PDP-15 A9
DECTAPE

IDENTIFICATION

DEC-TAPE

PRODUCT CODE:	MAINDEC-15-D3BB-D(D)
PRODUCT NAME:	TC02 BASIC EXERCISER
DATE CREATED:	FEBRUARY 5, 1971
MAINTAINER:	DIAGNOSTIC GROUP

(11)

1. ABSTRACT

The TC02 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 TC02 DECtape control and any configuration of 1 to 8 TU55 DECtape transports.

The Basic Exerciser consists of several basic routines that may be individually selected. Each routine, with the exception of the instruction test, will operate on either any drive or any configuration of 1 to 8 drives.

The Scope Loops will operate on any drive, but only use the lowest drive selected or Drive 8 if it is selected.

These routines include:

Basic Motion or Move Scope Loop

Search Scope Loop

Read Data Scope Loop

Write Data Scope Loop

Search Find All Blocks Test

Basic Write/Read Data Test

Parity Generation and Checking Test

Search Routine 2 (Read 256 Block Numbers and Compare or Print)

Start/Stop/Turnaround Test

2. REQUIREMENTS

2.1 Equipment

PDP-15 (Standard)

TC02 DECtape Control

1 to 8 TU55 DECtape Transports

API (Optional)

2.2 Storage

Basic Exerciser - Part 1 occupies most of memory from address 0000 to 4700 and utilizes three buffer areas as follows:

Address	Function
06400 to 06777	Output Buffer Area
07000 to 07377	Input Buffer 1
07400 to 0777	Input Buffer 2

Note: Buffer addresses are relative to the extend memory bank specified for data break.

2.3 Preliminary Programs

None

3. LOADING PROCEDURE

Place the ABS Binary tape in the Paper Tape Reader (high speed, if available).

Set the ADDRESS switches to 17700.

Set BANK MODE switch to 1.

Depress I/O RESET

Press READ IN

4. STARTING PROCEDURE

4.1 Control Switch Settings

Any configuration of 1 to 8 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.

Switch	Drive
0	8
1	1
2	2

4.1 Control Switch Settings

Switch	Drive
3	3
4	4
5	5
6	6
7	7

AC switch 12 = 0 indicates do not utilize API
AC switch 12 = 1 indicates utilize API

The particular routine to be used is selected by placing the number of the routine in SWITCH REGISTER bits 14 to 17.

SWS 14 to 17	Routine	Read Paragraph
0	Move scope Loop	7.1
1	Search Scope Loop	7.2
2	Read Data Scope Loop	7.3
3	Write Data Scope Loop	7.4
4	Search Find All Blocks	7.5
5	Write/Read Data Test	7.6
6	Parity Generation Test	7.7
7	Basic Search Routine 2	7.8
10	Start/Stop/Turnaround Test	7.9

4.2 Starting Address

The Basic Exerciser starts at address 200.

4.3 Program and/or Operator Action

a. Set the ADDRESS SWITCHES to 0200.

b. Set the SWITCH REGISTER to:

(1) Select drives. (SW0 to SW7 = is invalid)
Master bit selection in switches 0 to 7 per paragraph 4.1.

(2) Select Test Routine to be run. Place the test number to be run in switches 14 through 17 per paragraph 4.1. (Also, read the test routine description if you haven't already done so.)

(3) Select API option.
Switch 12=0 NO API Switch 12=1 API

c. Press I/O RESET.

d. Press START.

e. The processor halts at address 240.

f. Reset the SWITCH REGISTER ro 000000 or as desired for the particular test and per paragraphs 5.1, 5.1.1, and 5.1.2.

g. Press CONTINUE.

5.. OPERATING PROCEDURE

5.1 Operational Switch Setting (Standard)

SW0 = 1 is delete all typeouts

SW1 = 1 is delete error halts

SW2 = 1 is type only first 4 Data Compare errors

SW3,4, and 5 indicate the extended memory bank to use for data breaks in routines 1, 2, 3, and 5.

Read each of the individual test descriptions to determine if any other switch settings apply to that particular routine.

5.1.1 Test Routines and Applicable RUN Switches

Test Routine	Switches	Usage
0 MOVE SCOPE LOOP	SW0 to SW17 = 0	Change direction at end zones
	SW0 to SW17 ≠ 0	Move tape in current direction either until end zone or for approximately 18 ms for each switch increment - change direction at end zones or when switches = increment count.
1 SEARCH SCOPE LOOP	Switches 3 and 4	Select memory bank for search data break.
	Switches 5 to 17 = 0	Turn around at end zones.
	Switches 5 to 17 ≠ 0	Move tape in current direction until end zone or for 36 ms for each increment of switches turn around at end zone or when switches = increment count.
2 READ SCOPE LOOP	Switches 3 and 4	Specify memory bank for READ DATA data breaks.
3 WRITE SCOPE LOOP	Switches 3 and 4	Specify memory bank for WRITE DATA data breaks.
	Switches 15 to 17	Select data pattern 0 to 7. (See paragraph 5.1.2.)
4 SEARCH FIND ALL BLOCKS	Switch 0 = 1	Delete all typeouts.
	Switch 1 = 1	Delete error halts.
5 WRITE/READ DATA TEST	Switch 0 = 1	Delete all typeouts.
	Switch 1 = 1	Delete error halts.
	Switch 2 = 1	Only type first 4 data errors.
	Switches 3 and 4	Select memory bank for data breaks.

5.1.1 Test Routines and Applicable RUN Switches Cont.

Test Routine	Switches	Usage
5 WRITE/READ DATA TEST (Cont.)	Switch 8 = 1	Halt at end of WRITE pass or at end of READ pass. (Could be used for Data Compatability - Rotate reels of tape at HALT - Press CONTINUE).
	Switch 9	Ignore PARITY, DATA and WC ERRORS. (To be used as an aid to scoping.)
	Switch 10 = 0	Take next pattern to be exercised from SWS 15, 16, 17.
	Switch 10 = 1	Exercise patterns sequentially (after last block on tape has been written backwards).
	Switch 11 = 1	Read Data only. (SW11 overrides SW12) (Can also be used with switch 8 for Data Compatiblity to make sure each reel is readable on all drives.)
	Switch 12 = 1	Write Data only. (Overridden by SW11.)
	Switch 13 = 1	Exercise WRITE/READ SEQUENCE 1 block at a time. (If DECtape will run error-free in the 1 BLOCK MODE but not 16 BLOCKS or length of tape MODES the problem would appear to be in synchronizing between blocks in write or read data). (SW13 overrides SW14.)
	Switch 14 = 1	Exercise WRITE/READ SEQUENCE 16 blocks at a time. (Overridden by SW13 = 1.)

5.1.1 Test Routines and Applicable RUN Switches Cont.

Test Routine	Switches	Usage
	Switches 13 & 14 = 0	Exercise WRITE/READ SEQUENCE length of tape.
	Switches 15, 16, & 17	Select DATA PATTERNS 0 to 7. (See paragraph 5.1.2.)
6 PARITY TEST	Switch 0 = 1	Delete all typeouts.
	Switch 1 = 1	Delete error halts.
	Switch 8 = 1	Halt at end of each step (1 to 6) in the Parity Test. a. Could be used to prepare tapes so that all reverse checksums are in a known state (00) for some other test routine. (Start parity test with SW8 = 1 when HALT is executed all reverse checksums = 00.) b. Could be used in conjunction with SW12 for a drive compatibility test.
	Switch 12 = 1	Repeat Current Parity Test Pass.
7 BASIC SEARCH	NONE	Switch 0 will still delete all typeouts, but the only reason to run this routine is to get the typeouts.
10 START STOP TURNAROUND	Switch 0 = 1	Delete all typeouts.
	Switch 1 = 1	Delete error halts.
	Switch 12 = 1	Repeat START FORWARD/STOP twice. (Only exercise every 6th block the length of tape.)

5.1.2 Pattern Selection Switches 15, 16, and 17

Data patterns for the Write Data Scope Loop and Write/Read Data Tests are selectable via AC Switches 15, 16, and 17. The patterns that may be selected are as follows:

Pattern	Description
0	All zeros pattern. (All ones when read in direction opposite written.)
1	All ones pattern. (All zeros when read in direction opposite written.)
2	A word of zeros followed by a word of ones followed by a word of zeros.
3	Words of 252525.
4	Words of 525252.
5	Words of 070707.
6	Words of 707070.
7	Words of 252525 alternate with 525252. The first word out is 252525 and the last word is 525252. When read in the direction opposite written, the first word is 525252.

5.2 Subroutine Aspects

5.2.1 Search Subroutine

The search subroutine is a common subroutine used by several of the test routines and operates in the following manner:

a. Symbolic location RECORD (address 0641) contains the block number that the test routine wants the search subroutine to find.

b. Symbolic location DIRFLG (address 0640) indicates the direction that the block should be found in. If the contents of DIRFLG is all zeros, the block should be found in the forward direction. If the contents of DIRFLG is all ones the block will be found in the backward direction.

5.2.1 Search Subroutine Cont.

c. The search subroutine always starts its initial search in the direction opposite the direction indicated by DIRFLG (i.e., DIRFLG = 0 for forward, start search backward). Therefore, this search subroutine will always make one turn around before finding the block.

d. Turnaround is not made until the tape is at least two block numbers past the block wanted (i.e., block wanted forward is 0005 - turnaround is made at block 0003 going backwards).

5.3 Program and/or Operator Action

This series of routines is designed for initial checkout of a TC02 DECtape Control and its associated drives, or maintenance and repair of the control and drives after installation.

The following procedure could be used for initial checkout 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 TC02 Instruction Test (in Part 2). It is used to verify control operations (see Part 2 for operating instructions).

Upon completing the TC02 Instruction Test (Part 2), read in Part 1 (this part) and proceed per procedure given below:

5.3.1.1 Tape Motion and Timing Pulse Generation

Put one drive on line and start the Basic Motion Routine with that drive SWITCH REGISTER selected.

Set the ADDRESS switches to 000200.

Put a DECtape on the drive to be run with the tape positioned close to the front of tape (i.e., with approximately ten feet of tape on the right-hand reel).

Set the AC switch corresponding to the drive selected to 1, and all other switches to 0.

Press I/O RESET.

5.3.1.1 Tape Motion and Timing Pulse Generation Cont.

Press START.

The program will record in memory the drive and routine selection and HALT at address 000240.

Set all AC switches to 0.

Press CONTINUE.

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 C2 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 press STOP.

Forward tape motion on the selected drive should be set and the right-hand reel should be free with a small amount of torque holding the tape tight.

Now, set the AC switches to 0001000_8.

Press CONTINUE.

The tape on the selected drive should start moving forward again and after a short period of time, 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.

Now press STOP again.

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.

5.3.1.2 New U + M Delay

The new unit and motion delay can be generated by any one 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 TC02 logic diagrams. An example of a program that could be used is:

5.3.1.2 New U + M Delay Cont.

Manually move the tape until approximately an even amount of tape is on both reels.

Start the Basic Motion Routine (Test Routine 0) as described in Paragraph 5.3.1.1.

Instead of setting all AC switches to 0 at the first HALT, set the switches to 00020_8.

Press CONTINUE.

The tape will rock from FWD, BKWD, FWC, etc.

Every time tape direction changes, the NEW U + M Delay should fire.

5.3.1.3 END ZONE Detection

The tape motion portion of the TC02 instruction test (Part 2) verifies that END ZONE is detected and that END generates ERR STOP and clears MOTION (status A, bit 4) and the BMR 1 (Buffered Motion Bit 1).

The program can be used to determine if the end zone is being detected by starting the program and watching whether or not the tape runs off the reel. Also watch the end bit in the error status. If the END indicator lights and the tape does not stop, error stop in the control is not being generated and possibly END status does not read to the AC on a DTRB. 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 the Basic Motion program to scope the END ZONE detection by setting the AC switches to 00020 to rock tape in the END ZONE area.

5.3.2 Search Operations

For the next sequence of operations, any of the three search routines could be used; but the Search Scope Loop (Test Routine 1) 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 C2 should appear to be incrementing, indicating timing pulses are being generated.

5.3.2 Search Operations Cont.

US (up to speed) should indicate 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 status 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 TC02 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 (Test Routine 4) be used to prove that the control will correctly read block numbers. The Basic Search Routine (Test Routine 7) may be used to gain more information if Routine 1 will 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 (Test Routine 2) and the TC02 timing diagrams. Also, the TC02 instruction test (part 2) checks the Read Data Timing, that data breaks occur at the right times and the DECtape flag sets at State check going to 0.

5.3.5 Check Write Data Timing

Next, the Write Scope Loop (Test Routine 3) 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 Routines 4 or 7 to verify this.) Also, the TC02 instruction test times Write Data data breaks to insure that they occur within the data area of tape and that the DECtape flag sets at the point State Check goes to 0.

5.3.6 Visual Check of Data Buffer, RWB, and LPB Utilizing the Write Scope Loop

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 and 12, 13, and 14 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 and 12, 13, and 14 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 discernible. (The LPB only complements on 0s and will contain the complement of the reverse checksum of the block it is passing over.)

5.3.6 Visual Check of Data Buffer, RWB, and LPB Utilizing the Write Scope Loop Cont.

Pattern 2 (alternate words of 0s and 1s)	All bits in the Data Buffer, RWB and LPB should glow dimly.
Pattern 3 (252525)	The even numbered bits of the buffers should act as a pattern 0 and the odd numbered bits as pattern 1.
Pattern 4 (525252)	The even numbered bits of the buffers should act as pattern 1 and the odd numbered bits as pattern 0.
Pattern 5 (070707)	The right-most three bits (of each six bits) should appear as pattern 1 and the left-most as pattern 0.
Pattern 6 (707070)	The left-most three bits (of each six bits) should appear as pattern 1 and the right-most as pattern 0.
Pattern 7 (252525 alternate with 525252)	Should appear as pattern 2. No steady states discernible in the buffers.

5.3.7 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, 8, and 12, 13, and 14 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.

5.3.8 Check Correct Data

Run the Write/Read Data Test (Test Routine 5) to verify that data is correctly read and written. Utilize the different switch configurations for a complete test or scope loop that 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.

5.3.9 Check Checksum Generation

The Parity Generation Test (Test Routine 6) verifies that checksums are being generated properly and that partiy errors will be detected if they occur.

5.3.10 Check Turnaround Function

Run the Start/Stop/Turnaround Test (Test Routine 10). 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.

5.3.11 Check API Option

Run test routine "API Test" (Part 2) to verify that the API works in a static condition. Then read in this program (Part 1) and run routines 4, 5, 6, and 7 with the API option selected (SW12 = 1 at start from 00200).

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 Test uses the following format:

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

6.1.1 Search Error Typeouts Cont.

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 in line B is the same as the direction in line C, the turn-around 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 = 000000 and the directions in lines B and C disagree. It could indicate the turnaround block, if the directions are the same and BLOCKS READ = 000000.

2. That the DECtape did not turn around in one PDP-9 block length, if BLOCKS READ = 000001, STAT B = 000100, and the directions indicated are the same.

3. The BLOCK in error, if BLOCKS READ does not = 000000 and STAT B is an error status (i.e., 600000 MARK TRACK ERROR) other than END ZONE (500000).

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 000100 ignore this line. If STAT B does = 000100, LAST BLOCK compared against BLOCK FOUND will indicate that the last two block numbers were not sequential or, that IORS Bit 10 is a 0.

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

g. This is the DECtape Status B register and the IORS status, if STAT B does not = 000100, or if bit 10 of IORS does not = 1, this is the error condition that caused the typeout. If STAT B equals 500000 (end-zone interrupt), and the directions in lines B and C are the same, it should mean that the drive made one turnaround and went the length of the tape without finding any blocks. 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 2 WANTED FWD or BLOCK 1075 WANTED BKWD).

6.1.1 Search Error Typeouts Cont.

The Start/Stop Turnaround Test adds one line to the format for search error typeouts and indicates the last valid tape position and the direction tape was moving at the completion of the last successful operation on that drive.

```
DRIVE 7
SEARCH FWD
000005 BLOCK WANTED FWD
000005 BLOCK FOUND
000003 BLOCK RD LAST
000003 BLOCKS READ
000100 STAT B 000200 IORS

000000 FWD LAST POS
```

This typeout would indicate that two block numbers in a row were read that were not sequential (last block was 3 and the next block read was 5 skipping block 4). The last valid tape position before Start Up the tape had been stopped at Block 0 going FWD.

```
DRIVE 7
SEARCH FWD
000005 BLOCK WANTED FWD
000006 BLOCK FOUND
000001 BLOCKS READ
000100 STAT B 000200 IORS

000004 BKWD LAST POS
```

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.

```
DRIVE 8
SEARCH FWD
001077 BLOCK WANTED BKWD
000773 BLOCK FOUND
000000 BLOCKS READ
600000 STAT B 000200 IORS
```

This typeout indicates a Mark-Track error (600000 Stat B). Block Found should be

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 and for Test 5 the direction the block was written. Again, depending upon which test routine is being run, one of several typeouts could occur.

6.1.2 Read Data Status Error Typeouts Cont.

```
DRIVE 1
READ DATA FWD
000046 BLOCK WRITTEN FWD
420100 STAT B  000200 IORS
```

This typeout indicates a Parity Error was detected reading Block 46 in the Forward direction

Note: If any Data Errors in the block, this typeout will be followed by a Data Error typeout (see Paragraph 6.1.5).

```
DRIVE 2
READ DATA BKWD
000100 BLOCK
777757 DATA
4201000 STAT B  000200 IORS
```

(This typeout is used by the Parity Generation Test. The Data line of 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
000100 BLOCK PARITY ERROR EXP
020000 DATA
000100 STAT B  000200 IORS
```

(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 Data 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 00 after reading the block. READ DATA and STATE CHECK going to 0 and 1 PB not equal to 77 is 1 to PARITY ERROR. See paragraph 7.7 for a complete description of the parity test.)

```
DRIVE 8
READ DATA FWD
000001 BLOCK
777400 WC
000100 STAT B  000200 IORS
```

In the read data typeouts, the contents of the word count register (address 30) are included only if the WC did not go to 0. Or, if the DECtape status B was normal (000100) 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 TC02 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 4
READ ALL FWD
CKSUM ERR
000200 BLOCK
040000 DATA                 (First word of block as read from tape.)
730000 DATA CKSUM EXP       (What the contents of the Checksum should be.)
770000 CKSUM READ           (Checksum from tape as Read-In Read-All.)
```

```
DRIVE 6
READ ALL FWD
CKSUM ERR
000020 BLOCK
757500 DATA                 (First Data word of block as read from tape.)
770000 DATA CKSUM EXP       (What the contents of the Checksum should be.)
750000 CKSUM READ           (Checksum from tape as Read-In Read-All)
```

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                     (This typeout indicates a mark-track error
WRITE DATA FWD              while doing a Write Data Forward on block
000765 BLOCK                0765. The WC typeout indicates that the
777715 WC                   error occurred with 63₈ words left to be
600000 STAT B  000200 IORS  written.)
```

```
DRIVE 7                     (If STAT B indicates a normal block interrupt
WRITE DATA BKWD             (000100) and the WC has not gone to 0, this
000005 BLOCK                typeout occurs.)
7777400 WC
000100 STAT B  000200 IORS
```

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 (including the direction that the block was written in.)

