

The image displays a grid of 10 columns and 10 rows of small data tables. Each table contains various numerical and text-based information, likely representing test results or performance metrics. The data is organized into a structured grid, with each cell containing a small table of its own. The content is too small to read in detail but appears to be a series of related data points or exercises.

A small table located in the bottom right corner of the page, containing a few rows of data.

B01

EOF1DZRPDASB0411

00010000

770224

PDP10 411

E:HDR1DZTMHESEQ

00010000

770224
SEG 0001

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZTMH-E-D
PRODUCT TITLE: TM,A,B-11 MULTIDRIVE DATA RELIABILITY EXERCISER
DATE CREATED: FEB 1977
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: R. B. BARNES/RON PLATUKIS/R. SOLER

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1975, 1977 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

PARAGRAPH	SUBJECT	PAGE
1	ABSTRACT	1
2	REQUIREMENTS	1
3	LOADING PROCEDURE	1
4	STARTING PROCEDURE	2
5	DATA PATTERNS	7
6	RANDOMIZATION	8
7	DYNAMIC PARAMETER	9
8	CONSOLE SWITCHES	10
9	ERROR PRINTOUT	14
10	STATISTIC PRINTOUT	20
11	AUTO SEQUENCE	22
12	TESTING PROCEDURES	24
13	LISTING	

1. ABSTRACT

THIS PROGRAM IS DESIGNED TO BE USED BY AN EXPERIENCED ENGINEER /TECHNICIAN FOR EVALUATION AND DEBUGGING OF MAG TAPE DRIVES. THE PROGRAM IS CAPABLE OF EXERCISING ANY TAPE DRIVE THAT CAN BE OPERATED ON A UNIBUS PDP-11 SYSTEM THROUGH THE TM,A,B-11 MAG TAPE CONTROLLER. ANY TYPE OF TAPE DRIVE; 7 OR 9 TRACK MAY BE USED. ANY NUMBER OF DRIVES, SINGLE OR MULTIDRIVE SYSTEMS, UP TO EIGHT (8), MAY BE TESTED BY A SINGLE EXECUTION OF THE PROGRAM. THIS FLEXIBILITY IS POSSIBLE BECAUSE THE PROGRAM HAS NO FIXED PARAMETERS OR TESTING SEQUENCE. THE ENTIRE TEST PLAN, INCLUDING PARAMETERS AND OPERATING SEQUENCE, IS DETERMINED BY THE OPERATOR THROUGH RESPONSES TO TELETYPE REQUESTS AND SETTING OF CONSOLE SWITCHES.

THE PROGRAM PROVIDES FOR TESTING OF ALL TAPE DRIVE FUNCTIONS SUCH AS WRITING, READING, REWINDING, TAPE POSITIONING, EOT - BOT SENSING AND ASSUMES A GOOD CONTROLLER.

HOWEVER; THE CONTROLLER IS TESTED SOMEWHAT INTRINSICALLY DURING THE TEST CYCLE IN ORDER TO PROVIDE FULL INFORMATION ABOUT ANY ERROR CONDITIONS DETECTED.

DURING A TEST CYCLE, CHECKS ARE MADE FOR STATUS ERRORS, DATA ERRORS, POSITION ERRORS, WORD COUNT AND CURRENT MEMORY ADDRESS ERRORS WHEREVER APPLICABLE.

2. REQUIREMENTS (HARDWARE)

-
- A. ANY PDP-11 PROCESSOR
 - B. 8K OF CORE
 - C. TELETYPE
 - D. TM,A,B-11 TAPE CONTROL UNIT
 - E. 1 TO 8 TSO3 OR TU10,N,W MAG TAPE DRIVES

3. LOADING PROCEDURE

-
- A. USE STANDARD PROCEDURE FOR LOADING BINARY TAPES
 - B. PROGRAM IS LOADABLE AND CHAINABLE IN 8K OF MEMORY.
DEFAULT CHAIN MODE IS A SINGLE PASS ON DRIVE 0
AT 9TRK, 800 BPI, 100 RECORDS OF 200 CHARACTERS EACH,
WITH PATTERN ONE AND ALL SWITCHES 0.

