 TAD K0214
1610 1060 TAD UNFUNC
1611 6766 IOT 766
1612 7440 SZA
1613 7402 HLT
1614 1025 TAD IDCON
1615 3430 DCA I CALOC
1616 6773 IOT 773
1617 5216 JMP .+1
1620 7710 SPA CLA
1621 5255 JMP WRTSEZ
1622 1115 TAD K0050
1623 6764 IOT 764
1624 7440 SZA
1625 7402 HLT
1626 1032 TAD BF1LOC
1627 3430 DCA I CALOC
1630 1064 TAD K7577

/PAT NUM + TABLE ADRS

/GENERATE PATTERN
/SEARCH +
/DRIVE

/IOT 766 OR 764 DID NOT CLEAR AC

/WAIT FOR FLAG

/ERROR STATUS

/CHANGE TO WRITE

/IOT 764 DID NOT CLEAR AC

```

```

1631 3427 DCA I WCLOC
1632 6761 IOT 761
1633 0075 AND K0200
1634 7650 SNA CLA
1635 5243 JMP WRSC01
1636 1427 TAD I WCLOC
1637 7650 SNA CLA
1640 5243 JMP .+3
1641 6771 IOT 771
1642 5232 JMP .+10
1643 1375 TAD K7730
1644 3370 DCA TEMPY
1645 6771 IOT 771
1646 7410 SKP
1647 5252 JMP .+3
1650 2370 ISE TEMPY
1651 5245 JMP .+4
1652 1376 TAD K0052
1653 6764 IOT 764
1654 5212 JMP WRSCOP+12
1655 6772 WRTSEZ, IOT 772
1656 0374 AND K1000
1657 7650 SNA CLA
1660 5265 JMP .+5
1661 6761 IOT 761
1662 7040 CHA
1663 0102 AND K0400
1664 5207 JMP WRSCOP+7
1665 6761 IOT 761
1666 5263 JMP .+3

```

/WAIT FOR FLAG

/LEAVE ERROR FLAGS SET

/END ZONE
 /NO START SEARCH AGAIN
 /END ZONE SET
 /CHANGE DIRECTION

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

```

1667 1372 RDSCOP, TAD K0220
1670 1060 TAD UNFUNG
1671 6766 IOT 766
1672 7440 SEA
1673 7402 HLT
1674 1033 TAD BF2LOC
1675 3430 DCA I CALOC
1676 1064 TAD K7577
1677 3427 DCA I WCLOC
1700 0761 IOT 761
1701 0075 AND K0200
1702 7650 SNA CLA
1703 5311 JMP .+6
1704 1427 TAD I WCLOC
1705 7650 SNA CLA
1706 5311 JMP .+3
1707 6771 IOT 771
1710 5304 JMP .+4
1711 1375 TAD K7730

```

/READ DATA
 /+ DRIVE

/IOT 764 DID NOT CLEAR AC

/MONITOR MOTION
 /BIT IN CASE IT=0

/MONITOR WORD K

/AND FLAGS

```

1712 3370 DCA TEMPY
1713 6773 IOT 773
1714 7410 SKP
1715 5320 JMP .+3
1716 2370 ISE TEMPY
1717 5313 JMP .+4
1720 0374 AND K1000
1721 7640 SEA CLA
1722 5330 JMP .+6
1723 6761 IOT 761
1724 7040 CMA
1725 0075 AND K0200
1726 6764 IOT 764
1727 5272 JMP RDSCOP+3
1730 6761 IOT 761
1731 7040 CMA
1732 0102 AND K0400
1733 5267 JMP RDSCOP
RDSC01,

```

```

/MONITOR FLAG
/FOR 230 MICRO SEC:

```

```

/END ZONE SET
/YES, REVERSE

```

```

/IN CASE GO=0
/RESET ENABLES

```

```

/CHANGE DIRECTION

```

```

/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 SEA
1740 7402 HLT
1741 1377 TAD K3500
1742 3010 DCA 10
1743 1031 TAD BF1ND1
1744 3430 DCA I CALOC
1745 1431 TAD I BF1ND1
1746 2010 ISE 10
1747 5346 JMP .-1
1750 7200 CLA
1751 6772 IOT 772
1752 0374 AND K1000
1753 7640 SEA CLA
1754 5362 JMP .+6
1755 6761 IOT 761
1756 7040 CMA
1757 0075 AND K0200
1760 6764 IOT 764
1761 5337 JMP SRSCOP+3
1762 6761 IOT 761
1763 0075 AND K0200
1764 7640 SEA CLA
1765 7402 HLT
1766 1373 TAD K0600A
1767 5360 JMP .-7

```