```
DRIVE 6
READ DATA FWD
000241 BLOCK WRITTEN FWD

252525 DATA WR 01400 ADRS      (Data Written and its Memory Location)
252527 DATA RD 014400 ADRS     (Data Read and its Memory Location)

525252 DATA WR 014001 ADRS
525256 DATA RD 014401 ADRS
```

7. PROGRAM DESCRIPTION

7.1 Basic Motion or Move Scope Loop (Routine 0)

This routine ma y be used as a visual verification of the tape motion operation of the DECtape drives and some sections of the TC02 control. Initially, the routine starts tape moving in the FWD direction and periodically monitors the tape motion bit (Status A Bit 4) and the contents of the Accumulator Switches.

The program utilizes the DTLA IOT to load a MOVE FWD (on the lowest drive selected) into the TC02 Status A register. Status A Bit 4 is then monitored, by the program for approximately 17 ms using the DTRA IOT.

If the motion bit does not go to 0 during the 17 ms, the program reads the accumulator switches. If the switches are all 0, the program then returns to the loop monitoring the tape motion bit. If the switches are not all 0, the program ISZ's a core memory location and when this memory register is equal to the contents of the AC switches or greater than the contents of the AC switches tape direction is reversed by the program, using the DTXA IOT with AC bit 3 on a 1.

If Bit 4 of Status A goes to 0 during the 17 ms loop, Status B is read. If Bit 2 of Status B is a 1, indicating End Zone, tape direction is reversed and tape motion is set to a 1 by a DTXA IOT with AC Bits 3 and 4 on a 1. If Bit 2 of Status B is a 0, tape motion is set to a 1 by a DTXA IOT with AC Bit 4 on a 1 and tape direction is not reversed.

7.2 Search Scope Loop (Test Routine 1)

This routine starts forward in search function and reverses direction at End Zones. The DECtape flag and all error status except end zone are ignored. The program starts forward in search and displays the last block number received in the AC while doing a series of IOT NOP's for approximately 36 ms. At completion of the IOT NOP 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 scope loop also monitors AC switches 5 to 17 and if they are not equal to 0, the program will increment a memory location and will cause the tape to search in the opposite direction if the contents of the memory equals or is greater than the switches.

The DECtape IOT's used are DTCA, DTRA, DTXA, and DTRB.

While the tape is moving backward, the AC should appear to increment and while tape is moving backward, the AC should appear to decrement.

AC switches 3 and 4 may be used to specify an extended memory bank and if they are not both equal to 0, the respective memory bank will be used for reading block numbers into.

7.3 Read Scope Loop (Test Routine 2)

This routine starts forward in read data and sets up to read in 256-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.

AC switches 3 and 4 may be used to specify an extended memory bank to read data into. None of the other AC switches are pertinent.

The DECtape IOT's exercised are DTLA, DTDF, DTEF, DTRA, DTXA, and DTRB. Unlike the MOVE and SEARCH Scope Loops, this routine uses the SKIP IOT's to monitor flags.

7.4 Write Data Scope Loop (Test Routine 3)

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 (picking up the next block in sequence) 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

7.4 Write Scope Loop (Test Routine 3) Cont.

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 15, 16, 17. See paragraph 7.6. The routine has to be restarted from 200 to change pattern or extended memory selections.

AC switches 3 and 4 may be used to select on extended memory bank for pattern generation and to output patterns from.

The DECtape IOT's exercised are DTLA, DTRA, DTXA, DTDF, DTEF, and DTRB.

7.5 Search Find All Blocks (Test Routine 4)

Before a program can verify that the DECtape system can write correctly, it must prove that the 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 a 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 1077; then moves the tape into forward end zone, reverses the tape and tests that blocks are numbered 1077 to 0000. Or, if PDP-7 format tapes are specified (SW 13=1 at Start), the blocks should be numbered 0000 to 1101 and 1101 to 0000.

7.6 Write/Read Data Test (Test Routine 5)

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, read backward. The sequence may be selected for 1 block at a time, 16 blocks at a time, or the lenght of tape. The program recylces 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	000000
1	777777
2	000000, 777777, 000000 (alternate words of 0's and 1's)
3	252525
4	525252
5	070707
6	707070
7	252525, 525252, 252525 (alternate words of 25's and 52's)

7.6 Write/Read Data Test (Test Routine 5) Cont.

The pattern to be written is selected in SWITCH REGISTER bit 15, 16, and 17. Place the number of the pattern desired in these switches.

Switches 13 and 14 are used to select the sequence of operation as follows:

SW13	SW14	Operation
0	0	Write and read sequence the length of the tape.
0	1	Write and read sequence in 16 block increments.
1	0	
or		Write and read sequence one block at a time.
1	1	
SW10=0		Take the next pattern to be exercised from SWs 15, 16, and 17.
SW10=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 1077 has been written backwards.
SW9=0		Type out parity error information and data errors.
SW9=1		Ignore parity and data errors. Mark track, timing, and select errors are not ignored.
SW12=0		Sequence from write to read data
SW12=1		Write data only.
SW11=0		Sequence from read data to write data.
SW11=1		Read data only (SW11 overrides SW12)
SW8=1		Stop at the WRITE PASS one direction or, stop at the End of a READ PASS both directions.

Switches 3 and 4 are monitored for extended memory selection at the start of each new pattern selection and this selection is held until that pattern has been exercised the length of tape.

7.7 Parity Generation and Checking Test (Test Routine 6)

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 charcters 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 of 77_8, 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 the 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 characters 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 util 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 as read forward or the last word of the block if read backward (actually same word but complement obverse if read backward).

b. This word will contain either one or two non-zero 6-bit characters that do not equal 77 (BKWD).

c. If there is only one 6-bit character, the LPB should be all 0's at the time it is strobed for parity error. This is true whether read occurred in a forward or backward direction.

7.7 Parity Genration and Checking Test (Test Routine 6) Cont.

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=020200, LPB=02.

e. If the read direction is backward and there are two characters not equal to 77, the LPB should have equaled the complement of one of the characters when it was strobed for parity error; i.e., WORD=775757, LPB=20.

Two switches apply to the Parity Test, switch 8 and switch 12.

SW8 = 1 is Halt at end of Each Pass

SW12 = 1 is Repeat Current Parity Test Pass

7.8 Basic Search Routine (Test Routine 7)

In this routine, the tape is searches in either direction until a series of 256 block numbers are 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 number read are sequential. When started the program types:

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

At this point, type in a "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 256 block numbers have been read and stored in memory. It then types:

XXXXXX	STATB	000200 IORS	If an Error Status other
XXXXXX	BLOCKS	READ	than DTF

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 no sequential. The program always types the first and last block numbers read as follows:

XXXXXX	FIRST
XXXXXX	LAST

7.8 Basic Search Routine (Test Routine 7) Cont.

The following series to typeouts would occur if the Search Direction was forward, 256 block numbers were read without an Error Status and the second selection was to compare block numbers:

```
DRIVE 7

TYPE IN F FOR FORWARD
ALL OTHERS BKWD F

TYPE IN C FOR COMPARE
ALL OTHERS PRINT C

000061  FIRST
000460  LAST
```

The following series to typeouts could occur if the Search Direction was backward and an END ZONE was detected while searching and the selection was to PRINT:

```
DRIVE 7

TYPE IN F FOR FORWARD
ALL OTHERS BKWD 8
5000000 STAT B 000200 IORS

000016 BLOCKS READ

TYPE IN C FOR COMPARE
ALL OTHERS PRINT P

000015   FIRST
000015   000014  000013  000012  000011  000010  000007  000006
000005   000004  000003  000002  000001  000000  LAST
```

7.9 Start/Stop/Turnaround Test (Test Routine 10)

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:

```
TURNAROUND Both direction on BLOCK 0 BDWD then FWD/STOP
Start FORWARD/STOP
Start BACKWARD/STOP
Start FORWARD/Wait UP TO SPEED/Turnaround/STOP
Start BACKWARD/Wait UP TO SPEED/Turnaround/STOP
```

The sequence is repeated for the length of tape. Turnaround is tested in both directions on block 1077.

7.9 Start/Stop/Turnaround Test (Test Routine 10) CONT

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. (This is a TU55 spec and should not be confused with the programming spec. Normal search operations should wait for 1 more block to pass before doing a turnaround.

The routine does each step of the operation for each drive selected. After each "STOP" for the last drive selected and before starting the next operation on the first drive, the program times out 1/2 second. Normally, the sequence will exercise all blocks equally the length of tape.

If AC SW12 is a 1 the program will repeat Start FORWARD/STOP twice and every sixth block will be exercised the length of tape.

```
                        .TITLE TC02B1
                        .ABS
                /TC02 BASIC EXERCISER PDP-15 - PART1
                /TAPE 1
                /EQUATE DECTAPE IOT'S
707541          DTCA=707541
707542          DTRA=707542
707544          DTXA=707544
707545          DTLA=DTCA!DTXA
707561          DTEF=707561
707562          DTRB=707562
707601          DTDF=707601
                /EQUATE DECTAPE FUNCTIONS (STAT A)
021000          SRCHFW=21000
061000          SRCHBW=61000
060000          MOVBKW=60000
022000          RDATAF=22000
005000          WRALL=5000
003000          RDALL=3000
                /BIT CONSTANTS EQUATED
020000          GOBIT=20000
040000          DIRBIT=40000
100000          EZBIT=100000
010000          MODEBT=10000
020000          PARBIT=20000
000100          DTFBIT=100
000400          ENABLI=400
775000          BLKTIM=LAW -3000        /ABOUT 36 MSEC
777400          BLENTH=LAW -400 /=256 FOR BLOCK WC -400 OCTAL
000030          WCLOC=30                        /WORD COUNT
000031          CALOC=31                        /CURRENT ADDRESS
006400          BUFFER=6400
007000          BUFFR2=BUFFER+400
000300          INHCLR=300
006377          BLKFND=BUFFER-1
007400          BUFFR3=BUFFR2+400
                /
                        .EJECT
```

```
                        /
                        /IN CASE OF CAL
00020                           .LOC 20
00020   000020                  20
00021   700304                  IORS-10
00022   700301                  KSF
00023   741000                  SKP
00024   740040                  HLT
00025   700042                  ION
00026   703344                  DBR
00027   620020                  JMP* 20
                        /
                                .EJECT
```

```
/STARTING ADDRESS IS 200
/SET SW REG TO SELECT DRIVES BIT0=DRIVE 0 SW1 IS DRV 1
/TO SW7 IS DRIVE 7
/SW14 TO SW17 SELECT TEST
/=0 IS MOVE TIMED TURN AROUND AC SWS=0 IS BOUNCE OFF EZ
/=1 IS SEARCH SCOPE LOOP TIMED TURN AROUND
/=2 IS READ DATA SCOPE LOOP BOUNCE OFF EZ
/=3 IS WRITE DATA SCOPE LOOP BOUNCE OFF EZ
/=4 IS SEARCH FIND ALL BLOCKS
/=5 IS WRITE AND REAAD DATA TEST
/=6 IS PARITY GENERATION TEST
/=7 IS READ 256 BLOCK NUMBERS AND COMPARE OR PRINT THEM
/=10 IS START = STOP = TA
/SW12 = 1  IS USE API FOR INTERRUPT
/
/PATTERN SELECTION FOR 3 AND 5 IS
/SW15 TO SW17
/=0 ALL ZEROS =1 ALL ONES =2 ALT WORD 0 WORD 1
/=3 IS 252525 =4 IS 525252 =5 IS 070707
/=6 IS 707070 =7 IS 252525 525252
/
/
/OTHER SWITCH CONTROLS TEST 5
/SWS = 10     IS 16 BLOCKS
/SWS = 20     OR 30 IS 1 BLOCK
/SWS = 40     IS WRITE ONLY
/SWS = 100    OR 140 IS READ ONLY
/SWS = 200    IS SEQUENCE PATTERNS
/SWS = 1000   IS HALT AFTER WRITE PASS OR READ PASS BOTH DIRECTIONS
/SWS = 400    IS DELETE ERROR CHECKING THAT DOES NOT STOP TAPE
/SWS 3,4 AND 5 NON 0 WILL USE EXTENDED MEMORY (4K AT A TIME)
/000 = BANK 0 =001 = BANK 1 =010 = BANK 2 =0 11 = BANK 3
/100 = BANK 4 =101 = BANK 5 =110 = BANK 6 =111 = BANK 7
/SWITCH CONTROL TEST 6
/SWS = 1000 IS HALT END OF PASS
/SWS = 40   IS REPEAT LAST PASS
/
/
/STANDARD
/SWITCH 0 IS DELETE ALL TYPEOUTS
/SWITCH 1 IS DELETE ERROR HALT
/SWITCH 2 IS TYPE ONLY 4 DATA ERRORS
        .EJECT
```

```
                    /SELECT TESTS SETUP FOR
                    /FIRST DRIVE 0 TO DRIVE TABLES
00200                       .LOC 200
00200   750004      TC02TS  LAS
00201   504555              AND (776000             /MASK DRIVE BITS
00202   740200              SZA
00203   600206              JMP .+3
00204   740040              HLT                     /TEST NUMBER TO HIGH
00205   600200              JMP TC02TS              /OR IF AC=0 NO DRIVES
00206   040643              DAC MSBITS
00207   750004              LAS
00210   504556              AND (17
00211   344557              TAD (TSTTBL-NDTTBL
00212   740100              SMA             /LEGAL TEST NUMBER
00213   600204              JMP TC02TS+4            /NO
00214   750004              LAS
00215   504556              AND (17
00216   344560              TAD (JMP* TSTTBL
00217   040244              DAC GOTST
00220   204561              LAC (1102
00221   040264              DAC EZBLOK
00222   344562              TAD (777777     /END BLOCK IS EZBLOK-1
00223   040265              DAC ENDBLK
00224   204563              LAC (DTRA+10
00225   041273              DAC WTFLGS+1
00226   750004              LAS
00227   504564              AND (40
00230   740200              SZA
00231   101345              JMS APISET
00232   200244              LAC GOTST
00233   740040              HLT
00234   100266              JMS RSFDRV              /GET FIRST DRIVE
00235   160340              DZM* POSITN     /0 DRIVE TABLES
00236   160341              DZM* DIRECT
00237   160342              DZM* LSBLOK
00240   100303              JMS NEWDRV              /DONE ALL DRIVES
00241   600235              JMP .-4                 /0 NEXT DRIVE TABLES
00242   204565              LAC (JMP* 2
00243   040001              DAC 1
00244   620245      GOTST   JMP* TSTTBL     /LOC CHANGE TO + TEST NUMBER
                            .EJECT
```

```
00245  000646      TSTTBL  PTTRNA
00246  000713              SREZLP
00247  000764              RDEZLP
00250  001024              WDEZLP
00251  001162              SRCHTS
00252  001543              WRARD
00253  002266              PARTST
00254  003210              SRCH2R
00255  003453              STRSTP
       000256      NDTTBL=.
00264                      .LOC TSTTBL+17
00264  000000      EZBLOK  0
00265  000000      ENDBLK  0
                           .EJECT
```

```
                        /RESET SELECTION TO FIRST DRIVE
00266   600266  RSFDRV  JMP .
00267   140641          DZM CDRIVE      /START WITH DRIVE 8
00270   204566          LAC (400000
00271   040642          DAC CBIT                /SET COMPARE BIT
00272   500643          AND MSBITS
00273   740200          SZA             /DRIVE SELECTED
00274   600301          JMP .+5         /YES
00275   200642          LAC CBIT                /SET UP
00276   440641          ISZ CDRIVE      /FOR NEXT DRIVE
00277   744020          RCR
00300   600271          JMP RSFDRV+3
00301   100322          JMS SETUNT              /GENERATE UNIT TABLE
00302   620266          JMP* RSFDRV     /ADDRESSES POS FOR IOTS
                        /SELECT NEXT DRIVE SKIP
                        /EXIT ADDRS IF LAST DRIVE PAST
00303   600303  NEWDRV  JMP .
00304   200642          LAC CBIT
00305   744020          RCR             /POSITION
00306   440641          ISZ CDRIVE
00307   544567          SAD (1000
00310   600317          JMP .+7
00311   040642          DAC CBIT
00312   500643          AND MSBITS
00313   741200          SNA             /DRIVE SELECTED
00314   600304          JMP NEWDRV+1    /NO TRY NEXT
00315   100322          JMS SETUNT      /SET UP THIS DRIVE
00316   620303          JMP* NEWDRV     /EXIT
00317   100266          JMS RSFDRV      /RESET TO FIRST DRIVE
00320   440303          ISZ NEWDRV      /STEP FOR SKIP
00321   620303          JMP* NEWDRV
00322   600322  SETUNT  JMP .
00323   200641          LAC CDRIVE
00324   744020          RCR             /POSITION DRIVE
00325   742020          RTR             /NUMBER FOR
00326   740020          RAR             /DECTAPE FUNCTIONS
00327   040631          DAC UNFUNC
00330   200641          LAC CDRIVE      /GENERATE TABLE
00331   344570          TAD (POSTBL     /ADDRESSES
00332   040340          DAC POSITN
00333   344571          TAD (10
00334   040341          DAC DIRECT
00335   344571          TAD (10
00336   040342          DAC LSBLOK
00337   620322          JMP* SETUNT
                        .EJECT
```

```
00340  000000  POSITN  0
00341  000000  DIRECT  0
00342  000000  LSBLOK  0
00343  000000  POSTBL  0
00353                  .LOC POSTBL+10  /RESERVE 8 LOCATIONS
00353  000000  DIRTBL  0               /EACH FOR DIRECTION
00363                  .LOC DIRTBL+10  /POSITION AND
00363  000000  LSTTBL  0               /LAST BLOCK WRITTEN
00373                  .LOC LSTTBL+10
                       /
                       /REWIND 1 DRIVE TO REVERSE ENDZONE
00373  600373  REWDRV  JMP .
00374  204572          LAC (GOBIT+DIRBIT
00375  340631          TAD UNFUNC
00376  707545          DTLA            /MOVE TAPE BKWD
00377  707572          DTRB+10         /WAIT FOR FLAGS
00400  741200          SNA
00401  600377          JMP .-2
00402  707562          DTRB
00403  544573          SAD (500000     /CAN ONLY BE ENDZONE
00404  600411          JMP .+5
00405  104423  MOVERR  JMS STPTAP              /MOVE ERROR TYPEOUT
00406  104304          JMS TYMOVE
00407  104153          JMS ERRHLT
00410  600374          JMP REWDRV+1
00411  700314          IORS            /EF SHOULD IORS
00412  504574          AND (200
00413  740200          SZA
00414  620373          JMP* REWDRV
00415  600405          JMP MOVERR
                       .EJECT
```