4. STARTING PROCEDURE

THERE ARE FOUR (4) STARTING ADDRESSES THAT MAY BE USED;
200(8), 204(8), 210(8), AND 240(8):

- A. 200(8): THIS ADDRESS MUST BE USED ON INITIAL START FROM LOAD AS ALL PARAMETERS ARE ENTERED FROM HERE. REQUESTS ARE PRINTED ON THE TELETYPE FOR ENTRY OF CONTROLLER REGISTER STARTING ADDRESS, VECTOR ADDRESS, UNIT NUMBER, DENSITY, PARITY, RECORD COUNT, CHARACTER COUNT, PATTERN NUMBER, TAPE MARK (EOF) OPTION, AND STALL FOR READ, WRITE, AND TURNAROUND. ALL RESPONSES SHOULD BE MADE IN OCTAL AND WITHIN THE LIMITS OF THE PARAMETER. A QUESTION MARK (?) WILL BE TYPED IF ANY CHARACTER ENTERED IS NOT BETWEEN 0 THRU 7 (OCTAL). THE CHARACTER MAY BE RETYPED FOLLOWING THE QUESTION MARK. IF THE RESPONSE IS NOT WITHIN ITS LIMITS, A QUESTION MARK (?) IS TYPED AND THE ENTIRE RESPONSE MAY BE REENTERED. SOME RESPONSES REQUIRE MORE THAN ONE (1) CHARACTER, BUT NONE REQUIRES MORE THAN SIX (6). RESPONSES NEED NOT HAVE LEADING ZEROS AND SHOULD BE TERMINATED BY A CARRIAGE RETURN IF LESS THAN THE MAXIMUM NUMBER OF CHARACTERS IS INPUT.
- B. 204(8): THIS ADDRESS SHOULD BE USED ANYTIME A RESTART OF THE PROGRAM IS NECESSARY AND THE PARAMETERS ENTERED AT THE INITIAL START OF 200(8) NEED NOT BE CHANGED. ALSO NOTE THAT ANY DATA PATTERN WHICH HAD BEEN GENERATED BY SETTING THE RANDOM DATA SWITCH (CONSOLE SWITCH EIGHT) WILL NOT BE OVERWRITTEN AND THEREFORE IS HELD IN CORE FOR USE UNTIL CONSOLE SWITCH EIGHT(8) IS AGAIN SET.
- C. 210(8): THIS ADDRESS IS THE SAME AS USING 204(8) IN THAT THE PREVIOUSLY SET PARAMETERS ARE USED; HOWEVER, THE DATA PATTERN IS RETURNED TO THE FIXED PATTERN ORIGINALLY CALLED FOR AT THE 200(8) START. ALSO ALL STATISTICS PREVIOUSLY GATHERED WILL BE CLEARED.
- D. 240(8): THIS IS A SPECIAL ADDRESS WHICH WILL CAUSE THE PROGRAM TO EXECUTE A PREDETERMINED TEST PLAN ON ALL AVAILABLE UNITS. THE ONLY INPUT REQUIRED BY THE OPERATOR IS A RESPONSE TO REQUESTS FOR THE CONTROLLER ADDRESS, VECTOR ADDRESS, AND CONTINUOUS OPERATION OF THE SEQUENCE.

SEE ITEM 11, (PAGE 22) FOR FULL DETAILS.

THE FOLLOWING IS AN EXPLANATION OF THE INITIAL START (200 OCTAL) REQUESTS AND RESPONSES:

REGISTER START: THE RESPONSE REQUIRED FOR THIS REQUEST IS TO ENTER THE ADDRESS OF THE FIRST CONTROLLER REGISTER (MTS) AS A SIX DIGIT UNIBUS ADDRESS.