```

/DISPLAY BLOCK
/WAIT 1 BLOCK
/APPROX

```

```

/READ B
/END ZONE
/YES REVERSE
/IN CASE GO=0

```

```

/EEZ DID NOT ZERO MOTION
/COMPLEMENT DIRECTION

```

```

TEMPY, 0
PATTBA, PTABLE

```

```

1770 0000
1771 4724

```

1772 0220 K0220, 220
1773 0600 K0600A, 600
1774 1000 K1000, 1000
1775 7730 K7730, 7730
1776 0052 K0052, 52
1777 3500 K3500, 3500

PAUSE

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

5600 PARTST. TAD K2701 /FIND 2701
5601 DCA RECORD
5602 CMA /BACKWARDS
5603 DCA DIRFLG
5604 JMS I SRCHIT /SEARCH TO READ DATA
5605 TAD K0030
5606 IOT 764
5607 TAD BF1LOC

5610 DCA I CALOC /DUMMY INPUT
5611 TAD K7600B
5612 DCA I WCLOC
5613 TAD I WCLOC
5614 SNA CLA
5615 JMP .+4 /WAIT FOR WCT0=-1
5616 IOT 771 /FLAG SET
5617 JMP .+4 /COULD BE END ZONE
5620 JMP PAREZ1
5621 IAC
5622 CMA
5623 DCA I WCLOC /2 WORDS

5624 TAD BF1LOC /TO WRITE ALL CONTINUOUS
5625 DCA I CALOC
5626 TAD K0170
5627 IOT 764
5630 CMA
5631 DCA I BF1WD1
5632 JMS I WAITI
5633 1
5634 JMP .+3 /BACK TO READ DATA
5635 TAD K0170
5636 JMP PARTST+6
5637 JMS I .+2
5640 JMP PARTST
5641 WAETYP

5641 5151

5642 6772 PAREZ1, IOT 772 /END ZONE INT;
 5643 0110 AND EZBIT
 5644 7640 SEA 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 PARHL1, TAD PARLOC
 5654 3010 DCA I0
 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 /WRITE DATA
 5662 6764 IOT 764
 5663 1410 TAD I I0
 5664 3431 DCA I BFINDI
 5665 1032 TAD BFILOC

5666 3430 DCA I CALOC /ONLY WRITE 1 WD
 5667 7040 CMA /REST OF BLOCK SHOULD
 5670 3427 DCA I WCLOC /GO TO ZERO'S
 5671 4440 JMS I WAITI
 5672 0001 1
 5673 5344 JMP PARNER
 5674 1431 TAD I BFINDI
 5675 7040 CMA
 5676 7650 SNA CLA
 5677 5303 JMP .+4 /DO ALL ONES YET?
 5700 2020 ISZ RECORD /YES
 5701 2422 ISZ I POSITN /COUNT BLOCKS
 5702 5262 JMP PARHL1*7

/INCREMENTING PARITY PATTERNS
 /0100 TO 7700 AND 0101 TO 7777
 PARWL2, DCA I BFINDI

5703 3431
 5704 6764
 5705 7040 CMA
 5706 3427 DCA I WCLOC
 5707 1032 TAD BFILOC
 5710 3430 DCA I CALOC
 5711 1431 TAD I BFINDI
 5712 1123 TAD K0100 /UPPER +1
 5713 3431 DCA I BFINDI
 5714 1431 TAD I BFINDI
 5715 7450 SNA /UPPER GONE TO 0
 5716 5323 JMP .+5 /YES, SET WORD TO 0101
 5717 0071 AND K0077
 5720 7640 SEA CLA /INCREMENTING LWR

```

/TC01 BASIC EXERCISER TAPE 1 PAL10 V141
5721 2431 ISZ I BF1W01 /ADD 1 TO LOWER
5722 5325 JMP .+3
5723 1124 TAD K0101 /DONE UPPER TO 7700
5724 3431 DCA I BF1W01 /START BOTH EQUAL
5725 4440 JMS I WAIT1
5726 2001 1 /SHOULD GET NO ERROR STATUS
5727 5344 JMP PARWER
5730 2422 ISZ I POSITN
5731 2020 ISZ RECORD
5732 1431 TAD I BF1W01
5733 7040 CMA CLA
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
5743 6200 PARTS1 /FOR CORRECT CKSUMS
                    /READ AND CHECK