```
                        /SEARCH ROUTINE
                        /FIND BLOCK IN (RECORD) IN
                        /DIRECTION INDICATED BY DIRFLG=LAM IS BKWD
                        /DIRFLG=0 IS FWD
00416   600416  SEARCH  JMP .
00417   754000          CLA!CLL
00420   540633          SAD DIRFLG      /FORWARD
00421   744002          STL             /YES IS SET LINK
00422   204575          LAC (2
00423   741400          SZL             /BKWD IS+2
00424   777776          LAW -2          /FWD IS -2 FOR TA
00425   340634          TAD RECORD
00426   040635          DAC TAPONT      /TURN AROUND BLOCK
00427   344575          TAD (2
00430   540634          SAD RECORD      /FORWARD
00431   204576          LAC (SRCHBW ENABLI      /WANT BLOCK FWD
00432   544576          SAD (SRCHBW ENABLI      /START BKW IS WANT FWD
00433   741000          SKP
00434   204577          LAC (SRCHFW ENABLI      /START IS FWD TO FIND BKWD
00435   340631          TAD UNFUNC
00436   707545          DTLA
00437   750001          CLA!CMA
00440   040636          DAC FRSFLG
00441   100506          JMS SWTBLK
00442   600446          JMP DOTURN      /BLOCK=TAPOINT
00443   600446          JMP DOTURN      /BLOCK PAST TAPOINT
00444   707554          DTXA+10         /NOT RCHED TAPOINT
00445   600441          JMP .-4
                        /FOUND TURN AROUND POINT ON TAPE
                        /OR PASSED IT
00446   750001  DOTURN  CLA!CMA
00447   040636          DAC FRSFLG
00450   707552          DTRA+10
00451   740001          CMA
00452   504600          AND (GOBIT      /IN CASE GO CLRD
00453   344601          TAD (DIRBIT
00454   707544          DTXA            /CHNG DIRECTION
00455   200634          LAC RECORD
00456   040635          DAC TAPONT
00457   100506          JMS SWTBLK      /WAIT FOR BLOCK INT
00460   620416          JMP* SEARCH     /FOUND BLOCK EXIT
00461   600560          JMP SRERR1      /PAST BLOCK ERROR
00462   707554          DTXA+10         /NOT REACHED YET
00463   600457          JMP .-4         /WAIT NEXT BLOCK
                        .EJECT
```

```
                        /SEE IF ERROR INTERRUPT WAS
                        /PROPER END ZONE
00464   200635  SRCHEZ  LAC TAPONT
00465   540634          SAD RECORD      /AFTER TURN AROUND
00466   600560          JMP SRERR1      /EVERY ERR FLAG IS ERROR
00467   707572          DTRB*10
00470   504602          AND (EZBIT
00471   741200          SNA
00472   600560          JMP SRERR1              /NOT END ZONE INT
00473   200635          LAC TAPONT
00474   741100          SPA             /BLOCK 0 OR 1 TRW ARND
00475   600446          JMP DOTURN      /YES SEE IF BLK 0
00476   540264          SAD EZBLOK              /LAST ON TAPE TURN AROUND
00477   600446          JMP DOTURN      /YES
00500   344562          TAD (777777
00501   540264          SAD EZBLOK
00502   600446          JMP DOTURN
00503   440636          ISZ FRSFLG      /MORE THAN 1 BLOCK RD
00504   600560          JMP SRERR1      /YES SHOULD BE TURN ARND
00505   600446          JMP DOTURN      /BEFORE THIS
                        /WAIT FOR DTF OR ERR FLAG
                        /DETERMINE=GREATER OR LESS
                        /TEST BLOCK NUMBERS TO BE CONSECUTIVE
00506   600506  SWTBLK  JMP .
00507   204603          LAC (BLKFND
00510   040031          DAC CALOC
00511   101272          JMS WTFLGS              /WAIT FOR
00512   600464          JMP SRCHEZ              /ERROR OR
00513   206377          LAC BLKFND              /DECTAPE FLAG
00514   440636          ISZ FRSFLG
00515   600544          JMP SNOTFS
00516   504604          AND (7777
00517   046377          DAC BLKFND
00520   540635  SWTTES  SAD TAPONT             /BLOCK = LOOKED FOR
00521   600537          JMP SWTXIT      /YES GET OUT
                        .EJECT
```

```
00522   740001              CMA             /MAKE BLOCK FOUND
00523   300635              ADD TAPONT
00524   440506              ISZ SWTBLK      /STEP XIT ADDRESS
00525   741100              SPA             /BLOCK FOUND LESS OR
00526   600534              JMP .+6         /GREATER THAN SEARCHED
00527   707552              DTRA+10
00530   504601              AND (DIRBIT
00531   741200              SNA             /BLOCK FOUND LESSER
00532   440506              ISZ SWTBLK      /AND FWD DIR IS NOT REACHED
00533   600537              JMP SWTXIT
00534   707552              DTRA+10
00535   740001              CMA
00536   600530              JMP .-6
00537   200637      SWTXIT  LAC LSTBLK              /SAVE LAST TWO
00540   040640              DAC LSTBLK+1    /BLOCK NUMBERS
00541   206377              LAC BLKFND      /FOR ERROR INFO
00542   040637              DAC LSTBLK
00543   620506              JMP* SWTBLK
                    /NOT FIRST BLOCK NUMBER
                    /SEE IF CONSECUTIVE WITH LAST
00544   707552      SNOTFS  DTRA+10
00545   504601              AND (DIRBIT
00546   751200              SNA!CLA         /FORWARD GOING
00547   204605              LAC (1          /IS +1
00550   741200              SNA             /BACKWARD
00551   777777              LAW -1          /IS -1
00552   340637              TAD LSTBLK
00553   546377              SAD BLKFND
00554   741000              SKP
00555   600560              JMP SRERR1
00556   206377              LAC BLKFND
00557   600520              JMP SWTTES
00560   100563      SRERR1  JMS SRERRO
00561   104153              JMS ERRHLT
00562   600417              JMP SEARCH+1
                            .EJECT
```

```
00563   600563      SRERRO  JMP .
00564   104423              JMS STPTAP
00565   104316              JMS TYSRCH
00566   104124              JMS TYPTEX
00567   006741              6741            /WA
00570   566445              566445          /NTE
00571   447700              447700          /D EOM
00572   200633              LAC DIRFLG
00573   740200              SZA
00574   600577              JMP .+3
00575   104411              JMS TYFWD
00576   741000              SKP
00577   104416              JMS TYBKW
00600   206377              LAC BLKFND
00601   104467              JMS TYPBLK
00602   104124              JMS TYPTEX
00603   004657              4657            /FO
00604   655644              655644          /UND
00605   770000              770000          /EOM
00606   777777              LAW -1
00607   340636              TAD FRSFLG
00610   741100              SPA
00611   600620              JMP .+7
00612   200637              LAC LSTBLK
00613   104467              JMS TYPBLK
00614   104124              JMS TYPTEX
00615   006244              6244            /RD
00616   005441              5441            /LA
00617   636477              636477          /ST EOM
00620   200636              LAC FRSFLG
00621   344605              TAD (1
00622   104467              JMS TYPBLK
00623   104124              JMS TYPTEX
00624   630062              630062          /S R
00625   454144              454144          /EAD
00626   770000              770000          /EOM
00627   104447              JMS TYSTAB
00630   620563              JMP* SRERRO
00631   000000      UNFUNC  0
00632   000000      NDEX    0
00633   000000      DIRFLG  0
00634   000000      RECORD  0
00635   000000      TAPONT  0
00636   000000      FRSFLG  0
00637   000000      LSTBLK  0
00640   000000              0
00641   000000      CDRIVE  0
00642   000000      CBIT    0
00643   000000      MSBITS  0
00644   000000      TIMCTR  0
00645   000000      NDEX1   0
                    .EJECT
```

```
                        /PROGRAM TIMED TURN AROUND
                        /USES MOVE ONLY TIME FROM SWS
00646   204600  PTTRNA  LAC (GOBIT              /MOVE FORWARD
00647   340631          TAD UNFUNC      /+ DRIVE NUMBER
00650   707545          DTLA            /START TAPE
00651   140644          DZM TIMCTR
00652   775000          BLKTIM
00653   040632          DAC NDEX
00654   707552          DTRA+10
00655   504600          AND (GOBIT
00656   741200          SNA             /GO BIT STILL SET
00657   600702          JMP PTTEZ               /NO
00660   440632          ISZ NDEX                /1 BLOCK TIME PASSED
00661   600654          JMP .-5         /NO
00662   750004          LAS
00663   741200          SNA             /WAIT FOR EZ
00664   600651          JMP PTTRNA+5    /IF AC = 0
00665   440644          ISZ TIMCTR      /COUNT NUMBER OF BLOCKS
00666   540644          SAD TIMCTR
00667   600674          JMP .+5
00670   740001          CMA             /IN CASE SWITCHES
00671   300644          ADD TIMCTR      /WERE CHNGD
00672   741100          SPA
00673   600652          JMP PTTRNA+4
00674   707552          DTRA+10
00675   740001          CMA
00676   504600          AND (GOBIT
00677   344601          TAD (DIRBIT
00700   707544          DTXA
00701   600651          JMP PTTRNA+3
00702   707572  PTTEZ   DTRB+10
00703   504602          AND (EEBIT
00704   741200          SNA             /END ZONE
00705   600711          JMP .+4         /NO GO JUST CLEARED
00706   204572          LAC (GOBIT+DIRBIT
00707   707544          DTXA
00710   600651          JMP PTTRNA+3
00711   204600          LAC (GOBIT
00712   600707          JMP .-3
                        .EJECT
```

```
                        /SEARCH BOUNCE OFF EZ OR
                        /TURN AROUND TIMED AC SWITCHS
00713   204606  SREZLP  LAC (SRCHFW
00714   340631          TAD UNFUNC
00715   707545          DTLA            /START TAPE FWD
00716   140644          DZM TIMCTR      /ZERO BLOCK COUNTER
00717   750004          LAS
00720   504607          AND (70000
00721   344610          TAD (BUFFER
00722   040031          DAC CALOC               /SET CA
00723   773240          LAW 13240
00724   040632          DAC NDEX                /ABOUT 36 MSEC
00725   707702          EEM
00726   220031          LAC* CALOC
00727   707704          LEM
00730   700000          IOT             /SHOULD CAUSE
00731   700000          IOT             /BLOCK NUMBER TO
00732   700000          IOT             /SHOW IN AC
00733   440632          ISZ NDEX
00734   600730          JMP .-4         /TIME OUT
00735   707572          DTRB+10
00736   504602          AND (EZBIT
00737   740200          SZA             /END ZONE SET
00740   600756          JMP SREZTA      /YES TURN AROUND
00741   707552          DTRA+10
00742   740001          CMA
00743   504600          AND (GOBIT
00744   707544          DTXA            /RESET GO IF NECESSARY
00745   750004          LAS
00746   504611          AND (17777
00747   741200          SNA             /TIMEOUT OR
00750   600716          JMP SREZLP+3    /WAIT FOR EZ TO TA
00751   440644          ISZ TIMCTR
00752   740001          CMA
00753   300644          ADD TIMCTR
00754   741100          SPA             /TIMED OUT THIS DIR
00755   600717          JMP SREZLP+4    /NO
00756   707552  SREZTA  DTRA+10
00757   740001          CMA             /IN CASE GO ALRDY=1
00760   504600          AND (GOBIT
00761   344601          TAD (DIRBIT
00762   707544          DTXA            /TURN AROUND
00763   600716          JMP SREZLP+3
                        .EJECT
```

```
                        /READ DATA BOUNCE OFF  EZ LOOP
00764   204612  RDEZLP  LAC (RDATAF
00765   340631          TAD UNFUNC
00766   707545          DTLA
00767   777400          BLENTH          /LAM BLOCK WORD COUNT
00770   040030          DAC WCLOC
00771   750004          LAS
00772   504607          AND (70000
00773   344613          TAD (BUFFR2-1   /FOR EXTENDED MEMORY BITS
00774   040031          DAC CALOC
00775   200030          LAC WCLOC
00776   741200          SNA             /WC GO TO 0
00777   601002          JMP .+3         /YES
01000   707561          DTEF            /ERROR FLG SET
01001   741000          SKP             /NO
01002   601005          JMP .+3
01003   707601          DTDF            /DECTAPE FLG
01004   600775          JMP .-7         /=0 WAIT
01005   777500          LAW -300
01006   040632          DAC NDEX
01007   440632          ISZ NDEX                /TIME SAFE TO XOR TO
01010   601007          JMP .-1         /STATUS A
01011   707572          DTRB+10
01012   504602          AND (EZBIT
01013   740200          SZA             /EZ SET
01014   601022          JMP .+6         /YES TURN AROUND
01015   707552          DTRA+10
01016   740001          CMA             /IN CASE GO CLRD
01017   504600          AND (GOBIT
01020   707544          DTXA
01021   600767          JMP RDEZLP+3
01022   204572          LAC (MOVBKW
01023   601020          JMP .-3
                        .EJECT
```

```
                        /WRITE DATA BOUNCE OFF EZ
                        /SCOPE LOOP
01024   750004 WDEZLP   LAS
01025   504614          AND (7
01026   041160          DAC PATNUM
01027   101077          JMS GSTPAT      /GENERATE PATTERN
01030   204606          LAC (SRCHFW
01031   340631          TAD UNFUNC
01032   707545          DTLA            /START SRCH FWD
01033   204603          LAC (BLKFND
01034   040031          DAC CALOC
01035   707561          DTEF            /WAIT FOR
01036   741000          SKP             /FLAGS
01037   601067          JMP WDEZTA      /TEST FOR EZ
01040   707601          DTDF
01041   601035          JMP .-4
01042   707561          DTEF
01043   741000          SKP
01044   601067          JMP WDEZTA      /SEE IF EZ
01045   204615          LAC (WRALL
01046   707544          DTXA            /CHNG TO WRITE DATA
01047   777400          BLENTH          /LAM =BLOCK WOK
01050   040030          DAC WCLOC
01051   204603          LAC (BUFFER-1
01052   341161          TAD XTDFLD              /ADD EXTEND BITS
01053   040031          DAC CALOC
01054   707561          DTEF            /WAIT FOR FLGS
01055   741000          SKP
01056   601061          JMP .+3
01057   707601          DTDF
01060   601054          JMP .-4
01061   707552          DTRA+10         /IN CASE GO CLRD
01062   740001          CMA
01063   504600          AND (GOBIT
01064   344615          TAD (WRALL
01065   707544          DTXA            /CHNG BACK TO SEARCH
01066   601033          JMP WDEZLP+7
01067   707572 WDEZTA   DTRB+10
01070   504602          AND (EZBIT
01071   741200          SNA             /EZ INTERRUPT
01072   601075          JMP .+3         /NO
01073   204572          LAC (MOVBKW     /YES
01074   601065          JMP WDEZTA-2    /CHANGE DIRECTION
01075   204600          LAC (GOBIT
01076   601065          JMP WDEZTA-2
                        .EJECT
```

```
                        /GENERATE 1 OF 8 STANDARD PATTERNS
01077   601077  GSTPAT  JMP .
01100   777400          BLENTH
01101   040632          DAC NDEX
01102   750004          LAS
01103   504607          AND (70000
01104   041161          DAC XTDFLD
01105   344603          TAD (BUFFER-1
01106   040010          DAC 10
01107   201160          LAC PATNUM
01110   504614          AND (7
01111   344616          TAD (PATTBL
01112   040645          DAC NDEX1
01113   201161          LAC XTDFLD
01114   740200          SZA                 /EXTENDED MEORY IND
01115   707702          EEM                 /YES SET EXTEND MODE
01116   620645          JMP* NDEX1
01117   601127  PATTBL  JMP PAT0
01120   601131          JMP PAT1
01121   601133          JMP PAT2
01122   601135          JMP PAT3
01123   601140          JMP PAT4
01124   601142          JMP PAT5
01125   601144          JMP PAT6
01126   601146          JMP PAT7
01127   754000  PAT0    CLA!CLL
01130   601150          JMP GENPAT
01131   754001  PAT1    CLC!CLL
01132   601150          JMP GENPAT
01133   754002  PAT2    CLA!STL
01134   601150          JMP GENPAT
01135   204617  PAT3    LAC (252525
01136   744000          CLL
01137   601150          JMP GENPAT
01140   204620  PAT4    LAC (525252
01141   601136          JMP PAT3+1
01142   204621  PAT5    LAC (070707
01143   601136          JMP PAT3+1
01144   204622  PAT6    LAC (707070
01145   601136          JMP PAT3+1
01146   204617  PAT7    LAC (252525
01147   744002          STL
                        .EJECT
```

```
01150   060010      GENPAT  DAC* 10
01151   440632              ISZ NDEX
01152   601155              JMP .+3         /NOT DONE YET SEE IF CMP
01153   707704              LEM
01154   621077              JMP* GSTPAT
01155   741400              SZL
01156   740001              CMA
01157   601150              JMP GENPAT
01160   000000      PATNUM  0
01161   000000      XTDFLD  0
                            .EJECT
```

```
                        /TEST FOR CONSECUTIVE BLOCK
                        /NUMBERS LENGTH OF TAPE
01162   100373  SRCHTS  JMS REWDRV              /MOVE TAPE INTO REVERSE END ZONE
01163   100303          JMS NEWDRV
01164   601162          JMP .-2
01165   140634  FINDAL  DZM RECORD              /BLOCK 0
01166   140633          DZM DIRFLG              /FWD
01167   100416          JMS SEARCH
01170   440634          ISZ RECORD              /COUNT FOR NEXT BLOCK
01171   707554          DTXA+10         /RESET ENABLES
01172   206377          LAC BLKFND
01173   040637          DAC LSTBLK
01174   101272          JMS WTFLGS              /WAIT FOR PROGRAM INT
01175   601212          JMP FINDEZ              /ERROR STATUS RETURN
01176   206377          LAC BLKFND
01177   440636          ISZ FRSFLG              /COUNT BLOCKS READ
01200   540634          SAD RECORD
01201   601170          JMP FINDAL+3    /RECORD FOUND OK
01202   100563  SRFERR  JMS SRERRO
01203   104153          JMS ERRHLT
01204   200634          LAC RECORD
01205   440634          ISZ RECORD
01206   344605          TAD (1
01207   540264          SAD EZBLOK              /LAST BLOCK ON TAPE
01210   601216          JMP .+6         /YES
01211   601167          JMP FINDAL+2
01212   707572  FINDEZ  DTRB+10         /READ DECTAPE STATUS
01213   504602          AND (EZBIT
01214   741200          SNA             /END ZONE
01215   601202          JMP SRFERR              /NO, ALL OTHER STATUS IS ERROR
01216   200634          LAC RECORD
01217   540264          SAD EZBLOK              /END ZONE IN ERROR
01220   741000          SKP             /NO
01221   601202          JMP SRFERR
01222   100303          JMS NEWDRV              /SKIP N.I. IF ALL DRIVE DONE
01223   601165          JMP FINDAL
01224   601226          JMP SRBACK
01225   601202          JMP SRFERR
                        .EJECT
```

```
                        /SEARCH BKWDS FIND
                        /BLOCK NUMBERS DECREMENT FOR
                        /LENGTH OF TAPE
01226   200265  SRBACK  LAC ENDBLK              /LAST BLOCK ON TAPE
01227   040634          DAC RECORD
01230   777777          LAW -1
01231   040633          DAC DIRFLG              /BACKWARDS
01232   100416          JMS SEARCH              /FIND IT
01233   707554          DTXA+10         /RESET ENABLES
01234   777777          LAW -1
01235   340634          TAD RECORD              /LAST BLOCK -1
01236   040634          DAC RECORD
01237   206377          LAC BLKFND
01240   040637          DAC LSTBLK              /FOR TYPEOUTS
01241   101272          JMS WTFLGS              /WAIT FOR INT,
01242   601257          JMP BKWDEZ              /ERROR STATUS RETURN
01243   440636          ISZ FRSFLG              /COUNT BLOCKS
01244   206377          LAC BLKFND
01245   540634          SAD RECORD              /RIGHT BLOCK
01246   601233          JMP SRBACK+5    /YES WAIT FOR NEXT
01247   100563  SRBERR  JMS SRERRO              /
01250   104153          JMS ERRHLT
01251   200634          LAC RECORD
01252   344562          TAD (777777
01253   040634          DAC RECORD
01254   741100          SPA             /DONE BLOCK 0
01255   601263          JMP .+6         /YES
01256   601232          JMP SRBACK+4
01257   707572  BKWDEZ  DTRB+10         /READ DECTAPE STATUS
01260   504602          AND (EZBIT              /MASK FOR EZ ERROR
01261   741200          SNA             /EZ
01262   601247          JMP SRBERR              /NO ALL OTHER STATUS IS ERROR
01263   200634          LAC RECORD
01264   740100          SMA             /PAST BLOCK 0
01265   601247          JMP SRBERR              /NO EZ WAS IN ERROR
01266   100303          JMS NEWDRV              /TAKE N.I. IF NOT DONE
01267   601226          JMP SRBACK              /ALL DRIVES
01270   601165          JMP FINDAL
01271   601247          JMP SRBERR
                        .EJECT
```