VECTOR ADDRESS: THE RESPONSE FOR THIS REQUEST IS TO ENTER THE INTERRUPT VECTOR ADDRESS USED BY THE CONTROLLER AS A THREE (3) DIGIT ADDRESS.

UNIT NUMBER: THE UNIT NUMBER IS ENTERED AS ONE (1) OCTAL CHARACTER AND MUST BE WITHIN THE LIMITS OF 0 THROUGH 7. WHEN THE UNIT NUMBER HAS BEEN ENTERED AND IS LEGAL, THE PROGRAM TESTS FOR THE PRESENCE OF A UNIT OF THAT NUMBER. IF THE UNIT IS AVAILABLE A PRINTOUT OF 7 CHANNEL OR 9 CHANNEL WILL BE MADE TO ASSIST THE OPERATOR IN SETTING DENSITY AND PARITY. IF THE UNIT IS NOT AVAILABLE, A MESSAGE STATING SO WILL BE PRINTED AND A NEW UNIT NUMBER REQUEST WILL BE ISSUED. WHEN A GOOD UNIT NUMBER HAS BEEN ENTERED, REQUESTS FOR OPERATING DENSITY AND PARITY ARE MADE FOR THAT UNIT AND SHOULD BE RESPONDED TO ACCORDING TO THAT PARTICULAR UNIT'S NEEDS. AS MANY AS EIGHT (8) UNIT NUMBER REQUESTS MAY BE USED, HOWEVER, AT LEAST ONE MUST BE USED. THE UNIT NUMBER AND THEIR RESPECTIVE DENSITY AND PARITY MAY BE ENTERED IN ANY ORDER. THE INFORMATION FOR EACH UNIT ENTERED IS LOADED INTO A TABLE FOR REFERENCE IN TESTING. IF LESS THAN EIGHT(8) UNITS ARE REQUIRED, THEN RESPONDING TO THE UNIT NUMBER REQUEST WITH A CARRIAGE RETURN WILL TERMINATE THE UNIT ENTRIES AND CONTINUE TO THE NEXT PARAMETER. IT SHOULD BE REMEMBERED THAT AT LEAST ONE UNIT NUMBER REQUEST MUST BE ENTERED. IF THE FIRST REQUEST IS RESPONDED TO BY A CARRIAGE RETURN, THEN THE REQUEST WILL BE REPEATED.

DENSITY: THE DENSITY REQUEST IS RESPONDED TO BY ONE (1) OCTAL CHARACTER AND MUST BE WITHIN THE LIMITS OF 0 THRU 3. AS EACH UNIT NUMBER IS ENTERED, A REQUEST FOR THE OPERATING DENSITY FOR THAT UNIT IS TYPED. THE RESPONSE MEANINGS ARE AS FOLLOWING:

- A. 0 = 200BPI, 7 CHANNEL NRZI
- B. 1 = 556BPI, 7 CHANNEL NRZI
- C. 2 = 800BPI, 7 CHANNEL NRZI
- D. 3 = 800BPI, 9 CHANNEL NRZI

PARITY: THE PARITY REQUEST IS RESPONDED TO BY ONE (1) OCTAL CHARACTER AND MUST BE EITHER 0 OR 1.

A. 1 = EVEN PARITY
B. 0 = ODD PARITY

RECORD COUNT: THIS REQUEST IS RESPONDED TO BY A SIX (6) CHARACTER OCTAL NUMBER FROM 1 TO 177777. REMEMBER LEADING ZEROS ARE NOT REQUIRED AND IF LESS THAN SIX CHARACTERS ARE ENTERED, A CARRIAGE RETURN WILL TERMINATE THE RESPONSE. THE RECORD COUNT IS USED IN CONJUNCTION WITH THE CHARACTER COUNT TO ESTABLISH A BLOCKING FACTOR FOR USE IN READ OR WRITE CYCLES.