5744 4444 PARWER, JMS I ERRSTP
5745 4450 JMS I WDATTY
5746 1026 TAD RECRDK
5747 4451 JMS I TYPCON
5750 2042 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

6000 /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 BRWD
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
6015 5221 JMP .+4
        /WAIT FOR LAST
        /WORD IN

6016 6771 IOT 771
6017 5213 JMP .+4
6020 5356 JMP PARRE3
        /NO FLAGS FOR READ DATA
        /WRITE CHECKSUM TO FIRST WORD
6021 1032 TAD BF1LOC
6022 3430 DCA I CALOC
6023 7001 TAC
6024 7040 CMA
6025 3427 DCA I WCLOC
6026 1122 TAD K0170
6027 6764 IOT 764
6030 1431 TAD I BF1WD1
6031 7040 CMA
6032 7650 SNA CLA
6033 3431 DCA I BF1WD1
6034 4440 JMS I WAITI
        /WRITE ALL
        /WRITE ALL CONTINUOUS
6035 0001 1 JMP PRWAE
        /WRITE ALL STATUS ERR
6036 5311 JMS PR4INC
6037 4361 JMP PARTS4+5
6040 5205

6041 1075 TAD K0200
6042 6764 IOT 764
6043 4437 JMS I NEWDRV
6044 5200 JMP PARTS4

```

```

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

6054 1064 TAD K7577
6055 3427 DCA I WCLOC
6056 4440 JMS I WAITI
        /EXPECT PARITY
        /ERROR
6057 4201 4201
6060 5277 JMP PRT5E3
6061 6772 IOT 772
6062 0075 AND K0200
6063 7650 SNA CLA
        /PARITY ERROR SET
        /NO

6064 5314 JMP PARRE4
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

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

6077 1114 PRT5E2, 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 SEA 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 PARRE3, 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

6200 3020 PARTS1, DCA RECORD
6201 3114 DCA DIRFLG
6202 4435 JMS I SRCHIT
6203 7040 CMA
6204 3430 DCA I CALOC
6205 1074 TAD K0020
6206 6764 IOT 764
6207 1130 TAD K0010

6210 6764 IOT 764
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
6217 5223 JMP ,+4

6220 6771 IOT 771
6221 5215 JMP ,+4
6222 5342 JMP PARRE1
6223 7040 CMA
6224 3427 DCA I WCLOC

/READ THE GENERATED CKSUMS BACK
/AND VERIFY THAT THEY ARE CORRECT
*6200
/FIND BLOCK 0
/CA # 7777 IN CASE
/R ALL BREAKS BEFORE DATA

/READ ALL TO RD DATA

/WAIT FOR WC TO =0

/IN CASE READ ERROR

6225 1130 TAD K0010
6226 6764 IOT 764
6227 4440 JMS I WAITI
6230 0001 1
6231 5342 JMP PARRE1
6232 1431 TAD I BF1WD1
6233 0071 AND K0077
6234 7640 SZA CLA
6235 5240 JMP :+3
6236 1431 TAD I BF1WD1
6237 0070 AND K7700
6240 1756 TAD I CKSLOC
6241 1123 TAD K0100
6242 7640 SZA CLA
6243 5310 JMP CKSERR
6244 4755 JMS I PR2INC
6245 5207 JMP PARTS1+7
6246 1075 TAD K0200
6247 6764 IOT 764
6250 4437 JMS I NEWDRV
6251 5200 JMP PARTS1

/RD DATA TO RD ALL
/WAIT FOR CKSUM IN
/TEST FOR 2 CHAR
/IN FIRST WORD
/CKSUM SHOULD =7700
/CKSUM CORRECT
/NO
/NO
/STOP TAPE
/DO NEXT DRIVE

6252 1422 /READ BLOCKS BKWD FOR NO PARITY
6253 3020 /ERRORS
6254 7040 PARTS2.
6255 3114 TAD I POSITN
6256 4435 DCA RECORD
6257 1116 CMA
6260 6764 DCA DIRFLG
6261 1032 JMS I SRCHIT
6262 3430 TAD K0030
6263 1064 IOT 764
6264 3427 TAD BF1LOC
6265 4440 DCA I CALOC
6266 0001 TAD K7577
6267 5273 DCA I WCLOC
6270 4755 JMS I WAITI
6271 5260 1
6272 5302 JMP PARREZ
6273 1020 JMS I PR2INC
6274 7700 JMP PARTS2+6
6275 5345 JMP PARTS3+2
6276 4437 TAD RECORD
6277 5252 SMA CLA
/NO, ERROR STATUS
/DONE ALL DRIVES
/DO NEXT