```
                        /WAIT FOR INTERRUPT TEST FLAGS AND BIT 12 OF IORS
01272   601272  WTFLGS  JMP .           /INT ROUTINE IS ISZ 0
01273   707552          DTRA+10         /READ STAT A
01274   504623          AND (7000               /MASK OFF F
01275   740200          SZA             /IS IT MOVE FUNTION
01276   777770          LAW -10         /NO 5 SECOND WAIT
01277   741200          SNA             /MOVE GET LAW -110
01300   777670          LAW -110                /MOVE IS 45 SECONDS
01301   041542          DAC CLFCTR
01302   140007  WTPION  DZM 7
01303   204624          LAC (WTTOK
01304   040002          DAC 2
01305   700042          ION             /
01306   440007          ISZ 7
01307   601306          JMP .-1
01310   441542          ISZ CLFCTR
01311   601306          JMP .-3
01312   700002          IOF
01313   707552  WTTOUT  DTRA+10         /HERE IF NO INTERRUPT
01314   504600          AND (GOBIT
01315   344625          TAD (INHCLR
01316   707544          DTXA            /STOP TAPE
01317   104124          JMS TYPTEX              /NO P,I,
01320   777777          777777          /TYPE OUT
01321   777777          777777
01322   565700          565700          /NO
01323   601651          601651          /P,I
01324   167700          167700          /, EOM
01325   104447          JMS TYSTAB
01326   621272          JMP* WTFLGS     /FOLLOWED BY OPERATION
01327   707561  WTTOK   DTEF            /RETURN AFTER INT
01330   741000          SKP
01331   621272          JMP* WTFLGS     /EF = 1
01332   707601          DTDF
01333   621272          JMP* WTFLGS     /DTF = 0
01334   707572          DTRB+10
01335   504626          AND (DTFBIT
01336   741200          SNA
01337   621272          JMP* WTFLGS     /DTF DOES NOT = 1
01340   700314          IORS
01341   504574          AND (200
01342   740200          SZA             /DOES BIT 10 OF IORS = DTF
01343   441272          ISZ WTFLGS              /SKIP FOR ALL OK EXIT
01344   621272          JMP* WTFLGS
                        .EJECT
```

```
                        /SELECTION IS TO RUN INTERRUPTS
                        /OFF FROM API INSTEAD OF PI
                        /SET UP WTFLGS TO JMP TO WAIT AP
                        /
01345   601345  APISET  JMP .
01346   204627          LAC (JMP WAITAP /INSTEAD OF
01347   041273          DAC WTFLGS+1    /DTRA MAKE IT A JMP
01350   204630          LAC (37
01351   040010          DAC 10          /TO STORE JMS
01352   204631          LAC (JMS CH0ERR /GET FIRST API ERROR JMS
01353   060010          DAC* 10         /STORE INTO CHAN LOC
01354   544632          SAD (JMS CH37ER /SET UP ALL CHAN
01355   601360          JMP .+3         /YES SET UP DT AND RTC
01356   344575          TAD (2          /ERR JMS +2
01357   601353          JMP .-4         /STORE IT
01360   204633          LAC (JMS DTAPI  /SET UP DECTAPE
01361   040044          DAC 44          /API CHANNEL
01362   204634          LAC (JMS CLFAPI /API CHANNEL
01363   040051          DAC 51          /RTC API CHANNEL
01364   621345          JMP* APISET     /EXIT
                        /
                        /STARTING SELECTION WAS TO USE
                        /API INSTEAD OF PI
                        /
01365   707552  WAITAP  DTRA+10
01366   504623          AND (7000
01367   740200          SZA
01370   777324          LAW -454                /5 SEC COUNT NOT MOVE
01371   741200          SNA
01372   773614          LAW -4164               /45 SEC COUNT MOVE
01373   041542          DAC CLFCTR              /TO COUNT CLOCK INT
01374   744000          CLL
01375   777777          LAW -1
01376   040007          DAC 7           /FOR CLOCK TO INT
01377   140000          DEM 0
01400   204635          LAC (WTAPI+2
01401   040002          DAC 2
01402   204566          LAC (400000
01403   700044          CLON            /TURN CLOCK ON
01404   700042          ION             /PIF ON
01405   705504          ISA             /ENABLE API
01406   000000  WTAPI   CAL             /WAIT FOR BREAKS
01407   601406          JMP .-1
01410   740040          HLT             /GOT A PROGRAM INTERRUPT ERROR
                        /
                        /IF HALT IS EXECUTED PROBABLY CAUSED BY
                        /PROGRAM INTERRUPT INSTEAD OF API
                        /CHECK ADDRESS 0 TO SEE IF IT POINTS TO HLT
                        .EJECT
```

```
                        /API INTERRUPT ON CHANNEL 44 (DECTAPE)
                        /IS JMS TO DTAPI
                        /
01411   601411  DTAPI   JMP .
01412   700002          IOF             /DISABLE PIE
01413   705514          ISA+10          /TURN API OFF
01414   703304          DBK             /CLEAR DECTAPE ACTIVE
01415   201411          LAC DTAPI
01416   544636          SAD (WTAPI              /COME FROM CLOCK
01417   741000          SKP             /NO
01420   703304          DBK             /CLEAR CLOCK FLAG
01421   200000          LAC Ø
01422   740200          SZA             /PI OCCUR ALSO
01423   740040          HLT             /API DID'NT BLOCK PI
01424   601327          JMP WTTOK               /TEST DECTAPE FLAGS
                        /
01425   601425  CH0ERR  JMP .           /API ERROR
01426   740040          HLT             /EXECUTED JMS IN 40
01427   601427  CH1ERR  JMP .
01430   740040          HLT             /EXECUTED 41
01431   601431  CH2ERR  JMP .
01432   740040          HLT             /EXECUTED 42
01433   601433  CH3ERR  JMP .
01434   740040          HLT             /EXECUTED 43
01435   601435  CH4ERR  JMP .           /CAN'T GET HERE
01436   740040          HLT             /44 GOES TO DTAPI
01437   601437  CH5ERR  JMP .
01440   740040          HLT             /EXECUTED 45
01441   601441  CH6ERR  JMP .
01442   740040          HLT             /EXECUTED 46
01443   601443  CH7ERR  JMP .
01444   740040          HLT             /EXECUTED 47
01445   601445  CH10ER  JMP .
01446   740040          HLT             /EXECUTED 50
01447   601447  CH11ER  JMP .           /CAN'T GET HERE CLOCK GOES TO CLFAPI
01450   740040          HLT             /EXECUTED 51
01451   601451  CH12ER  JMP .
01452   740040          HLT             /EXECUTED 52
01453   601453  CH13ER  JMP .
01454   740040          HLT             /EXECUTED 53
                        .EJECT
```

```
01455   601455      CH14ER  JMP .
01456   740040              HLT     /EXECUTE 54
01457   601457      CH15ER  JMP .
01460   740040              HLT     /EXECUTE 55
01461   601461      CH16ER  JMP .
01462   740040              HLT     /EXECUTE 56
01463   601463      CH17ER  JMP .
01464   740040              HLT     /EXECUTE 57
01465   601465      CH20ER  JMP .
01466   740040              HLT     /EXECUTE 60
01467   601467      CH21ER  JMP .
01470   740040              HLT     /EXECUTE 61
01471   601471      CH22ER  JMP .
01472   740040              HLT     /EXECUTE 62
01473   601473      CH23ER  JMP .
01474   740040              HLT     /EXECUTE 63
01475   601475      CH24ER  JMP .
01476   740040              HLT     /EXECUTE 64
01477   601477      CH25ER  JMP .
01500   740040              HLT     /EXECUTE 65
01501   601501      CH26ER  JMP .
01502   740040              HLT     /EXECUTE 66
01503   601503      CH27ER  JMP .
01504   740040              HLT     /EXECUTE 67
01505   601505      CH30ER  JMP .
01506   740040              HLT     /EXECUTE 70
01507   601507      CH31ER  JMP .
01510   740040              HLT     /EXECUTE 71
01511   601511      CH32ER  JMP .
01512   740040              HLT     /EXECUTE 72
01513   601513      CH33ER  JMP .
01514   740040              HLT     /EXECUTE 73
01515   601515      CH34ER  JMP .
01516   740040              HLT     /EXECUTE 74
01517   601517      CH35ER  JMP .
01520   740040              HLT     /EXECUTE 75
01521   601521      CH36ER  JMP .
01522   740040              HLT     /EXECUTE 76
01523   601523      CH37ER  JMP .
01524   740040              HLT     /EXECUTE 77
01525   601525              JMP .
                    .EJECT
```

```
                        /CLOCK INTERRUPT FROM API
                        /COMES TO CLFAPI
                        /
01526   601526  CLFAPI  JMP .
01527   441542          ISZ CLFCTR              /WAITED MAX TIME
01530   601535          JMP .+5         /NO
01531   700002          IOF             /TIMED OUT
01532   705514          ISA+10          /DISABLE API AND PIE
01533   703304          DBK             /CLEAR CLOCK ACTIVE
01534   601313          JMP WTTOUT              /TYPEOUT
01535   777777          LAW -1
01536   040007          DAC 7           /SET FOR CLOCK INT
01537   700044          CLON            /ENABLE CLOCK
01540   703344          DBR             /CLEAR CLOCK ACTIVE
01541   621526          JMP* CLFAPI     /GO BACK TO WTAPI
                        /
01542   000000  CLFCTR  0               /TO COUNT CLOCK INTR
                        /TC02 BASIC EXERCISER TAPE 2
                        /WRITE DATA PATTERNS 0 TO 7 FROM SWITCHES
                        /WRITE AND READ DATA LENGTH 16 BLOCKS OR 1 BLOCK OF TAPE SWITCHES
01543   750004  WRARD   LAS
01544   504614          AND (7
01545   041160          DAC PATNUM
01546   750004          LAS
01547   504626          AND (100
01550   740200          SZA
01551   601622          JMP RDLOOP+2
01552   101077          JMS GSTPAT              /GENERATE PATTERN
01553   160340          DZM* POSITN
01554   160341          DZM* DIRECT     /0 DRIVE TABLES
01555   160342          DZM* LSBLOK
01556   100303          JMS NEWDRV      /DO ALL DRIVES
01557   601553          JMP .-4
01560   204605          LAC (1
01561   042260          DAC BLKINC
01562   140633          DZM DIRFLG
01563   220342  WRFIRS  LAC* LSBLOK     /GET LAST WRITE
01564   340633          TAD DIRFLG
01565   040634          DAC RECORD      /FWD IS 1 AHEAD
01566   100416          JMS SEARCH
01567   204615          LAC (WRALL
01570   707544          DTXA            /CHNG TO WRITE DATA
01571   777400          BLENTH          /LAM -WC
01572   040030          DAC WCLOC
01573   204603          LAC (BUFFER-1
01574   341161          TAD XTDFLD
01575   040031          DAC CALOC
01576   101752          JMS WRSWS               /TEST SWITCH OP
01577   601570          JMP .-7         /NOT DONE ALL
01600   100303          JMS NEWDRV              /DONE ALL DRIVES
01601   601563          JMP WRFIRS              /NO DO WRITE PASS ON NEXT DRIVE
01602   200633          LAC DIRFLG
01603   042261          DAC WRPASD
01604   740001          CMA
01605   040633          DAC DIRFLG      /SET UP OTHER
```

```
01606   202260              LAC BLKINC      /DIRECTION
01607   740001              CMA
01610   344605              TAD (1
01611   042260              DAC BLKINC
01612   750004              LAS
01613   504567              AND (1000
01614   740200              SZA
01615   740040              HLT
01616   750004              LAS
01617   504564              AND (40
01620   740200              SZA             /WRITE ONLY
01621   601714              JMP RDTERM      /YES
01622   777777              LAW -1
01623   042257              DAC RPASFL      /SET 1ST RD FLG
                            .EJECT
```

```
                        /READ LOOP LENGTH OF TAPE 1 OR 16 BLOCKS
01624   220343  RDLOOP  LAC* POSITN     /GET NEXT BLOCK
01625   040634          DAC RECORD      /NUMBER
01626   100416          JMS SEARCH      /FIND IT
01627   204637          LAC (RDALL
01630   707544          DTXA            /TO RD DATA
01631   777400          BLENTH          /LAM=WC
01632   040030          DAC WCLOC
01633   204613          LAC (BUFFR2-1   /SET CA AND WC
01634   341161          TAD XTDFLD
01635   040031          DAC CALOC
01636   102046  RDFIRS  JMS RDSWS               /WAIT FLAGS RD SWS
01637   601672          JMP RDLEND      /LAST BLOCK RD
01640   707554          DTXA+10         /RESET ENABLES
01641   777400          BLENTH
01642   040030          DAC WCLOC               /NEXT BLOCK
01643   204640          LAC (BUFFR3-1   /INTO 3RD BUFFR
01644   341161          TAD XTDFLD
01645   040031          DAC CALOC
01646   102142          JMS CODATA      /COMPARE WRITTEN
01647   007000          BUFFR2          /AGAINST READ
01650   200635          LAC TAPONT              /NUMBER OF NEXT
01651   040634          DAC RECORD              /BLOCK IN SEQUENCE
01652   102132          JMS RDERCK      /CHECK ERRORS
01653   601626          JMP RDLOOP+2    /AND RESTART
01654   102046          JMS RDSWS               /WAIT DONE BLOCK
01655   601672          JMP RDLEND      /RD ENDED
01656   707554          DTXA+10         /RESET ENABLES
01657   777400          BLENTH
01660   040030          DAC WCLOC               /SET WC
01661   204613          LAC (BUFFR2-1   /AND CA
01662   040031          DAC CALOC
01663   102142          JMS CODATA      /COMPARE DATA
01664   007400          BUFFR3          /AGAINST 3RD BUF
01665   200635          LAC TAPONT              /NUMBER OF NEXT
01666   040634          DAC RECORD              /BLOCK IN SEQUENCE
01667   102132          JMS RDERCK
01670   601626          JMP RDLOOP+2    /ERROR, RESYNC
01671   601636          JMP RDFIRS      /NO ERROR
                        .EJECT
```

```
                    /LOOP BACK AND WAIT 2ND BUFFR
                    /TO FILL UP
01672   777401     RDLEND  BLENTH+1
01673   340031             TAD CALOC
01674   504611             AND (17777
01675   041677             DAC .+2
01676   102142             JMS CODATA      /CHECK LAST DATA
01677   007000             BUFFR2
01700   100303             JMS NEWDRV              /DONE READ PASS ON ALL DRIVES
01701   601624             JMP RDLOOP              /NO START READ PASS ON NEXT
01702   200633             LAC DIRFLG
01703   740001             CMA             /MAKE CONSTANTS
01704   040633             DAC DIRFLG      /OTHER DIRECTION
01705   202260             LAC BLKINC
01706   740001             CMA
01707   344605             TAD (1
01710   042260             DAC BLKINC
01711   442257             ISZ RPASFL      /READ BOTH DIRECT
01712   741000             SKP             /YES
01713   601624             JMP RDLOOP
01714   750004     RDTERM  LAS
01715   504567             AND (1000
01716   740200             SZA
01717   740040             HLT
01720   750004             LAS
01721   504626             AND (100
01722   740200             SZA             /RD ONLY
01723   601622             JMP RDLOOP+2
01724   200633             LAC DIRFLG
01725   740200             SZA
01726   601563             JMP WRFIRS
01727   220342             LAC* LSBLOK
01730   540264             SAD EZBLOK              /DONE LENGTH TAPE
01731   741000             SKP             /DONE 1 PASS
01732   601563             JMP WRFIRS      /DO NEXT SEQUENCE
01733   104276             JMS TYCRLF
01734   201160             LAC PATNUM
01735   344641             TAD (260
01736   104172             JMS TYPCHA
01737   104124             JMS TYPTEX
01740   004556             4556            /EN
01741   447700             447700          /D EOM
01742   750004             LAS
01743   504574             AND (200
01744   741200             SNA
01745   601543             JMP WRARD
01746   201160             LAC PATNUM
01747   344605             TAD (1
01750   504614             AND (7
01751   601545             JMP WRARD+2
                           .EJECT
```

```
                        /WAIT FOR WRITE FLAGS
                        /TEST SWITCH OPTIONS
01752   601752  WRSWS   JMP .
01753   101272          JMS WTFLGS      /WAIT FOR FLGS
01754   602027          JMP WREZTS      /NOT NORMAL STATUS
01755   200030          LAC WCLOC
01756   740200          SZA
01757   602040          JMP WRSERR
01760   200634          LAC RECORD
01761   060340          DAC* POSITN     /NEW POSITION
01762   342260          TAD BLKINC
01763   040634          DAC RECORD      /NEXT TO BE WRITTEN
01764   750004          LAS
01765   504642          AND (30
01766   741200          SNA             /LENGTH OF TAPE
01767   621752          JMP* WRSWS              /YES GET OUT
01770   504643          AND (20
01771   740200          SZA             /32 BLOCKS
01772   602003          JMP WRTERM      /NO 1 BLOCK
01773   200633          LAC DIRFLG
01774   504556          AND (17
01775   040632          DAC NDEX
01776   200634          LAC RECORD
01777   504556          AND (17         /DONE 16
02000   540632          SAD NDEX                /YES
02001   741000          SKP
02002   621752          JMP* WRSWS              /NOT REACHED 16
02003   707552  WRTERM  DTRA+10
02004   504600          AND (GOBIT
02005   707544          DTXA            /STOP TAPE
02006   441752          ISZ WRSWS       /STEP EXIT
02007   200633          LAC DIRFLG
02010   740200          SZA
02011   621752          JMP* WRSWS
02012   200634          LAC RECORD      /UPDATE LAST BLK
02013   060342          DAC* LSBLOK     /IF GOING FORWARD
02014   621752          JMP* WRSWS
                        /IF SW 10=1 AND MOTION NOT=0 IGNORE ERRORS
02015   602015  GOCLRD  JMP .
02016   750004          LAS
02017   504644          AND (400
02020   741200          SNA
02021   602025          JMP .+4
02022   707552          DTRA+10
02023   504600          AND (GOBIT
02024   741200          SNA
02025   442015          ISZ GOCLRD
02026   622015          JMP* GOCLRD
                        .EJECT
```