CHARACTER COUNT: THIS RESPONSE IS ENTERED AS FOUR (4) OCTAL CHARACTERS WITHIN THE LIMITS OF 4 THRU 4000. AGAIN LEADING ZEROS ARE NOT REQUIRED AND A CARRIAGE RETURN TERMINATES A LESS THAN FOUR (4) CHARACTER RESPONSE. THE CHARACTER COUNT IN CONJUNCTION WITH THE RECORD COUNT IS USED TO ESTABLISH THE BLOCK SIZE (CHARACTERS PER RECORD, AND RECORDS PER BLOCK) USED IN READ AND WRITE CYCLES. THE SAME BLOCKING IS USED ON ALL AVAILABLE UNITS.

PATTERN NUMBER: THIS RESPONSE IS A TWO (2) CHARACTER OCTAL NUMBER WITHIN THE LIMITS OF 0 THRU 20(8). THE NUMBER ENTERED WILL CAUSE A SPECIFIC DATA PATTERN TO BE USED FOR ALL READING AND WRITING. THIS DATA PATTERN IS NOT CHANGED UNLESS RANDOM DATA IS REQUESTED BY SETTING CONSOLE SWITCH EIGHT (8) TO A ONE. RESETTING OF THE RANDOM DATA SWITCH DOES NOT CAUSE REVERSION TO THE FIXED PATTERN, BUT WILL HOLD THE LAST GENERATED PATTERN UNTIL A RESTART IS DONE FROM LOCATION 210(8) OR 200(8). THE SELECTION OF DATA PATTERN ZERO (0) HAS A SPECIAL USE. PATTERN NUMBER ZERO (0) WILL CAUSE TO BE READ IN AT THE HIGH SPEED PAPER TAPE READER ANY DATA PATTERN DESIRED. THE EXTERNAL INPUT DATA THROUGH THE READER IS DONE BY PREPARING A PAPER TAPE WITH A PROGRAM CALLED DTC. (MAINDEC-11-DZTUF) ANY CONFIGURATION OF BITS AND CHARACTERS MAY BE USED AND A LIMIT OF 377(8) CHARACTERS IS IMPOSED. WHEN EXTERNAL DATA IS INPUT, THE ENTIRE WRITE BUFFER IN CORE IS FILLED WITH THE PATTERN SO THAT ANY SIZE RECORD MAY BE USED. DATA PATTERN ZERO (0) EXTERNAL PAPER TAPE NEED ONLY BE READ ONCE AT INITIAL START OF 200(8), AND NEED NOT BE READ AGAIN UNLESS OVERWRITTEN BY RANDOM DATA. BE SURE TO LOAD THE READER BEFORE PRESSING START.

SEE ITEM 5, (PAGE 7) FOR A DESCRIPTION OF THE DATA PATTERNS.

TAPE MARK: THE TAPE MARK REQUEST IS USED TO DETERMINE IF THE OPERATOR WISHES TO HAVE EACH DATA BLOCK SEPARATED BY A TAPE MARK (OFTEN CALLED EOF FOR END OF FILE). IF RESPONDED TO BY A ONE(1) THE TAPE MARK WILL BE WRITTEN AND WHEN READING WILL BE EXPECTED AT THE END OF EACH DATA BLOCK. A ZERO(0) RESPONSE WILL DISALLOW THE TAPE MARK OPTION. PLEASE NOTE THAT THE TAPE MARK RECORD INCREASES THE BLOCK SIZE BY ONE(1) RECORD; IN OTHER WORDS, A BLOCK OF 100 RECORDS WILL HAVE THE TAPE MARK AS RECORD 101.

SINGLE PASS: IF RESPONDED TO WITH A ONE, THE PROGRAM WILL HALT AND PRINT AN END OF PASS MESSAGE WHEN THE LAST AVAILABLE UNIT REACHES END OF TAPE AND IS REWOUND.