/FIND LAST BLOCK BKWDS
/TEST FOR END ZONE
/DONE BLOCK 0
/NO, ERROR STATUS
/DONE ALL DRIVES
/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	6744	IOI 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 TYPTEX	
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 TYPTEX	
6325	0044	44	
6326	4164	4164	
6327	4177	4177	
6330	1356	TAD CKSLOC	
6331	4451	JMS I TYPCON	
6332	4456	JMS I TYPTEX	
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	PARRE	
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	1075	FORDTY	1076	K2701	0100	PARRSE	5466
BF1LOC	0032	FWDTYP	4111	K3500	1777	PARTAB	4672
BF1WD1	0031	GBKWL	3214	K4000	0066	PARTS1	6200
BF2LOC	0033	GET4IN	2172	K4215	0296	PARTS2	6252
BF3LOC	0034	GETMIN	2310	K5000	0142	PARTS3	6300
BKWTYP	4110	GNPAT0	4600	K5076	0126	PARTS4	6000
BLKBIT	0120	GNPAT1	4604	K5077	0077	PARTS5	6045
BLKBT8	0117	GNPAT2	4610	K7000	2345	PARTS6	6073
BLKFLG	0750	GNPAT3	4614	K7472	2673	PARTST	5600
BLKFND	0021	GNPAT4	4622	K7475	3131	PARWER	5744
BLKINC	0113	GNPAT5	4630	K7574	0134	PARWL1	5653
BLOCKK	3770	GNPAT6	4636	K7577	0064	PARWL2	5703
BNOTCN	3722	GNPAT7	4644	K7600	4566	PATCHK	1122
BUFFR2	7175	GNPTRS	0462	K7600A	6175	PATNUM	4570
BUFFR3	7376	GNSTRA	4651	K7600B	5757	PATT9A	1771
BUFFRS	6774	GOB	5392	K7700	0070	PATTRL	4545
CALOC	0030	HLTNS	0311	K7730	1775	PNTRS	0501
CDRIVE	0057	IDCON	0025	K7760	0063	POSITN	0022
CHKB	5331	IOT	6000	K7767	0067	POSSAV	0111
CHKGO	2744	IRECD	0254	K7772	4374	PR2INC	6355
CHNGOR	0440	K0003	0076	KM25	0325	PR4INC	6161
CIPHER	0211	K0006	4373	KNDATA	5145	PREBLK	0751
CKSERR	6310	K0007	0073	LSTBLK	0024	PRTSEZ	6077
CXSLOC	6356	K0010	0130	MSBITS	0061	PRWAF	6111
COCOMP	5031	K0020	0074	MVBKWD	2223	PT4BLE	4724
CODATA	5000	K0030	0116	MVCHNG	2273	RDATTY	0046
COERR1	5107	K0037	0127	MVEND	2305	RDEND	4515
COERRO	5051	K0040	0121	MVEGUL	2047	RDCLP1	4470
COFLAG	5150	K0050	0115	MVFWO	2216	RDGOMP	4451
COINCR	5024	K0052	1776	MVG9KW	2121	RDCPAS	4564
COLOOP	5017	K0070	0140	MVGCHG	2135	RDSO01	1723
COMBIT	0062	K0077	0071	MVGFWO	2113	RDSOOP	1667
CRLFLF	1242	K0100	0123	MVGRPT	2156	RDSEND	4531
DAPSAV	4100	K0101	0124	MVGSTP	2127	RDSERR	5502
DATACO	0042	K0140	6155	MVGWAT	2147	RDSW32	5434
DIRECT	0023	K0170	0122	MVREST	2020	RDSWLP	4562
DIRFLG	0114	K0200	0075	MVRPT	2257	RDSWS	5400
DIRSAV	0112	K0204	0125	MVRTBL	2073	RECORD	0020
DOLOOP	2200	K0212	1260	MVSTOP	2230	RECRDK	0026
DOTHEM	2112	K0214	0103	MVTEST	2000	REFLGS	4734
DRIVTY	0045	K0215	1257	MVWAIT	2246	REPOSI	0400
DRVTYP	2743	K0220	1772	NEWDRV	0037	REERFLG	4567
DTCHK	1077	K0240	0072	NOSERR	5526	RESBEZ	4336
DTCNT	0133	K0400	0102	NUMWRD	5147	RESBK9	4320
DTSAV	0132	K0600	2344	OTHRTX	2721	RESBKW	4276
ERRSTP	0044	K0600A	1773	PARZ31	5642	RESFWD	4260
ERSSTA	0053	K0604	0065	PARLOC	5756	RESFWD	4213
ERSTP	1310	K0614	0101	PARRE1	6342	RESFWF	4232
EZBIT	0110	K1000	1774	PARRE2	6345	RESXIT	4250
EZERR	3366	K2525	0131	PARRE3	6156	RESYEZ	4354
EZERRA	3436	K260	1535	PARRE4	6114	RESYNC	4200
EZTYPE	4067	K2700	3540	PARREZ	6273	REWIND	0036