```
02027   707572      WREZTS  DTRB*10
02030   504602              AND (EZBIT      /END ZONE
02031   741200              SNA             /ERROR
02032   602040              JMP WRSERR      /NOT EZ STATUS ERR
02033   200634              LAC RECORD
02034   540264              SAD EZBLOK              /END OF TAPE
02035   602003              JMP WRTERM      /FORWARD
02036   741100              SPA             /END OF TAPE
02037   602003              JMP WRTERM      /BACK WARD
02040   102015      WRSERR  JMS GOCLRD
02041   601760              JMP WRSWS+6
02042   104423              JMS STPTAP
02043   104363              JMS TYWDAT
02044   104153              JMS ERRHLT
02045   601563              JMP WRFIRS
                    /WAIT FOR READ DONE
                    /TEST SWITCH OPTIONS
02046   602046      RDSWS   JMP .
02047   142262              DZM RSEFLG      /CLR ERR FLG
02050   101272              JMS WTFLGS
02051   602105              JMP RDEZTS      /STATUS ERR
02052   200030              LAC WCLOC
02053   740200              SZA             /WC GO TO 0
02054   602116              JMP RDSERR              /NO ERROR
02055   200634              LAC RECORD
02056   060340              DAC* POSITN     /NEW POSITION
02057   342260              TAD BLKINC
02060   040635              DAC TAPONT
02061   750004              LAS
02062   504642              AND (30         /LENTH OF TAPE
02063   741200              SNA             /OR SHORTER
02064   602077              JMP RDSW
02065   504643              AND (20
02066   740200              SZA             /1 BLOCK
02067   602101              JMP RDSW+2      /YES 1 ONLY
02070   200633              LAC DIRFLG
02071   504556              AND (17         /SEE IF DONE
02072   040632              DAC NDEX                /16 BLOCKS
02073   200635              LAC TAPONT
02074   504556              AND (17
02075   540632              SAD NDEX
02076   602101              JMP .+3         /NOT DONE 16
                            .EJECT
```

```
02077   442046   RDSW    ISZ RDSWS               /STEP EXIT
02100   622046           JMP* RDSWS
02101   707552           DTRA+10
02102   504600           AND (GOBIT      /STOP TAPE
02103   707544           DTXA
02104   622046           JMP* RDSWS              /DONE ALL XIT
02105   707572   RDEZTS  DTRB+10
02106   504602           AND (EZBIT
02107   741200           SNA             /EZ INT
02110   602116           JMP RDSERR      /NO STAT ERR
02111   200634           LAC RECORD
02112   540264           SAD EZBLOK              /EOT FWD
02113   602101           JMP RDSW+2      /YES
02114   741100           SPA             /EOT BKWD
02115   602101           JMP RDSW+2      /YES
02116   102015   RDSERR  JMS GOCLRD
02117   602055           JMP RDSWS+7
02120   104423           JMS STPTAP
02121   104340           JMS TYRDAT
02122   104522           JMS WRIDIR
02123   200030           LAC WCLOC
02124   740200           SZA
02125   104330           JMS TYPWC
02126   104447           JMS TYSTAB
02127   777777           LAW -1
02130   042262           DAC RSEFLG      /SET STATUS FLAG
02131   602055           JMP RDSWS+7
02132   602132   RDERCK  JMP .
02133   442262           ISZ RSEFLG      /STATUS ERR
02134   442265           ISZ CDFLAG      /COMPARE ERRO
02135   602140           JMP .+3         /ONE OR OTHER
02136   442132           ISZ RDERCK      /NO ERROR
02137   622132           JMP* RDERCK     /SKIP EXIT
02140   104153           JMS ERRHLT
02141   622132           JMP* RDERCK
                         .EJECT
```

```
02142   602142      CODATA  JMP .
02143   777777              LAW -1
02144   042265              DAC COFLAG
02145   777400              BLENTH          /-256
02146   040632              DAC NDEX
02147   204610              LAC (BUFFER
02150   341161              TAD XTDFLD
02151   042263              DAC GEDATA      /GENERATED ADDRESS
02152   222142              LAC* CODATA
02153   341161              TAD XTDFLD
02154   042264              DAC INDATA      /INPUT ADDRESS
02155   442142              ISZ CODATA
02156   750004              LAS
02157   504644              AND (400                /DELETE ERROR CHECK
02160   740200              SZA
02161   622142              JMP* CODATA
02162   201161              LAC XTDFLD
02163   740200              SZA
02164   707702              EEM
02165   222263              LAC* GEDATA
02166   562264              SAD* INDATA     /SAME DATA
02167   602204              JMP COSAME      /YES
02170   200633              LAC DIRFLG
02171   344605              TAD (1
02172   342261              TAD WRPASD
02173   740200              SZA             /RD OPOSITE WRITE
02174   602204              JMP COSAME      /NO
02175   222263              LAC* GEDATA
02176   740001              CMA
02177   562264              SAD* INDATA     /COMPLEMENTS =
02200   741000              SKP             /YES
02201   102221              JMS COERRO
02202   102212              JMS TSCTRS
02203   602175              JMP .-6
02204   222263      COSAME  LAC* GEDATA
02205   562264              SAD* INDATA     /WORDS =
02206   741000              SKP             /YES
02207   102221              JMS COERRO      /COMPARE ERROR
02210   102212              JMS TSCTRS
02211   602204              JMP COSAME
                            .EJECT
```

```
                        /INCREMENT COMPARE ADDRESS
                        /EXIT IF LAST WORD
02212   000000  TSCTRS  0
02213   442263          ISZ GEDATA
02214   442264          ISZ INDATA
02215   440632          ISZ NDEX
02216   622212          JMP* TSCTRS
02217   707704          LEM
02220   622142          JMP* CODATA
02221   602221  COERRO  JMP .
02222   442265          ISZ COFLAG      /FIRST ERROR
02223   602227          JMP .+4         /NO
02224   104423          JMS STPTAP              /STOP TAPE TYPE HDR
02225   104340          JMS TYRDAT
02226   104522          JMS WRIDIR              /TYP WRITE DIR
02227   750004          LAS
02230   504602          AND (100000             /TYPE ALL ERROR
02231   741200          SNA             /YES
02232   602237          JMP .+5
02233   202265          LAC COFLAG
02234   344645          TAD (777774
02235   740100          SMA             /DONE 4 ERR TYPE
02236   622221          JMP* COERRO     /YES STOP AT 4
02237   104276          JMS TYCRLF              /SPACE BETWEEN PAIRS
02240   222263          LAC* GEDATA
02241   104536          JMS TYDATA              /DATA WRITTEN
02242   104124          JMS TYPTEX
02243   006762          6762            /WR
02244   000077          77              /EOM
02245   202263          LAC GEDATA              /WRITE ADDRS
02246   104547          JMS TYPADR
02247   222264          LAC* INDATA     /DATA READ
02250   104536          JMS TYDATA
02251   104124          JMS TYPTEX
02252   006244          6244            /RD
02253   000077          77              /EOM
02254   202264          LAC INDATA              /AORS DATA READ
02255   104547          JMS TYPADR
02256   622221          JMP* COERRO
02257   000000  RPASFL  0               /COUNT READ PASSES
02260   000000  BLKINC  0
02261   000000  WRPASD  0               /SAVE LAST WRITE DIR
02262   000000  RSEFLG  0               /READ STATUS ERROR FLAG
02263   000000  GEDATA  0
02264   000000  INDATA  0
02265   000000  COFLAG  0
                        .EJECT
```

```
                        /PARITY GENERATION AND CHECKING FIRST PASS
                        /WRITES REV CK SUMS TO 0
                        /(77 GOING BACKWARDS) 2ND PASS
                        /GENERATE PATTERNS 3RD PASS
                        /TEST CHECKSUMS 4 READ BOTH DIRECTIONS
                        /FOR NO PARITY ERRORS
                        /5 GENERATE WRONG PARITY IN BLOCK
                        /6 READ BOTH DIRECTIONS EXPECT PARITY ERRORS
                        /STEP 1  WRITE REVERSE CHECKSUMS TO 0
02266   200265  PARTST  LAC ENDBLK
02267   040634          DAC RECORD
02270   777777          LAW -1
02271   040633          DAC DIRFLG      /BKWDS
02272   100416          JMS SEARCH      /FIND BLOCK 1077
02273   204637          LAC (3000
02274   707544  PARLP1  DTXA
02275   777400          BLENTH          /CHANGE TO READ
02276   040030          DAC WCLOC               /DUMMY READ
02277   204613          LAC (BUFFR2-1   /TO GET TO
02300   040031          DAC CALOC               /REVERSE CHECKSUM
02301   200030          LAC WCLOC
02302   740001          CMA
02303   740200          SZA             /1 WORD LEFT TO READ
02304   602301          JMP .-3
02305   740001          CMA
02306   046400          DAC BUFFER      /ALL 1'S TO WRITE
02307   204603          LAC (BUFFER-1
02310   040031          DAC CALOC               /SET ADDRESS
02311   777776          LAW -2
02312   040030          DAC WCLOC               /2 WORDS
02313   204646          LAC (17000              /SRCH TO WRITE ALL
02314   707544          DTXA            /CHANGE TO WRITE AC2
02315   101272          JMS WTFLGS
02316   602764          JMP WAERR0      /STATUS ERROR
02317   200030          LAC WCLOC
02320   740200          SZA             /WC = 0
02321   602764          JMP WAERR0      /NO STATUS ERROR
02322   777777          LAW -1
02323   340634          TAD RECORD      /BLOCK -1
02324   040634          DAC RECORD
02325   741100          SPA             /DONE 0 YET
02326   602331          JMP .+3         /YES
02327   204646          LAC (17000
02330   602274          JMP PARLP1      /CHANGE BACK TO READ
02331   204600          LAC (GOBIT
02332   707544          DTXA            /STOP TAPE
02333   100303          JMS NEWDRV      /DONE ALL DRIVES
02334   602266          JMP PARTST      /NO DO NEXT
02335   750004          LAS
02336   504567          AND (1000               /END OF PASS HLT
02337   740200          SZA
02340   740040          HLT             /HLT END OF PASS
02341   750004          LAS
02342   504564          AND (40         /REPEAT WRITE SET
0[illegible]   740200          SZA
```

```
02344   602266          JMP PARTST      /STAY IN STEP 1
                        .EJECT
```

```
                    /PARITY TEST STEP 2
                    /WRITE PARITY PATTERNS
                    /FIRST SLIDING BITS THEN
                    /A COUNT PATTERN
02345   140634      PARTS2  DZM RECORD              /START BLOCK 0
02346   140633              DZM DIRFLG
02347   204647              LAC (PARTBL
02350   040010              DAC 10          /FIXED PATTERNS
02351   146400              DZM BUFFER      /FIRST IS 0
02352   100416              JMS SEARCH
02353   204615              LAC (5000
02354   707544      PARWLP  DTXA            /WRITE DATA
02355   204603              LAC (BUFFER-1
02356   040031              DAC CALOC
02357   777777              LAW -1
02360   040030              DAC WCLOC               /1 WORD
02361   101272              JMS WTFLGS
02362   603001              JMP WDERR0      /WDATA STAT ERR
02363   200030              LAC WCLOC
02364   740200              SZA             /WC GO PAST 0
02365   603001              JMP WDERR0      /YES ALSO ERROR
02366   200634              LAC RECORD
02367   060637              DAC* LSTBLK
02370   060340              DAC* POSITN
02371   440634              ISZ RECORD
02372   220010              LAC* 10
02373   046400              DAC BUFFER
02374   750200              SZA!CLA         /DONE ALL FIXED
02375   602354              JMP PARWLP      /NO
02376   707544      PARWL2  DTXA            /WRITE INCR
02377   204603              LAC (BUFFER-1   /PATTERNS
02400   040031              DAC CALOC               /000000 TO
02401   777777              LAW -1          /777700
02402   040030              DAC WCLOC               /1 WORD PER BLOCK
02403   101272              JMS WTFLGS
02404   603005              JMP WDERR1
02405   200634              LAC RECORD
02406   060342              DAC* LSBLOK     /NEW BLOCK
02407   060340              DAC* POSITN     /NEW POSITION
02410   440634              ISZ RECORD
02411   206400              LAC BUFFER      /GET LAST PAT
02412   344650              TAD (10000      /+1 TO UPPER 6
02413   741200              SNA             /GONE 00 TO 77
02414   204651              LAC (10100              /YES STAT 0101 TO 7777
02415   046400              DAC BUFFER
02416   544651              SAD (10100
02417   602422              JMP .+3
02420   504652              AND (7700
02421   741200              SNA             /IN DOUBLE INCREMENT
02422   602430              JMP .+6         /YES
02423   206400              LAC BUFFER
02424   544652              SAD (7700               /DONE TO 777700
02425   602432              JMP .+5         /YES
                            .EJECT
```

```
02426  344626          TAD (100
02427  046400          DAC BUFFER      /UPDATE LWR 6 BITS
02430  750000          CLA
02431  602376          JMP PARWL2
02432  204600          LAC (GOBIT
02433  707544          DTXA            /STOP TAPE
02434  100303          JMS NEWDRV      /DONE ALL DRIVES
02435  602345          JMP PARTS2      /NO
02436  750004          LAS
02437  504567          AND (1000
02440  740200          SZA             /END OF PASS HLT
02441  740040          HLT             /YES
02442  750004          LAS
02443  504564          AND (40         /REPEAT PASS
02444  740200          SZA
02445  602345          JMP PARTS2      /REPEAT GEN PAT
                       .EJECT
```

```
                    /PARITY TEST STEP 3
                    /READ BLOCKS TEST
                    /FOR CORRECT PARITY GENERATED
02446   140634      PARTS3  DZM RECORD
02447   140633              DZM DIRFLG
02450   100416              JMS SEARCH
02451   204637              LAC (3000               /SEARCH TO READ DATA
02452   707544              DTXA
02453   204603              LAC (BUFFER-1
02454   040031              DAC CALOC
02455   777400              BLENTH
02456   040030              DAC WCLOC
02457   200030              LAC WCLOC               /WAIT FOR LAST
02460   740200              SZA                 /DATA WORD
02461   602457              JMP .-2
02462   204567              LAC (1000               /CHANGE TO
02463   707544              DTXA                /READ ALL
02464   101272              JMS WTFLGS
02465   603013              JMP RAERR0
02466   206400              LAC BUFFER
02467   504652              AND (7700               /TEST FOR
02470   741200              SNA                 /1 OR 2 CHAR
02471   602474              JMP .+3             /1 CHAR
02472   770000              LAW 10000               /2 CHAR CKSUM
02473   602476              JMP .+3             /SHOULD = 77
02474   206400              LAC BUFFER          /1 CHAR
02475   244653              XOR (770000         /CHECKSUM = COMP
02476   560031              SAD* CALOC
02477   741000              SKP
02500   603022              JMP PARCKE              /CHECKSUM INCO
02501   200634              LAC RECORD
02502   560342              SAD* LSBLOK         /DONE ALL BLOCKS
02503   602507              JMP .+4             /YES
02504   440634              ISZ RECORD
02505   204567              LAC (1000               /CHANGE BACK
02506   602452              JMP PARTS3*4        /TO READ DATA
02507   204600              LAC (GOBIT
02510   707544              DTXA                /STOP TAPE
02511   100303      PAR3ND  JMS NEWDRV              /DONE 1 ALL DRIVES
02512   602446              JMP PARTS3              /NO DO NEXT DRIVE
02513   750004              LAS
02514   504567              AND (1000               /STOP END OF
02515   740200              SZA                 /PASS UP
02516   740040              HLT                 /YES
02517   750004              LAS
02520   504564              AND (40
02521   740200              SZA
02522   602446              JMP PARTS3          /REPEAT PASS = 1
                            .EJECT
```

```
                        /PARITY TEST STEP 4
                        /READ BLOCKS BOTH DIRECTIONS
                        /SHOULD NOT GET PARITY ERROR
02523   220342  PARTS4  LAC* LSBLOK
02524   040634          DAC RECORD      /START
02525   777777          LAW -1          /BACKWARDS
02526   040633          DAC DIRFLG
02527   100416          JMS SEARCH
02530   204637          LAC (3000               /SEARCH TO
02531   707544          DTXA            /READ DATA
02532   777400          BLENTH
02533   040030          DAC WCLOC
02534   204603          LAC (BUFFER-1
02535   040031          DAC CALOC
02536   101272          JMS WTFLGS
02537   603072          JMP RDPERR      /MAY BE PARITY ERROR
02540   777777          LAW -1          /BLOCK OK
02541   340634          TAD RECORD
02542   040634          DAC RECORD
02543   750100          SMA!CLA         /DONE BLOCK 0
02544   602531          JMP PARTS4+6    /NO
02545   204600          LAC (GOBIT
02546   707544          DTXA            /STOP TAPE
02547   100303          JMS NEWDRV      /DONE ALL DRIVES
02550   602523          JMP PARTS4      /NO DO NEXT
02551   140634  PART4A  DZM RECORD              /BLOCK 0
02552   140633          DZM DIRFLG      /FORWARD
02553   100416          JMS SEARCH
02554   204637          LAC (3000               /READ DATA
02555   707544          DTXA
02556   777400          BLENTH
02557   040030          DAC WCLOC
02560   204603          LAC (BUFFER-1
02561   040031          DAC CALOC
02562   101272          JMS WTFLGS
02563   603106          JMP RDPER1
02564   200634          LAC RECORD
02565   440634          ISZ RECORD
02566   560342          SAD* LSBLOK     /DONE ALL FORWARD
02567   602572          JMP .+3         /YES
02570   750000          CLA
02571   602555          JMP PART4A+4
02572   204600          LAC (GOBIT
02573   707544          DTXA            /STOP TAPE
02574   100303  PAR4ND  JMS NEWDRV
02575   602551          JMP PART4A      /NOT DONE ALL DRIVES
02576   750004          LAS
02577   504567          AND (1000
02600   740200          SZA
02601   740040          HLT             /HALT END OF PASS
02602   750004          LAS
02603   504564          AND (40
02604   740200          SZA
02605   602523          JMP PARTS4      /REPEAT PASS
                        .EJECT
```

```
                        /STEP 5 WRITE BLOCKS TO
                        /WRONG PARITY
02606   140634  PARTS5  DZM RECORD
02607   140633          DZM DIRFLG
02610   100416          JMS SEARCH
02611   204637          LAC (3000
02612   707544          DTXA            /READ DATA
02613   777400          BLENTH
02614   040030          DAC WCLOC
02615   204603          LAC (BUFFER-1
02616   040031          DAC CALOC
02617   200030          LAC WCLOC
02620   740001          CMA             /WAIT FOR
02621   740200          SZA             /NEXT TO LAST WORD
02622   602617          JMP .-3
02623   204646          LAC (17000
02624   707544          DTXA            /CHANGE TO WRITE ALL
02625   777776          LAW -2
02626   040030          DAC WCLOC               /UNLESS CHARACTER
02627   204603          LAC (BUFFER-1   /IS 7777
02630   040031          DAC CALOC               /WRITE CHECKSUM
02631   206400          LAC BUFFER      /TO SAME AS DATA
02632   544654          SAD (777700
02633   146400          DZM BUFFER
02634   101272          JMS WTFLGS
02635   602777          JMP WAERR2      /WA STATUS ERROR
02636   200634          LAC RECORD
02637   440634          ISZ RECORD
02640   560342          SAD* LSBLOK     /DONE ALL BLOCKS
02641   602644          JMP .+3         /YES
02642   204646          LAC (17000      /NO CHANGE BACK
02643   602612          JMP PARTS5+4    /TO READ DATA
02644   204600          LAC (GOBIT
02645   707544          DTXA            /STOP TAPE
02646   100303          JMS NEWDRV
02647   602606          JMP PARTS5
02650   750004          LAS
02651   504567          AND (1000
02652   740200          SZA
02653   740040          HLT             /EWD OF PASS HALT
02654   750004          LAS
02655   504564          AND (40
02656   740200          SZA
02657   602606          JMP PARTS5      /REPEAT PASS
                        .EJECT
```