STALLS: THE STALL REQUESTS ARE RESPONDED TO BY A SIX (6) CHARACTER OCTAL NUMBER WITHIN THE LIMITS OF 1 THRU 177777. LEADING ZEROS ARE NOT REQUIRED AND AN ENTRY OF LESS THAN SIX (6) CHARACTERS SHOULD BE TERMINATED BY A CARRIAGE RETURN. EACH INCREMENT OF THE VALUE ADDS ABOUT 2.6 MICSEC TO THE DELAY.

READ: THE TIME DELAY BETWEEN EACH RECORD READ
WRITE: THE TIME DELAY BETWEEN EACH RECORD WRITTEN
TURN AROUND: TIME DELAY BETWEEN CHANGES OF TAPE DIRECTION (FORWARD, TO REVERSE, ETC.) AND BETWEEN BLOCKS.

FIXED PARAMETERS: IT SHOULD BE NOTED THAT ALL PARAMETERS EXCEPT FOR THE UNIT DESCRIPTION VALUES (UNIT NUMBER, DENSITY, AND PARITY) HAVE NOMINAL VALUES ALREADY STORED IN THE PROGRAM. AS EACH PARAMETER REQUEST (PATTERN NUMBER, RECORD COUNT, CHARACTER COUNT, AND STALLS) IS TYPED, ITS PRESENT STORED VALUE IS ALSO PRINTED. IF THESE VALUES NEED NOT BE CHANGED, SIMPLY TYPE A CARRIAGE RETURN AS RESPONSE AND NO CHANGE WILL BE MADE. EACH START OF THE PROGRAM AT 200(8) WILL SHOW THE CURRENT VALUES OF THESE PARAMETERS AS PER THE LAST ENTRY. WHEN A FRESH LOAD OF THE PAPER TAPE IS DONE, THE PARAMETERS WILL REFLECT THE FIXED VALUES STORED IN THE PROGRAM.

A. RECORD COUNT = 100
B. CHARACTER COUNT = 200
C. PATTERN NUMBER = 1
D. READ STALL = 1
E. WRITE = 1
F. TURN AROUND = 1

SAMPLE START AT 200(8):

THE FOLLOWING IS A SAMPLE OF THE
PRINTED REQUESTS AND THEIR RESPONSES.
RESPONSES ARE ENCLOSED IN PARENS FOR
CLARITY ONLY AND (CR) MEANS CARRIAGE RETURN

LOAD ADDRESS 200(8), SET CONSOLE SWITCHES, PRESS START SWITCH:

TM, A, B-11:TS03 OR TU10, N, W MULTIDRIVE DATA RELIABILITY EXERCISER
ENTER CONDITIONS IN OCTAL
REGISTER START = 172520 (CR)
VECTOR ADDRESS = 224 (CR)
UNIT NUMBER=(5) 9 TRK
DENSITY=(3)
PARITY=(0)
UNIT NUMBER=(2) 7 TRK
DENSITY=(2)
PARITY=(1)
UNIT NUMBER=(CR)
RECORD COUNT=100 (500)(CR)
CHARACTER COUNT=201 (38)^(7)(CR)
PATTERN NUMBER=1 (22)
?
(6)(CR)
TAPE MARK = 0 (1)(CR)
SINGLE PASS = 0(CR)

ENTER STALLS
READ=1 (CR)
WRITE=1 (CR)
TURN AROUND=1 (3000)(CR)

THE PROGRAM WILL NOW PERFORM THE TEST CYCLE SET IN
THE CONSOLE SWITCHES ON UNIT FIVE (5) THEN TWO (2),
ONE BLOCK ON EACH UNIT PER CYCLE, USING DATA PATTERN
NUMBER SIX (6) WITH A BLOCKING FACTOR OF 37 CHARACTERS
PER RECORD AND 500 RECORDS PER BLOCK. THE DELAYS ARE SET
FOR MINIMUM ON READ AND WRITE, AND APPROXIMATELY .75
SECONDS ON TURN AROUND.