TC01 BASIC EXERCISER TAPE 1	PAL17	V141	31-MAY-72	1>J1	PAGE 1-66
RSFOR1 0436	SSTBBE	3711	WRSC01	WRSC01	1643
RSFORV 0420	SSTBKW	0055	WRSCOP	WRSCOP	1600
SAVPAD 0043	SSTBR	3670	WRT32	WRT32	5312
SBABK 0106	SSTBSE	3703	WRTEZA	WRTEZA	5301
SBAFK 0107	SSTER1	3210	WRTEZT	WRTEZT	5235
SBCONS 0675	SSTEZ1	3501	WRTLP1	WRTLP1	4421
SBRECV 1331	SSTEZA	3470	WRTSEZ	WRTSEZ	1655
SBTYP E 0047	SSTFBE	3640	WRTSLP	WRTSLP	4561
SCH1ER 2433	SSTFR	3617	WRTSHA	WRTSHA	5226
SCH1ND 2462	SSTFSE	3632	WRTSWS	WRTSWS	5200
SCH1OK 2444	SSTFWD	0054	WRTTST	WRTTST	4400
SCH1ST 2421	SSTRNA	3200	WT500	WT500	0313
SCH2ER 2514	STPERR	0044	WTHALF	WTHALF	0041
SCH2OK 2526	SYNCRE	4107	WTJUMP	WTJUMP	0272
SEARCH 0600	TAPONT	0752	WTINT	WTINT	0234
SEK2A 1071	TEMP1	4563	XCHK	XCHK	0137
SEK2A 3000	TEMPY	1770	XCHKGO	XCHKGO	0136
SER2A1 2720	TSRDEZ	5492	XDTCHK	XDTCHK	0135
SER2LS 3111	TSTATB	1316	Z7600	Z7600	0141
SER2NZ 2674	TSTTBL	0277			
SER2ST 2637	TXSTOR	1307			
SER2TY 3123	TYALL	1503			
SERCH2 2600	TYBKW	1545			
SERCMP 3070	TYCHAR	1230			
SEREZ 2655	TYCONT	1261			
SFABK 0105	TYCOVR	1273			
SFAFK 0104	TYDATA	1475			
SFIERRO 2566	TYDIR	1536			
SRCHI 2400	TYDRV	1511			
SRCH2 2501	TYFWD	1555			
SRCHER 0747	TYINTX	2733			
SRCHIT 0035	TYMODE	1563			
SRCHTY 0052	TYMOVE	1400			
SRCNCK 0656	TYPCON	0051			
SREZTS 0722	TYPTX	0056			
SRHERR 1000	TYRALL	1433			
SRSCOP 1734	TYRDAT	1424			
SRTAFN 0635	TYREAD	1460			
SSTA1A 3267	TYSRCH	1411			
SSTA2A 3311	TYTEXT	1200			
SSTA3A 3334	TYWALL	1451			
SSTA3B 3353	TYWDAT	1442			
SSTA4A 3412	TYWRIT	1466			
SSTA4B 3424	UKDATA	5146			
SSTABW 3651	UNFUNC	0060			
SSTAER 4000	WAETYP	5151			
SSTAEZ 3437	WAIITI	0040			
SSTAFW 3600	WAIIN	2040			
SSTAN1 3254	WALOC	0027			
SSTAN2 3277	WDATTY	0050			
SSTAN3 3322	WRDIR	5157			
SSTAN4 3400	WRROND	5161			

ERRORS DETECTED: 0

LINKS GENERATED: 0

RUN-TIME: 23 SECONDS

3K CORE USED