```
                        /STEP 6 PARITY ERROR TEST
                        /BLOCKS HAVE BEEN WRITTEN TO WRONG PARITY
                        /SHOULD GET PARITY ERRORS READ BOTH DIRECTIONS
02660   140634  PARTS6  DZM RECORD              /START
02661   140633          DZM DIRFLG      /WITH BLOCK 0
02662   100416          JMS SEARCH      /FORWARD
02663   204637          LAC (3000               /READ DATA
02664   707544          DTXA
02665   204603          LAC (BUFFER-1
02666   040031          DAC CALOC
02667   777400          BLENTH
02670   040030          DAC WCLOC
02671   707572          DTRB+10         /WAIT FOR
02672   741200          SNA             /FLAGS
02673   602671          JMP .-2
02674   707562          DTRB
02675   740100          SMA             /SHOULD BE ERROR FLAG
02676   603120          JMP PARERE              /NO ERROR
02677   504600          AND (PARBIT
02700   741200          SNA             /PARITY ERROR
02701   603120          JMP PARERE              /NO
02702   200634          LAC RECORD
02703   440634          ISZ RECORD
02704   560342          SAD* LSBLOK     /DONE ALL FORWARD
02705   602710          JMP .+3         /YES
02706   750000          CLA
02707   602664          JMP PARTS6+4
02710   204600          LAC (GOBIT
02711   707544          DTXA            /STOP TAPE
02712   100303          JMS NEWDRV
02713   602660          JMP PARTS6      /DO NEXT DRIVE
                        .EJECT
```

```
                        /READ BLOCKS BACKWARDS FOR PARITY ERRORS
02714   220342  PART6A  LAC* LSBLOK
02715   040634          DAC RECORD
02716   777777          LAW -1
02717   040633          DAC DIRFLG
02720   100416          JMS SEARCH      /FIND IT BACKWARDS
02721   204637          LAC (3000               /TO READ DATA
02722   707544          DTXA
02723   204603          LAC (BUFFER-1
02724   040031          DAC CALOC
02725   777400          BLENTH
02726   040030          DAC WCLOC
02727   101272          JMS WTFLGS
02730   741000          SKP
02731   603132          JMP PAREE1
02732   707572          DTRB+10
02733   244655          XOR (420100
02734   740200          SZA             /PARITY ERROR AND DTF=1
02735   603132          JMP PAREE1              /NO
02736   777777          LAW -1
02737   340634          TAD RECORD
02740   040634          DAC RECORD
02741   750100          SMA!CLA         /DONE 0 BACKWARDS
02742   602722          JMP PART6A+6    /NO DO NEXT BLOCK
02743   204600          LAC (GOBIT
02744   707544          DTXA            /STOP TAPE
02745   100303  PAR6ND  JMS NEWDRV
02746   602714          JMP PART6A      /DO NEXT DRIVE
02747   750004          LAS
02750   504567          AND (1000
02751   740200          SZA
02752   740040          HLT             /END OF PASS HLT
02753   750004          LAS
02754   504564          AND (40
02755   740200          SZA
02756   602660          JMP PARTS6              /REPEAT PASS
02757   104124          JMS TYPTEX
02760   777777          777777
02761   455644          455644
02762   006077          6077
02763   602266          JMP PARTST
                        .EJECT
```

```
                        /PARITY TEST ERROR TYPEOUTS
                        /WRITE ALL ERROR TYPEOUT
02764   102766  WAERR0  JMS WAERRT
02765   602272          JMP PARTST+4
02766   602766  WAERRT  JMP .
02767   104423          JMS STPTAP
02770   104400          JMS TYWALL
02771   200634          LAC RECORD
02772   104467          JMS TYPBLK
02773   104330          JMS TYPWC
02774   104447          JMS TYSTAB
02775   104153          JMS ERRHLT
02776   622766          JMP* WAERRT
02777   102766  WAERR2  JMS WAERRT
03000   602610          JMP PARTS5+2
                        /WRITE DATA ERROR TYPEOUT
03001   104423  WDERR0  JMS STPTAP
03002   104363          JMS TYWDAT
03003   104153          JMS ERRHLT
03004   602352          JMP PARWLP+2
                        /ALSO A WRITE DATA ERROR TYPEOUT
03005   104423  WDERR1  JMS STPTAP
03006   104363          JMS TYWDAT
03007   104153          JMS ERRHLT
03010   100416          JMS SEARCH
03011   204615          LAC (5000
03012   602376          JMP PARWL2
                        /READ ALL STATUS ERROR TYPEOUT
03013   104423  RAERR0  JMS STPTAP
03014   104353          JMS TYRALL
03015   200634          LAC RECORD
03016   104467          JMS TYPBLK
03017   104447          JMS TYSTAB
03020   104153          JMS ERRHLT
03021   602450          JMP PARTS3+2
                        .EJECT
```

```
                        /TYPE OUT CHECKSUM ERROR
03022   104423  PARCKE  JMS STPTAP
03023   104353          JMS TYRALL
03024   104124          JMS TYPTEX
03025   777777          777777
03026   435300          435300          /CK
03027   636555          636555          /SUM
03030   004562          4562            /ER
03031   627700          627700          /R77
03032   200634          LAC RECORD
03033   104467          JMS TYPBLK
03034   206400          LAC BUFFER
03035   104536          JMS TYDATA
03036   206400          LAC BUFFER
03037   504652          AND (7700
03040   744000          CLL
03041   741200          SNA
03042   744002          STL
03043   770000          LAW 10000
03044   741400          SZL
03045   246400          XOR BUFFER
03046   104536          JMS TYDATA
03047   104124          JMS TYPTEX
03050   004353          4353            /CK
03051   006365          6365            /SU
03052   550045          550045          /M E
03053   706077          706077          /XP EOM
03054   104276          JMS TYCRLF
03055   207000          LAC BUFFR2
03056   104213          JMS TYPCON
03057   104124          JMS TYPTEX
03060   435300          435300          /CK
03061   636555          636555          /SUM
03062   006245          6245            /RE
03063   414477          414477          /AD EOM
03064   104153          JMS ERRHLT
03065   200634          LAC RECORD
03066   440634          ISZ RECORD
03067   560342          SAD* LSBLOK
03070   602511          JMP PAR3ND
03071   602450          JMP PARTS3+2
                        .EJECT
```

```
                        /READ PARITY ERROR BACKWARD
03072   104423  RDPERR  JMS STPTAP
03073   104340          JMS TYRDAT
03074   206777          LAC BUFFR2-1
03075   104536          JMS TYDATA
03076   104447          JMS TYSTAB
03077   104153          JMS ERRHLT
03100   777777          LAW -1
03101   340634          TAD RECORD
03102   741100          SPA
03103   602547          JMP PART4A-2
03104   040634          DAC RECORD
03105   602527          JMP PARTS4+4
                        /READ PARITY ERROR FORWARD
03106   104423  RDPER1  JMS STPTAP
03107   104340          JMS TYRDAT
03110   206400          LAC BUFFER
03111   104536          JMS TYDATA
03112   104447          JMS TYSTAB
03113   200634          LAC RECORD
03114   440634          ISZ RECORD
03115   560342          SAD* LSBLOK
03116   602574          JMP PAR4ND
03117   602553          JMP PART4A+2
                        .EJECT
```

```
                        /TYPE OUT FOR NO PARITY ERROR
                        /PARITY EXPECTED DIRECTION IS FWD
03120   103144  PARERE  JMS PARHDR
03121   206400          LAC BUFFER
03122   104536          JMS TYDATA
03123   104447          JMS TYSTAB
03124   200634          LAC RECORD
03125   104153          JMS ERRHLT
03126   440634          ISZ RECORD
03127   560342          SAD* LSBLOK
03130   602712          JMP PART6A+2
03131   602662          JMP PARTS6+2
                        /NO PARITY ERROR-AS-EXPECTED DIREC BKWD
03132   103144  PAREE1  JMS PARHDR
03133   206777          LAC BUFFR2-1
03134   104536          JMS TYDATA
03135   104447          JMS TYSTAB
03136   104153          JMS ERRHLT
03137   777777          LAW -1
03140   340634          TAD RECORD
03141   741100          SPA
03142   602745          JMP PAR6ND
03143   602720          JMP PART6A+4
03144   603144  PARHDR  JMP .
03145   104423          JMS STPTAP
03146   104340          JMS TYRDAT
03147   104124          JMS TYPTEX
03150   006041          6041            /PA
03151   625164          625164          /RIT
03152   710045          710045          /Y E
03153   626257          626257          /RRO
03154   620045          620045          /R E
03155   706077          706077          /XP EOM
03156   623144          JMP* PARHDR
                        .EJECT
```

```
                        /TABLE OF PARITY FIXED PATTERNS
03157   010000  PARTBL  10000
03160   020000          20000
03161   040000          40000
03162   100000          100000
03163   200000          200000
03164   400000          400000
03165   760000          760000
03166   750000          750000
03167   730000          730000
03170   670000          670000
03171   570000          570000
03172   370000          370000
03173   010100          10100
03174   020200          20200
03175   040400          40400
03176   101000          101000
03177   202000          202000
03200   404000          404000
03201   767600          767600
03202   757500          757500
03203   737300          737300
03204   676700          676700
03205   575700          575700
03206   373700          373700
03207   000000          0                       /END OF TABLE
                        /TC02 BASIC EXERCISER  -  TAPE 3
                        /TC02 BASIC SEARCH ROUTINE 2
                        /READS A SERIES OF 256 BLOCK NUMBERS
                        /INTO MEMORY AND WILL PRINT THEM OR
                        /COMPARE AND PRINT ERRORS
03210   104431  SRCH2R  JMS TYPDRV
03211   104124          JMS TYPTEX
03212   777777          777777
03213   777777          777777
03214   647160          647160          /TYP
03215   450051          450051          /E I
03216   560046          560046          /N F
03217   004657          004657          / FO
03220   620046          620046          /R F
03221   576267          576267          /ORW
03222   416244          416244          /ARD
03223   770000          770000          /77
03224   750000          CLA
03225   103425          JMS TYOTHR              /TYPE OUT OTHER WAIT KEY
03226   544656          SAD (306                /F TYPED IN
03227   744002          STL             /YES AC = 4
03230   204606          LAC (SRCHFW
03231   740400          SNL             /WAS IT FORWARD
03232   344601          TAD (DIRBIT     /NO GO BACKWARD
03233   340631          TAD UNFUNC
03234   707545          DTLA
03235   204610          LAC (BUFFER
03236   040031          DAC CALOC
03237   140030          DZM WCLOC
```

```
03240   707572      SR2WLP  DTRB+10         /WAIT FOR FLAG
03241   741200              SNA
03242   603240              JMP .-2
03243   741100              SPA             /ERROR FLAG UP
03244   603377              JMP SR2EZT              /YES CHECK EZ
03245   204644              LAC (400
03246   540030              SAD WCLOC               /DONE 256 BLOCKS
03247   603253              JMP .+4         /YES
03250   707554              DTXA+10         /RESET ENABLES
03251   440031              ISZ CALOC               /STEP ADDRESS
03252   603240              JMP SR2WLP              /WAIT AGAIN
03253   204600              LAC (GOBIT
03254   707544              DTXA            /STOP TAPE
03255   104124      SR2REQ  JMS TYPTEX
                            .EJECT
```

```
03256   777777              777777          /TYP
03257   777777              777777
03260   647160              647160          /E I
03261   450051              450051          /N C
03262   560043              560043          / FO
03263   004657              4657            /R C
03264   620043              620043          /OMP
03265   575560              575560          /ARE
03266   416245              416245          /EOM
03267   770000              770000
03270   204605              LAC (1
03271   040010              DAC 10
03272   204603              LAC (BUFFER-1)
03273   040011              DAC 11
03274   777777              LAW -1
03275   103425              JMS TYOTHR
03276   040012              DAC 12
03277   104276      SR2FRS  JMS TYCRLF              /TYPEOUT
03300   206400              LAC BUFFER              /FIRS BLOCK #
03301   104213              JMS TYPCON
03302   104124              JMS TYPTEX
03303   465162              465162          /FIR
03304   636477              636477          /ST EOM
03305   200012              LAC 12
03306   544657              SAD (303
03307   603332              JMP SR2COM              /YES COMPARE
03310   200011              LAC 11
03311   504614              AND (7
03312   544614              SAD (7
03313   104276      SR2END  JMS TYCRLF
03314   220011              LAC* 11
03315   104213              JMS TYPCON
03316   200010              LAC 10
03317   440010              ISZ 10
03320   504660              AND (777
03321   540030              SAD WCLOC
03322   741000              SKP
03323   603310              JMP SR2END-3
03324   104124              JMS TYPTEX
03325   544163              544163          /LAS
03326   647700              647700          /T EOM
03327   100303              JMS NEWDRV
03330   603210              JMP SRCH2R
03331   603210              JMP SRCH2R
                            .EJECT
```

```
03332  777777   SR2COM  LAW -1          /SET FIRST ERROR
03333  040012           DAC 12          /FLAG
03334  204610           LAC (BUFFER)
03335  040013           DAC 13
03336  707552           DTRA+10
03337  504601           AND (DIRBIT     /GET DIRECTION
03340  740200           SZA             /FORWARD
03341  777777           LAW -1          /NO BKWDS IS -1
03342  741200           SNA             /BKWD
03343  204605           LAC (1          /NO FWD IS +1
03344  360011           TAD* 11         /LAST BLOCK + OR -
03345  504604           AND (7777               /MASK OVER 4096
03346  560013           SAD* 13         /SEQUENTIAL
03347  603370           JMP SR2CM               /YES NUM OK
03350  440012           ISZ 12          /NEW ERROR
03351  603356           JMP .+5         /NO STILL IN AROW
03352  104276           JMS TYCRLF
03353  104276           JMS TYCRLF
03354  200011           LAC 11          /GET LAST BLOCK NUMBER
03355  344661           TAD (LAC
03356  043357           DAC .+1
03357  200000           LAC
03360  104213           JMS TYPCON              /TYPE FIRST IN ERROR
03361  104276           JMS TYCRLF
03362  200013           LAC 13          /GET CURRENT
03363  344661           TAD (LAC
03364  043365           DAC .+1
03365  200000           LAC
03366  104213           JMS TYPCON
03367  603372           JMP .+3         /SKIP FLAG RESTORE
03370  777777   SR2CM   LAW -1
03371  040012           DAC 12          /RESET FIRST ERROR
03372  440010           ISZ 10
03373  200010           LAC 10
03374  540030           SAD WCLOC               /DONE ALL
03375  603313           JMP SR2END
03376  603336           JMP SR2COM+4
03377  707552   SR2EZT  DTRA+10
03400  504600           AND (GOBIT
03401  740200           SZA             /MOTION STILL = 1
03402  104423           JMS STPTAP              /YES STOP TAPE
03403  104447           JMS TYSTAB
03404  200030           LAC WCLOC
03405  104467           JMS TYPBLK              /NUMBER OF BLOCKS
03406  104124           JMS TYPTEX
03407  630062           630062          /S
03410  454144           454144          /REA
03411  770000           770000          /D EOM
03412  200030           LAC WCLOC
03413  344662           TAD (777776
03414  740100           SMA             /READ 2 BLOCKS
03415  603255           JMP SR2REQ              /MORE THAN 2
03416  544662           SAD (777776     /READ ANY BLOCKS
03417  603210           JMP SRCH2R              /NO, NOT EVEN 1
03420  204605           LAC (1          /IF 1 OR 2
```

```
23421   040010          DAC 10          /THEN TYPEOUT
23422   204663          LAC (LAC BUFFER-1       /BOTH AND
23423   040011          DAC 11          /RESTART
23424   603276          JMP SR2FRS-1
                        .EJECT
```

```
                        /TYPE OUT ALL OTHERS FWD OR COMPARE
                        /WAIT FOR KEYBOARD FLAG
03425   603425  TYOTHR  JMP .
03426   040017          DAC 17
03427   104124          JMS TYPTEX
03430   777777          777777
03431   415454          415454          /ALL
03432   005764          5764            /OT
03433   504562          504562          /HER
03434   630077          630077          /S EOM
03435   440017          ISZ 17
03436   603444          JMP .+6         /TYPE BKWD
03437   104124          JMS TYPTEX
03440   606251          606251          /PRI
03441   566400          566400          /NT EOM
03442   770000          770000
03443   603445          JMP .+2
03444   104416          JMS TYBKW
03445   104272          JMS SPACE2
03446   700301          KSF
03447   603446          JMP .-1
03450   700312          KRB
03451   744000          CLL
03452   623425          JMP* TYOTHR
                        .EJECT
```

```
                        /START STOP TURN AROUND TEST
03453   777777  STRSTP  LAW -1
03454   040017          DAC 17
03455   100373          JMS REWDRV      /BACK TAPE INTO EZ
03456   777777          LAW -1
03457   060340          DAC* POSITN
03460   060341          DAC* DIRECT
03461   100303          JMS NEWDRV
03462   603455          JMP STRSTP+2
03463   440017          ISZ 17          /2ND PASS
03464   603470          JMP .+4         /YES
03465   204561          LAC (1102
03466   540264          SAD EZBLOK      /7 FORMAT TAPES
03467   603455          JMP STRSTP+2    /YES GO BACKWARD AGAIN
03470   104112          JMS WTHALF      /TIME OUT 500 MSEC
                        /START TAPE FORWARD, SHOULD BE FAR ENOUGH
                        /INTO END ZONE TO PICK UP BLOCK 0
                        /FORWARD ON START
03471   140634  STREZF  DZM RECORD      /BLOCK 0
03472   140633          DZM DIRFLG      /FORWARD
03473   103665          JMS STRFWD              /FIRST OUT OF EZ
03474   707554          DTXA+10
03475   204605          LAC (1
03476   040634          DAC RECORD      /NOW GO TO
03477   160340          DZM* POSITN     /BLOCK 1
03500   160341          DZM* DIRECT
03501   101272          JMS WTFLGS
03502   603537          JMP STEZER
03503   206377          LAC BLKFND
03504   440636          ISZ FRSFLG
03505   544605          SAD (1
03506   741000          SKP
03507   603537          JMP STEZER
03510   060340          DAC* POSITN
03511   140634          DZM RECORD
03512   777777          LAW -1          /TURN AROUND
03513   040633          DAC DIRFLG      /SHOULD GET
03514   103711          JMS TABKWD      /0 BKWDS
03515   204664          LAC (1100
03516   540264          SAD EZBLOK
03517   603522          JMP .+3
03520   707552          DTRA+10         /IF 7 FORMAT TAPES
03521   707545          DTLA            /GO BKWD FARTHER
03522   707572          DTRB+10         /WAIT FOR EZ
03523   740100          SMA
03524   603522          JMP .-2
                        .EJECT
```