5. DATA PATTERNS

THERE ARE TWENTY DATA PATTERN GENERATORS STORED IN CORE AND ANY ONE OF THESE MAY BE SELECTED. THE ONE UNIQUE CASE IS PATTERN ZERO(0); SELECTION OF PATTERN ZERO(0) REQUIRES THAT A PREVIOUSLY PREPARED PAPER TAPE BE ENTERED AT THE HIGH SPEED READER. THIS TAPE CONTAINS A DATA PATTERN OF NO MORE THAN 377 OCTAL CHARACTERS. THE FIRST CHARACTER READ IN IS THE NUMBER OF ACTUAL DATA CHARACTERS THAT ARE CONTAINED ON THE TAPE. EACH DATA CHARACTER MAY BE ANY COMBINATION OF BITS AND WILL BE LOADED INTO CORE AS THEY APPEAR ON THE TAPE. NO MATTER HOW MANY CHARACTERS ARE ON TAPE, THE ENTIRE WRITE BUFFER (2000 CHARACTERS) WILL BE FILLED WITH THE PATTERN ENTERED SO THAT ANY SIZE RECORD CAN BE USED.

THE FOLLOWING IS A LIST OF THE DATA PATTERNS AVAILABLE:

DATA0: EXTERNAL INPUT THRU HIGH SPEED READER (SEE DTC; MAINDEC-11-DZTUF-A)
DATA1: ALL ONE BITS IN ALL CHARACTERS
DATA2: ALL ZERO BITS IN ALL CHARACTERS
DATA3: A ONE BIT WALKING FROM RIGHT TO LEFT IN A FIELD OF ZEROS
DATA4: A ZERO BIT WALKING FROM RIGHT TO LEFT IN A FIELD OF ONES.
DATA5: ALTERNATING ONE AND ZERO BITS IN EACH CHARACTER
DATA6: ALTERNATING ZERO AND ONE BITS IN EACH CHARACTER
DATA7: SAME AS DATA5 BUT WITH EVERY OTHER CHARACTER COMPLEMENTED
DATA10: SAME AS DATA6 BUT WITH EVERY OTHER CHARACTER COMPLEMENTED
DATA11: INCREMENTING CHARACTERS (000-377)
DATA12: DECREMENTING CHARACTERS (377-000)
DATA13: ALTERNATING CHARACTERS OF ALL ZERO AND ALL ONE BITS
DATA14: ALTERNATING CHARACTERS OF ALL ONE AND ALL ZERO BITS
DATA15: SPECIAL PATTERN OF A WALKING ZERO BIT REPEATED 4 TIMES
DATA16: IBM COMPAT PATTERN 1: RIPPLE
DATA17: IBM COMPAT PATTERN 2: FIXED (ABCDEF)
DATA20: IBM COMPAT PATTERN 3: FIXED (J)

6. RANDOMIZATION

THERE ARE THREE (3) VALUES THAT MAY BE GENERATED RANDOMLY; DATA, CHARACTER COUNT, AND RECORD COUNT. THESE ARE NORMALLY SET TO SOME FIXED VALUE BUT MAY BE RANDOMIZED BY SETTING THE APPROPRIATE CONSOLE SWITCHES.