```
                    /NOW TURN AROUND SHOULD PICK UP BLOCK 0 FORWARD
                    /COMING OUT OF END ZONE
03525   777777              LAW -1
03526   060340              DAC* POSITN
03527   060341              DAC* DIRECT
03530   140634              DZM RECORD
03531   103673              JMS TAFWD               /DO FORWARD TURN AROUND
03532   204600              LAC (GOBIT
03533   707544              DTXA                /STOP TAPE
03534   100303              JMS NEWDRV
03535   603471              JMP STREZF
03536   603544              JMP STFWD1
03537   100563      STEZER  JMS SRERRO
03540   104153              JMS ERRHLT
03541   160340              DZM* POSITN
03542   160341              DZM* DIRECT
03543   603534              JMP STEZER+3
                    /TEST START FORWARD AFTER
                    /FWD STOP IF FAST TEST SWITCH
                    /IS SET MAKE 2 PASSES AND PROCEED
03544   777777      STFWD1  LAW -1              /TO COUNT 2 STARTS
03545   040017              DAC 17              /IF FAST TEST = 1
03546   104112              JMS WTHALF
03547   220340              LAC* POSITN
03550   344120              TAD SFWDK
03551   040634              DAC RECORD
03552   140633              DZM DIRFLG
03553   740001              CMA
03554   340265              TAD ENDBLK
03555   741100              SPA
03556   604044              JMP FEZTST
03557   103665              JMS STRFWD
03560   204600              LAC (GOBIT
03561   707544              DTXA                /STOP TAPE
03562   100303              JMS NEWDRV
03563   603547              JMP STFWD1+3
03564   440017              ISZ 17              /2ND PASS
03565   603572              JMP .+5             /FAST TEST
03566   750004              LAS
03567   504564              AND (40
03570   740200              SZA                 /FAST TEST
03571   603546              JMP STFWD1+2        /YES
                            .EJECT
```

```
                        /TEST START BKWD AFTER STOP FWD
03572   104112  STBKW1  JMS WTHALF
03573   220340          LAC* POSITN     /LAST POSITION
03574   344121          TAD SBKWK               /-3
03575   040634          DAC RECORD      /SHOULD FIND BKWD
03576   777777          LAW -1
03577   040633          DAC DIRFLG
03600   103703          JMS STRBKW              /START TAPE
03601   204600          LAC (GOBIT
03602   707544          DTXA            /STOP TAPE
03603   100303          JMS NEWDRV      /ALL DRIVES
03604   603573          JMP STBKW1+1    /NO
                        /TEST START FWD AFTER BKWD
03605   104112  STFWD2  JMS WTHALF
03606   220340          LAC* POSITN
03607   344122          TAD SFWDK2      /SHOULD FIND
03610   040634          DAC RECORD      /FWD AFTER STP BKWD
03611   140633          DZM DIRFLG
03612   103665          JMS STRFWD              /START TAPE
                        /NOW WAIT FOR 1 MORE BLOCK
                        /AND DO A BKWD TURN AROUND
03613   707554          DTXA+10         /PASS OVER
03614   440634          ISZ RECORD      /1 BLOCK
03615   101272          JMS WTFLGS
03616   603637          JMP SF2ERR      /STATUS ERROR
03617   206377  SF2ERT  LAC BLKFND
03620   440636          ISZ FRSFLG
03621   540634          SAD RECORD
03622   741000          SKP
03623   603637          JMP SF2ERR      /WRONG BLOCK
03624   060340          DAC* POSITN
03625   344562          TAD (777777     /BLOCK -1
03626   040634          DAC RECORD      /FOR TURN AROUND
03627   777777          LAW -1
03630   040633          DAC DIRFLG      /SET BKWDS
03631   103711          JMS TABKWD      /DO TURN AROUND
03632   204600          LAC (GOBIT
03633   707544          DTXA            /STOP TAPE
03634   100303          JMS NEWDRV      /DONE ALL DRIVES
03635   603605          JMP STFWD2      /NO
03636   603646          JMP STBKW2
                        .EJECT
```

```
03637  200634  SF2ERR  LAC RECORD
03640  540264          SAD EZBLOK
03641  604044          JMP FEZTST
03642  102563          JMS SRERRO
03643  104022          JMS TYPAD
03644  100416          JMS SEARCH
03645  603617          JMP SF2ERT
                       /NOW TEST START BKWD AFTER STOP BKWD
03646  104112  STBKW2  JMS WTHALF
03647  220340          LAC* POSITN     /LAST POSITION
03650  344123          TAD SBKWK2      /-5
03651  040634          DAC RECORD      /SHOULD FIND ON
03652  777777          LAW -1          /START BKWD
03653  040633          DAC DIRFLG
03654  103703          JMS STRBKW              /START TAPE
                       /NOW TEST FWD TURN AROUND
03655  440634          ISZ RECORD      /FWD TA
03656  140633          DZM DIRFLG      /SHOULD FIND +1
03657  103673          JMS TAFWD               /TURN AROUND
03660  204600          LAC (GOBIT
03661  707544          DTXA            /STOP TAPE
03662  100303          JMS NEWDRV      /DONE ALL
03663  603646          JMP STBKW2      /NO DO NEXT DRIVE
03664  603544          JMP STFWD1
                       .EJECT
```

```
                        /START TAPE FORWARD
03665   603665          STRFWD  JMP .
03666   204577                  LAC (SRCHFW+ENABLI
03667   340631                  TAD UNFUNC      /+ DRIVE
03670   707545                  DTLA            /LOAD STAT A
03671   103721                  JMS SFTROL
03672   623665                  JMP* STRFWD
                        /TURN AROUND GO FWD
03673   603673          TAFWD   JMP .
03674   707552                  DTRA+10         /IN CASE
03675   740001                  CMA             /GO CLRD
03676   504600                  AND (GOBIT
03677   344601                  TAD (DIRBIT
03700   707544                  DTXA            /COMP DIRECT
03701   103721                  JMS SFTROL
03702   623673                  JMP* TAFWD
                        /START TAPE BACKWARDS
03703   603703          STRBKW  JMP .
03704   204576                  LAC (SRCHBW+ENABLI
03705   340631                  TAD UNFUNC      /+ DRIVE
03706   707545                  DTLA            /LOAD A
03707   103762                  JMS SBTROL
03710   623703                  JMP* STRBKW
                        /DO TURN AROUND BACKWARDS
03711   603711          TABKWD  JMP .
03712   707552                  DTRA+10         /IN CASE GO = 0
03713   740001                  CMA
03714   504600                  AND (GOBIT
03715   344601                  TAD (DIRBIT
03716   707544                  DTXA            /COMP DIRECTION
03717   103762                  JMS SBTROL
03720   623711                  JMP* TABKWD
                                .EJECT
```

```
                        /CONTROL SEARCH FWD ROUTINE
                        /COMPARE FOR CONSECUTIVE BLOCK NUMBERS
                        /ERR IF PAST BLOCK EXIT ON BLOCK NUMBER
                        /
03721   603721  SFTROL  JMP .
03722   777777          LAW -1
03723   040636          DAC FRSFLG      /FIRST BLOCK FLAG
03724   707554          DTXA+10
03725   204603          LAC (BLKFND
03726   040031          DAC CALOC
03727   101272          JMS WTFLGS
03730   603753          JMP SFERRO      /STATUS ERROR
03731   206377          LAC BLKFND
03732   440636          ISZ FRSFLG      /FIRST BLOCK
03733   603737          JMP .+4         /NO
03734   504604          AND (7777               /GET RID OF GARBAGE
03735   040637          DAC LSTBLK      /SAVE BLOCK NUMBER
03736   603745          JMP SFCOMP      /COMPARE
03737   344562          TAD (777777     /THIS BLOCK
03740   540637          SAD LSTBLK      /1 MORE THAN LAST
03741   741000          SKP             /YES
03742   603753          JMP SFERRO      /NOT SEQUENTIAL
03743   344605          TAD (1
03744   040637          DAC LSTBLK
03745   540634  SFCOMP  SAD RECORD      /BLOCK LOOKED FOR
03746   603756          JMP SFERRO+3    /YES
03747   740001          CMA
03750   340634          TAD RECORD
03751   740100          SMA             /PAST BLOCK
03752   603724          JMP SFTROL+3    /NO WAIT NEXT
03753   100563  SFERRO  JMS SRERRO
03754   104022          JMS TYPAD
03755   100416          JMS SEARCH
03756   200634          LAC RECORD
03757   060340          DAC* POSITN     /NEW POSITION
03760   160341          DZM* DIRECT     /AND DIRECTION
03761   623721          JMP* SFTROL
                        .EJECT
```

```
                        /SEARCH BKWD CONTROL
                        /CHECK FOR CONSECUTIVE BLOCK NUMBERS
                        /ERROR IF PAST BLOCK WANTED
03762   603762  SBTROL  JMP .
03763   777777          LAW -1          /SET FIRST
03764   040636          DAC FRSFLG      /BLOCK FLAG
03765   707554          DTXA+10
03766   101272          JMS WTFLGS
03767   604012          JMP SBERRO      /STATUS ERROR
03770   206377          LAC BLKFND
03771   440636          ISZ FRSFLG      /FIRST BLOCK
03772   603776          JMP .+4         /NO
03773   504604          AND (7777
03774   040637          DAC LSTBLK      /SAVE BLOCK NUMBER
03775   604004          JMP SBCOMP
03776   344605          TAD (1          /THIS BLOCK
03777   540637          SAD LSTBLK      /1 LESS THAN LAST
04000   741000          SKP             /YES
04001   604012          JMP SBERRO
04002   344562          TAD (777777
04003   040637          DAC LSTBLK
04004   540634  SBCOMP  SAD RECORD      /EXIT BLOCK
04005   604015          JMP SBERRO+3    /YES
04006   740001          CMA
04007   340634          TAD RECORD
04010   741100          SPA             /PAST BLOCK
04011   603765          JMP SBTROL+3    /NO
04012   100563  SBERRO  JMS SRERRO
04013   104022          JMS TYPAD
04014   100416          JMS SEARCH
04015   200634          LAC RECORD
04016   060340          DAC* POSITN     /NEW POSITION
04017   777777          LAW -1          /AND
04020   060341          DAC* DIRECT     /DIRECTION
04021   623762          JMP* SBTROL
                        .EJECT
```

```
                        /TYPE OUT LAST BLOCK POSITION AND DIRECTION
04022   604022  TYPAD   JMP .
04023   220340          LAC* POSITN
04024   104213          JMS TYPCON
04025   220341          LAC* DIRECT
04026   741200          SNA
04027   604032          JMP .+3
04030   104416          JMS TYBKW
04031   741000          SKP
04032   104411          JMS TYFWD
04033   104124          JMS TYPTEX
04034   205441          5441            / LA
04035   636400          636400          /ST
04036   605763          605763          /POS
04037   777777          777777          /CR LF
04040   777777          777777          /CR LF
04041   770000          770000          /EOM
04042   104153          JMS ERRHLT
04043   624022          JMP* TYPAD
                        .EJECT
```

```
                    /TEST TURN AROUND BOTH
                    /DIRECTIONS ON LAST BLOCK ON TAPE
                    /
04044   777777      FEZTST  LAW -1
04045   040017              DAC 17
04046   040633              DAC DIRFLG
04047   340265              TAD ENDBLK
04050   040634              DAC RECORD
04051   100416              JMS SEARCH      /FIND LAST BLK-1
04052   777777              LAW -1
04053   060341              DAC* DIRECT     /WAS BKWD
04054   200634              LAC RECORD
04055   060340              DAC* POSITN
04056   200265              LAC ENDBLK      /NOW TA
04057   040634              DAC RECORD      /FIND LAST BLOCK
04060   140633              DZM DIRFLG      /FORWARD
04061   103673              JMS TAFWD
04062   707572      FEZWEZ  DTRB+10
04063   740100              SMA             /WAIT FOR EZ
04064   604062              JMP .-2
04065   204664              LAC (1100
04066   540264              SAD EZBLOK      /9 FORMAT
04067   604071              JMP .+2         /YES
04070   440017              ISZ 17          /2ND EZ 7 FORMAT
04071   604075              JMP .+4         /YES
04072   204600              LAC (GOBIT      /HIT EZ
04073   707544              DTXA            /TWICE FOR
04074   604062              JMP FEZWEZ      /7 FORMAT
04075   777777              LAW -1
04076   060340              DAC* POSITN     /INDICATE EZ
04077   040633              DAC DIRFLG      /NOW TA
04100   103711              JMS TABKWD      /FIND LAST BLOCK
04101   204600              LAC (GOBIT      /BACKWARDS
04102   707544              DTXA            /STOP TAPE
04103   100303              JMS NEWDRV      /DONE ALL DRIVES
04104   604044              JMP FEZTST              /NO
04105   104124              JMS TYPTEX              /TYPE END
04106   777777              777777
04107   455644              455644
04110   770000              770000
04111   603453              JMP STRSTP      /RESTART
                            .EJECT
```

```
                        /JUST TIME OUT
                        /HALF A SECOND
04112   604112  WTHALF  JMP .
04113   204665          LAC (272056
04114   040016          DAC 16
04115   440016          ISZ 16
04116   604115          JMP .-1
04117   624112          JMP* WTHALF
04120   000005  SFWDK   5
04121   777775  SBKWK   777775
04122   000003  SFWDK2  3
04123   777773  SBKWK2  777773
                        /LAST TAPE OF TC02 BASIC EXERCISER
                        /TYPE TEXT ROUTINE
                        /MASSAGE PACKED 6 BIT CODES
                        /ASCII -240 STORED AFTER JMS
                        /777777 IS CAR RET LINE FEED
                        /77 CHAR IS EOM
04124   604124  TYPTEX  JMP .
04125   224124          LAC* TYPTEX
04126   444124          ISZ TYPTEX
04127   044206          DAC TYPSAV
04130   740001          CMA
04131   740200          SZA
04132   604135          JMP .+3
04133   104276          JMS TYCRLF
04134   604125          JMP TYPTEX+1
04135   740001          CMA
04136   742020          RTR
04137   742020          RTR
04140   742020          RTR
04141   044207          DAC TYPSAV+1
04142   742020          RTR
04143   742020          RTR
04144   742020          RTR
04145   104163          JMS TYPCHR
04146   204207          LAC TYPSAV+1
04147   104163          JMS TYPCHR
04150   204206          LAC TYPSAV
04151   104163          JMS TYPCHR
04152   604125          JMP TYPTEX+1
                        /STOP ON ERROR IF DELETE SWS NOT SET
04153   604153  ERRHLT  JMP .
04154   750004          LAS
04155   504666          AND (600000
04156   740200          SZA
04157   624153          JMP* ERRHLT
04160   204153          LAC ERRHLT
04161   740040          HLT
04162   624153          JMP* ERRHLT
04163   604163  TYPCHR  JMP .
04164   504667          AND (77
04165   544667          SAD (77
04166   624124          JMP* TYPTEX
04167   344670          TAD (240
```

```
04170   104172              JMS TYPCHA
04171   624163              JMP* TYPCHR
                            .EJECT
```

```
04172   604172      TYPCHA  JMP .
04173   044205              DAC SAVCHA
04174   750004              LAS
04175   741100              SPA
04176   624172              JMP* TYPCHA
04177   204205              LAC SAVCHA
04200   700406              TLS
04201   700401              TSF
04202   604201              JMP .-1
04203   700402              TCF
04204   624172              JMP* TYPCHA
04205   000000      SAVCHA  0
04206   000000      TYPSAV  0
04207   000000              0
04210   000000              0
04211   000000              0
04212   000000              0
                    /TYPE CONTENTS OF THE
                    /AC IN OCTAL
04213   604213      TYPCON  JMP .
04214   104244              JMS DECONT
04215   104265              JMS TYPOCT
04216   204212              LAC TYPSAV*4
04217   104265              JMS TYPOCT
04220   204211              LAC TYPSAV*3
04221   104265              JMS TYPOCT
04222   204210              LAC TYPSAV*2
04223   104265              JMS TYPOCT
04224   204207              LAC TYPSAV*1
04225   104265              JMS TYPOCT
04226   204206              LAC TYPSAV
04227   104265              JMS TYPOCT
04230   104272              JMS SPACE2
04231   624213              JMP* TYPCON
                    /TYPE OUT LOWEST 3 CHAR
                    /IN OCTAL
04232   604232      TYPC03  JMP .
04233   104244              JMS DECONT
04234   204210              LAC TYPSAV*2
04235   104265              JMS TYPOCT
04236   204207              LAC TYPSAV*1
04237   104265              JMS TYPOCT
04240   204206              LAC TYPSAV
04241   104265              JMS TYPOCT
04242   104272              JMS SPACE2
04243   624232              JMP* TYPC03
                            .EJECT
```

```
04244   604244      DECONT  JMP .
04245   244236              DAC TYPSAV
04246   742020              RTR
04247   740020              RAR
04250   244207              DAC TYPSAV+1
04251   742020              RTR
04252   740020              RAR
04253   244210              DAC TYPSAV+2
04254   742020              RTR
04255   740020              RAR
04256   044211              DAC TYPSAV+3
04257   742020              RTR
04260   740020              RAR
04261   044212              DAC TYPSAV+4
04262   742020              RTR
04263   740020              RAR
04264   624244              JMP* DECONT
04265   604265      TYPOCT  JMP .
04266   504614              AND (7
04267   344641              TAD (260
04270   104172              JMS TYPCHA
04271   624265              JMP* TYPOCT
04272   604272      SPACE2  JMP .
04273   104124              JMS TYPTEX
04274   000077              77
04275   624272              JMP* SPACE2
04276   604276      TYCRLF  JMP .
04277   204671              LAC (215
04300   104172              JMS TYPCHA
04301   204672              LAC (212
04302   104172              JMS TYPCHA
04303   624276              JMP* TYCRLF
                    /TYPE OUT HEADERS
                    /DRIVE X
                    /OPER DIR MODE
                    /TYPE OUT MOVE TAPE
04304   604304      TYMOVE  JMP .
04305   104431              JMS TYPDRV
04306   104124              JMS TYPTEX
04307   555766              555766          /MOV
04310   450064              450064          /E T
04311   416045              416045          /APE
04312   770000              770000          /EOM
04313   104501              JMS TYPDIR
04314   104447              JMS TYSTAB
04315   624304              JMP* TYMOVE
                            .EJECT
```

```
                        /TYPE OUT SEARCH
04316   604316  TYSRCH  JMP .
04317   104431          JMS TYPDRV
04320   104124          JMS TYPTEX
04321   634541          634541          /SEA
04322   624350          624350          /RCH
04323   770000          770000          /EOM
04324   104501          JMS TYPDIR
04325   200634          LAC RECORD
04326   104467          JMS TYPBLK
04327   624316          JMP* TYSRCH
                        /TYPE OUT CONTENTS OF WORD COUNT REG
04330   604330  TYPWC   JMP .
04331   104276          JMS TYCRLF
04332   200030          LAC WCLOC
04333   104213          JMS TYPCON
04334   104124          JMS TYPTEX
04335   671643          671643          /W.C
04336   167700          167700          /.EOM
04337   624330          JMP* TYPWC
                        /TYPE OUT READ DATA
04340   604340  TYRDAT  JMP .
04341   104431          JMS TYPDRV
04342   104124          JMS TYPTEX
04343   624541          624541          /REA
04344   440044          440044          /D D
04345   416441          416441          /ATA
04346   770000          770000          /EOM
04347   104501          JMS TYPDIR
04350   200634          LAC RECORD
04351   104467          JMS TYPBLK
04352   624340          JMP* TYRDAT
                        /TYPE OUT READ ALL
04353   604353  TYRALL  JMP .
04354   104431          JMS TYPDRV
04355   104124          JMS TYPTEX
04356   624541          624541          /REA
04357   440041          440041          /D A
04360   545477          545477          /LL EOM
04361   104501          JMS TYPDIR
04362   624353          JMP* TYRALL
                        .EJECT
```

```
                        /TYPE OUT WRITE DATA
04363   604363  TYWDAT  JMP .
04364   104431          JMS TYPDRV
04365   104124          JMS TYPTEX
04366   676251          676251          /WRI
04367   644500          644500          /TE
04370   444164          444164          /DAT
04371   417700          417700          /AEOM
04372   104501          JMS TYPDIR
04373   200634          LAC RECORD
04374   104467          JMS TYPBLK
04375   104330          JMS TYPWC
04376   104447          JMS TYSTAB
04377   624363          JMP* TYWDAT
                        /TYPE OUT WRITE ALL
04400   604400  TYWALL  JMP .
04401   104431          JMS TYPDRV
04402   104124          JMS TYPTEX
04403   676251          676251          /WRI
04404   644500          644500          /TE
04405   415454          415454          /ALL
04406   770000          770000          /EOM
04407   104501          JMS TYPDIR
04410   624400          JMP* TYWALL
                        /TYPE OUT FWD
04411   604411  TYFWD   JMP .
04412   104124          JMS TYPTEX
04413   004667          4667            /FW
04414   447700          447700          /D EOM
04415   624411          JMP* TYFWD
                        /TYPE OUT BKWD
04416   604416  TYBKW   JMP .
04417   104124          JMS TYPTEX
04420   004253          4253            /BK
04421   674477          674477          /WD EOM
04422   624416          JMP* TYBKW
                        /STOP TAPE ON ERROR
                        /LEAVE ALL FLAGS SET
04423   604423  STPTAP  JMP .
04424   707552          DTRA+10
04425   504600          AND (GOBIT
04426   344625          TAD (INHCLR
04427   707544          DTXA
04430   624423          JMP* STPTAP
                        .EJECT
```

```
                        /TYPE OUT DRIVE NUMBER
                        /FIRST LINE OF EVERY TYPEOUT
04431   624431  TYPDRV  JMP .
04432   104124          JMS TYPTEX
04433   777777          777777
04434   777777          777777
04435   446251          446251          /DRI
04436   664500          664500          /VE
04437   770000          770000          /EOM
04440   200641          LAC CDRIVE
04441   741200          SNA
04442   344571          TAD (10
04443   344641          TAD (260
04444   104172          JMS TYPCHA
04445   104276          JMS TYCRLF
04446   624431          JMP* TYPDRV
                        /TYPEOUT DECTAPE STAT B
                        /FOLLOWED BY IORS CONTENTS
04447   604447  TYSTAB  JMP .
04450   104276          JMS TYCRLF
04451   707572          DTRB+10
04452   104213          JMS TYPCON
04453   104124          JMS TYPTEX
04454   636441          636441          /STA
04455   640042          640042          /T B
04456   000077          000077          /EOM
04457   700314          IORS
04460   104213          JMS TYPCON
04461   104124          JMS TYPTEX
04462   515762          515762          /IOR
04463   637700          637700          /S EOM
04464   104276          JMS TYCRLF
04465   104276          JMS TYCRLF
04466   624447          JMP* TYSTAB
                        /TYPE OUT THE AC CONTENTS
                        /AND THE WORD BLOCK
04467   604467  TYPBLK  JMP .
04470   044500          DAC SAVBLK
04471   104276          JMS TYCRLF
04472   204500          LAC SAVBLK
04473   104213          JMS TYPCON
04474   104124          JMS TYPTEX
04475   425457          425457          /BLO
04476   435377          435377          /CK
04477   624467          JMP* TYPBLK
04500   000000  SAVBLK  0
                        .EJECT
```

```
                        /TYPE OUT DIRECTION AND MODE
04501   604501  TYPDIR  JMP .
04502   707552          DTRA+10
04503   504601          AND (DIRBIT
04504   740200          SZA
04505   604510          JMP .+3
04506   104411          JMS TYFWD
04507   741000          SKP
04510   104416          JMS TYBKW
04511   707552          DTRA+10
04512   504650          AND (MODEBT
04513   741200          SNA
04514   624501          JMP* TYPDIR
04515   104124          JMS TYPTEX
04516   004300          4300            /C
04517   555744          555744          /MOD
04520   457700          457700          /E77
04521   624501          JMP* TYPDIR
                        /TYPE OUT WRITE DIRICTION
04522   604522  WRIDIR  JMP .
04523   104124          JMS TYPTEX
04524   006762          6762            / WR
04525   516464          516464          /ITT
04526   455677          455677          /EN EOM
04527   202261          LAC WRPASD
04530   740200          SZA
04531   604534          JMP .+3
04532   104411          JMS TYFWD
04533   624522          JMP* WRIDIR
04534   104416          JMS TYBKW
04535   624522          JMP* WRIDIR
                        /TYPE OUT AC CONTENTS AND THE
                        /WORD DATA AFTER CR LF
04536   604536  TYDATA  JMP .
04537   044500          DAC SAVBLK
04540   104276          JMS TYCRLF
04541   204500          LAC SAVBLK
04542   104213          JMS TYPCON
04543   104124          JMS TYPTEX
04544   444164          444164          /DAT
04545   410077          410077          /A EOM
04546   624536          JMP* TYDATA
                        /TYPE OUT (AC) AND ADRS
04547   604547  TYPADR  JMP .
04550   104213          JMS TYPCON
04551   104124          JMS TYPTEX
04552   414462          414462          /ADR
04553   637700          637700          /S EOM
04554   624547          JMP* TYPADR
        000000          .END
04555   776000  *L
04556   000017  *L
04557   777767  *L
04560   620245  *L
04561   001102  *L
```

```
04562   777777   *L
04563   707552   *L
04564   000040   *L
04565   620002   *L
04566   400000   *L
04567   001000   *L
04570   000343   *L
04571   000010   *L
04572   060000   *L
04573   500000   *L
04574   000200   *L
04575   000002   *L
04576   061400   *L
04577   021400   *L
04600   020000   *L
04601   040000   *L
04602   100000   *L
04603   006377   *L
04604   007777   *L
04605   000001   *L
04606   021000   *L
04607   070000   *L
04610   006400   *L
04611   017777   *L
04612   022000   *L
04613   006777   *L
04614   000007   *L
04615   005000   *L
04616   001117   *L
04617   252525   *L
04620   525252   *L
04621   070707   *L
04622   707070   *L
04623   007000   *L
04624   001327   *L
04625   000300   *L
04626   000100   *L
04627   601365   *L
04630   000037   *L
04631   101425   *L
04632   101523   *L
04633   101411   *L
04634   101526   *L
04635   001410   *L
04636   001406   *L
04637   003000   *L
04640   007377   *L
04641   000260   *L
04642   000030   *L
04643   000020   *L
04644   000400   *L
04645   777774   *L
04646   017000   *L
04647   003157   *L
04650   010000   *L
```

```
04651   210100   *L
04652   007700   *L
04653   770000   *L
04654   777700   *L
04655   420100   *L
04656   000306   *L
04657   000303   *L
04660   000777   *L
04661   200000   *L
04662   777776   *L
04663   206377   *L
04664   001100   *L
04665   272056   *L
04666   600000   *L
04667   000077   *L
04670   000240   *L
04671   000215   *L
04672   000212   *L
            SIZE=04673     NO ERROR LINES
```

```
APISET  01345
BKWDEZ  01257
BLENTH 777400
BLKFND 006377
BLKINC  02262
BLKTIM 775000
BUFFER 006400
BUFFR2 007000
BUFFR3 007400
CALOC  000031
CBIT    00642
CDRIVE  00641
CH0ERR  01425
CH1ERR  01427
CH10ER  01445
CH11ER  01447
CH12ER  01451
CH13ER  01453
CH14ER  01455
CH15ER  01457
CH16ER  01461
CH17ER  01463
CH2ERR  01431
CH20ER  01465
CH21ER  01467
CH22ER  01471
CH23ER  01473
CH24ER  01475
CH25ER  01477
CH26ER  01501
CH27ER  01503
CH3ERR  01433
CH30ER  01505
CH31ER  01507
CH32ER  01511
CH33ER  01513
CH34ER  01515
CH35ER  01517
CH36ER  01521
CH37ER  01523
CH4ERR  01435
CH5ERR  01437
CH6ERR  01441
CH7ERR  01443
CLFAPI  01526
CLFCTR  01542
CLOF   700004
CLON   700044
CLSF   700001
CODATA  02142
COERRO  02221
COFLAG  02265
COSAME  02204
DECONT  04244
DIRBIT 040000
```

```
DIRECT  00341
DIRFLG  00633
DIRTBL  00353
DOTURN  00446
DTAP!   01411
DTCA    707541
DTDF    707601
DTEF    707561
DTFBIT  000100
DTLA    707545
DTRA    707542
DTRB    707562
DTXA    707544
EEM     707702
ENABLI  000400
ENDBLK  00265
ERRHLT  04153
EZBIT   100000
EZBLOK  00264
FEZTST  04044
FEZWEZ  04062
FINDAL  01165
FINDEZ  01212
FRSFLG  00636
GEDATA  02263
GENPAT  01150
GOBIT   020000
GOCLRD  02015
GOTST   00244
GSTPAT  01077
INDATA  02264
INHCLR  000300
KRB     700312
KSF     700301
LEM     707704
LSBLOK  00342
LSTBLK  00637
LSTTBL  00363
MODEBT  010000
MOVBKW  060000
MOVERR  00405
MSBITS  00643
NDEX    00632
NDEX1   00645
NDTTBL  000256
NEWDRV  00303
PARBIT  020000
PARCKE  03022
PAREE1  03132
PARERE  03120
PARHDR  03144
PARLP1  02274
PARTBL  03157
PARTST  02266
PARTS2  02345
```

```
PARTS3  02446
PARTS4  02523
PARTS5  02606
PARTS6  02660
PART4A  02551
PART6A  02714
PARWLP  02354
PARWL2  02376
PAR3ND  02511
PAR4ND  02574
PAR6ND  02745
PATNUM  01160
PATTBL  01117
PAT0    01127
PAT1    01131
PAT2    01133
PAT3    01135
PAT4    01140
PAT5    01142
PAT6    01144
PAT7    01146
PCF    700202
POSITN  00340
POSTBL  00343
PSA    700204
PSB    700244
PSF    700201
PTTEZ   00702
PTTRNA  00646
RAERR0  03013
RCF    700102
RDALL  003000
RDATAF 022000
RDERCK  02132
RDEZLP  00764
RDEZTS  02105
RDFIRS  01636
RDLEND  01672
RDLOOP  01624
RDPERR  03072
RDPER1  03106
RDSERR  02116
RDSW    02077
RDSWS   02046
RDTERM  01714
RECORD  00634
REWDRV  00373
RPASFL  02257
RRB    700112
RSA    700104
RSB    700144
RSEFLG  02262
RSF    700101
RSFDRV  00266
SAVBLK  04500
```

```
SAVCHA  04205
SBCOMP  04004
SBERRO  04012
SBKWK   04121
SBKWK2  04123
SBTROL  03762
SEARCH  00416
SETUNT  00322
SFCOMP  03745
SFERRO  03753
SFTROL  03721
SFWDK   04120
SFWDK2  04122
SF2ERR  03637
SF2ERT  03617
SNOTFS  00544
SPACE2  04272
SRBACK  01226
SRBERR  01247
SRCHBW 061000
SRCHEZ  00464
SRCHFW 021000
SRCHTS  01162
SRCH2R  03210
SRERRO  00563
SRERR1  00560
SREZLP  00713
SREZTA  00756
SRFERR  01202
SR2CM   03370
SR2COM  03332
SR2END  03313
SR2EZT  03377
SR2FRS  03277
SR2REQ  03255
SR2WLP  03240
STBKW1  03572
STBKW2  03646
STEZER  03537
STFWD1  03544
STFWD2  03605
STPTAP  04423
STRBKW  03703
STREZF  03471
STRFWD  03665
STRSTP  03453
SWTBLK  00506
SWTTES  00520
SWTXIT  00537
TABKWD  03711
TAFWD   03673
TAPONT  00635
TCF    700402
TC02TS  00200
TIMCTR  00644
```

```
TLS     700406
TSCTRS  02212
TSF     700401
TSTTBL  00245
TYBKW   04416
TYCRLF  04276
TYDATA  04536
TYFWD   04411
TYMOVE  04304
TYOTHR  03425
TYPAD   04022
TYPADR  04547
TYPBLK  04467
TYPCHA  04172
TYPCHR  04163
TYPCON  04213
TYPCO3  04232
TYPDIR  04501
TYPDRV  04431
TYPOCT  04265
TYPSAV  04206
TYPTEX  04124
TYPWC   04330
TYRALL  04353
TYRDAT  04340
TYSRCH  04316
TYSTAB  04447
TYWALL  04400
TYWDAT  04363
UNFUNC  00631
WAERRT  02766
WAERR0  02764
WAERR2  02777
WAITAP  01365
WCLOC  000030
WDERR0  03001
WDERR1  03005
WDEZLP  01024
WDEZTA  01067
WRALL  005000
WRARD   01543
WREZTS  02027
WRFIRS  01563
WRIDIR  04522
WRPASD  02261
WRSERR  02040
WRSWS   01752
WRTERM  02003
WTAPI   01406
WTFLGS  01272
WTHALF  04112
WTPION  01302
WTTOK   01327
WTTOUT  01313
XTDFLD  01161
```

```
WCLOC  000030
CALOC  000031
DTFBIT 000100
TC02TS  00200
GOTST   00244
TSTTBL  00245
NDTTBL 000256
EZBLOK  00264
ENDBLK  00265
RSFDRV  00266
INHCLR 000300
NEWDRV  00303
SETUNT  00322
POSITN  00340
DIRECT  00341
LSBLOK  00342
POSTBL  00343
DIRTBL  00353
LSTTBL  00363
REWDRV  00373
ENABLI 000400
MOVERR  00405
SEARCH  00416
DOTURN  00446
SRCHEZ  00464
SWTBLK  00506
SWTTES  00520
SWTXIT  00537
SNOTFS  00544
SRERR1  00560
SRERR0  00563
UNFUNC  00631
NDEX    00632
DIRFLG  00633
RECORD  00634
TAPONT  00635
FRSFLG  00636
LSTBLK  00637
CDRIVE  00641
CBIT    00642
MSBITS  00643
TIMCTR  00644
NDEX1   00645
PTTRNA  00646
PTTEZ   00702
SREZLP  00713
SREZTA  00756
RDEZLP  00764
WDEZLP  01024
WDEZTA  01067
GSTPAT  01077
PATTBL  01117
PAT0    01127
PAT1    01131
PAT2    01133
```

```
PAT3    01135
PAT4    01140
PAT5    01142
PAT6    01144
PAT7    01146
GENPAT  01150
PATNUM  01160
XTDFLD  01161
SRCHTS  01162
FINDAL  01165
SRFERR  01202
FINDEZ  01212
SRBACK  01226
SRBERR  01247
BKWDEZ  01257
WTFLGS  01272
WTPION  01302
WTTOUT  01313
WTTOK   01327
APISET  01345
WAITAP  01365
WTAPI   01406
DTAPI   01411
CH0ERR  01425
CH1ERR  01427
CH2ERR  01431
CH3ERR  01433
CH4ERR  01435
CH5ERR  01437
CH6ERR  01441
CH7ERR  01443
CH10ER  01445
CH11ER  01447
CH12ER  01451
CH13ER  01453
CH14ER  01455
CH15ER  01457
CH16ER  01461
CH17ER  01463
CH20ER  01465
CH21ER  01467
CH22ER  01471
CH23ER  01473
CH24ER  01475
CH25ER  01477
CH26ER  01501
CH27ER  01503
CH30ER  01505
CH31ER  01507
CH32ER  01511
CH33ER  01513
CH34ER  01515
CH35ER  01517
CH36ER  01521
CH37ER  01523
```

```
CLFAPI  01526
CLFCTR  01542
WRARD   01543
WRFIRS  01563
RDLOOP  01624
RDFIRS  01636
RDLEND  01672
RDTERM  01714
WRSWS   01752
WRTERM  02003
GOCLRD  02015
WREZTS  02027
WRSERR  02040
RDSWS   02046
RDSW    02077
RDEZTS  02105
RDSERR  02116
RDERCK  02132
CODATA  02142
COSAME  02204
TSCTRS  02212
COERRO  02221
RPASFL  02257
BLKINC  02260
WRPASD  02261
RSEFLG  02262
GEDATA  02263
INDATA  02264
COFLAG  02265
PARTST  02266
PARLP1  02274
PARTS2  02345
PARWLP  02354
PARWL2  02376
PARTS3  02446
PAR3ND  02511
PARTS4  02523
PART4A  02551
PAR4ND  02574
PARTS5  02606
PARTS6  02660
PART6A  02714
PAR6ND  02745
WAERR0  02764
WAERRT  02766
WAERR2  02777
RDALL  003000
WDERR0  03001
WDERR1  03005
RAERR0  03013
PARCKE  03022
RDPERR  03072
RDPER1  03106
PARERE  03120
PAREE1  03132
```

PARHDR 03144
PARTBL 03157
SRCH2R 03210
SR2WLP 03240
SR2REQ 03255
SR2FRS 03277
SR2END 03313
SR2COM 03332
SR2CM 03370
SR2EZT 03377
TYOTHR 03425
STRSTP 03453
STREZF 03471
STEZER 03537
STFWD1 03544
STBKW1 03572
STFWD2 03605
SF2ERT 03617
SF2ERR 03637
STBKW2 03646
STRFWD 03665
TAFWD 03673
STRBKW 03703
TABKWD 03711
SFTROL 03721
SFCOMP 03745
SFERRO 03753
SBTROL 03762
SBCOMP 04004
SBERRO 04012
TYPAD 04022
FEZTST 04044
FEZWEZ 04062
WTHALF 04112
SFWDK 04120
SBKWK 04121
SFWDK2 04122
SBKWK2 04123
TYPTEX 04124
ERRHLT 04153
TYPCHR 04163
TYPCHA 04172
SAVCHA 04205
TYPSAV 04206
TYPCON 04213
TYPC03 04232
DECONT 04244
TYPOCT 04265
SPACE2 04272
TYCRLF 04276
TYMOVE 04304
TYSRCH 04316
TYPWC 04330
TYRDAT 04340
TYRALL 04353

```
TYWDAT  004363
TYWALL  004400
TYFWD   004411
TYBKW   004416
STPTAP  004423
TYPDRV  004431
TYSTAB  004447
TYPBLK  004467
SAVBLK  004500
TYPDIR  004501
WRIDIR  004522
TYDATA  004536
TYPADR  004547
WRALL   005000
BLKFND  006377
BUFFER  006400
BUFFR2  007000
BUFFR3  007400
MODEBT  010000
GOBIT   020000
PARBIT  020000
SRCHFW  021000
RDATAF  022000
DIRBIT  040000
MOVBKW  060000
SRCHBW  061000
EZBIT   100000
CLSF    700001
CLOF    700004
CLON    700044
RSF     700101
RCF     700102
RSA     700104
RRB     700112
RSB     700144
PSF     700201
PCF     700202
PSA     700204
PSB     700244
KSF     700301
KRB     700312
TSF     700401
TCF     700402
TLS     700406
DTCA    707541
DTRA    707542
DTXA    707544
DTLA    707545
DTEF    707561
DTRB    707562
DTDF    707601
EEM     707702
LEM     707704
BLKTIM  775000
BLENTH  777400
```