- A. RANDOM DATA: (CONSOLE SWITCH 8)
GENERATES AN ENTIRE BUFFER, CHARACTER BY CHARACTER, OF RANDOM DATA WHEN SWITCH 8 IS SET TO A ONE. ONCE SET, THE RESETTING OF SWITCH 8 CAUSES THE LAST GENERATED PATTERN TO BE RETAINED IN CORE. A RESTART AT LOCATION 200(8) OR 210(8) WILL CAUSE REVERSION OF THE DATA TO THE FIXED PATTERN REQUESTED INITIALLY. A RESTART AT LOCATION 204(8) WILL HOLD THE LAST GENERATED PATTERN IN CORE UNTIL SWITCH 8 IS AGAIN SET.
ALTHOUGH THE DATA IS GENERATED AS RANDOM, THE PROGRESSION OF RANDOM CHARACTERS IS ALWAYS THE SAME FROM THE OUTSET OF RANDOMIZATION. THEREFORE IT IS POSSIBLE TO GENERATE ONE TAPE REEL OF RANDOM DATA ON ONE UNIT, RESTART THE PROGRAM TO RE-ESTABLISH THE OUTSET POINT, AND READ THE RANDOM TAPE REEL ON ANOTHER UNIT FOR COMPATABILITY TESTING. IN MULTIDRIVE SYSTEMS THE SAME BLOCK OF DATA, WHETHER RANDOM OR FIXED, IS WRITTEN OR READ ON EACH AVAILABLE UNIT IN THE ORDER THAT THEY WERE ENTERED, BEFORE BEING CHANGED.
- B. RANDOM CHARACTER COUNT: (CONSOLE SWITCH 7)
GENERATES A DIFFERENT NUMBER OF CHARACTERS PER RECORD TO BE WRITTEN ON EACH BLOCK CYCLE. THE SAME NUMBER OF CHARACTERS PER RECORD IS WRITTEN OR READ ON EACH AVAILABLE UNIT BEFORE BEING CHANGED. RESETTING SWITCH 7 HOLDS THE LAST VALUE GENERATED.
- C. RANDOM RECORD COUNT: (CONSOLE SWITCH 6)
GENERATES A DIFFERENT NUMBER OF RECORDS FOR EACH BLOCK OF DATA WRITTEN OR READ ON EACH BLOCK CYCLE. THE SAME NUMBER OF RECORDS IS WRITTEN OR READ ON EACH AVAILABLE UNIT BEFORE BEING CHANGED. RESETTING SWITCH 6 HOLDS LAST VALUE GENERATED.

7. DYNAMIC PARAMETERS:

THE THREE (3) STALL VALUES ARE CONSIDERED TO BE DYNAMIC PARAMETERS AS THEY MAY BE CHANGED WHILE THE PROGRAM IS RUNNING BY TYPING A CONTROL C CHARACTER AT THE TELETYPE. AS SOON AS THE BUS IS RELEASED BY THE MAG TAPE OPERATION IN PROGRESS, THE PROGRAM WILL RESPOND TO THE CONTROL C INPUT BY TYPING A REQUEST FOR NEW STALL PARAMETERS. THE LAST VALUES THAT WERE ENTERED WILL BE PRINTED AS THE STORED VALUES AND MAY BE CHANGED BY ENTERING NEW VALUES OR LEFT UNCHANGED BY TYPING A CARRIAGE RETURN.

8. THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

1. <CR> IF NO CHANGES ARE TO BE MADE
2. 6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE ;LAST DIGIT FOLLOWED BY <CR>.
3. ↑U TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.
4. <LF> ONLY VALID FOR ACT-11 SYSTEMS-DO NOT USE

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G (CNTL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

8.1 CONSOLE SWITCH SETTINGS

THE CONSOLE SWITCHES ARE USED TO SET UP THE TEST CYCLE DESIRED, TO GENERATE RANDOM VALUES, AND TO CONTROL ERROR RESPONSES. THE SWITCHES SHOULD BE SET IN THE DESIRED MANNER BEFORE PRESSING THE START SWITCH BECAUSE THEY ARE ALL DYNAMIC AND WILL RUN THE PROGRAM IN ANY CONFIGURATION. ALL SWITCHES SET TO ZERO(0) IS NORMAL.

SW15: 1=STOP ON ERROR
0=CONTINUE ON ERROR

SW14: 1=YOZZLE ON CURRENT BLOCK
0=DO NOT YOZZLE ON BLOCK

SW13: 1=DO NOT CHECK DATA ERRORS
0=CHECK DATA ERRORS

SW12: 1=DO NOT CHECK WRITE STATUS ERRORS
0=CHECK WRITE STATUS ERRORS

SW11: 1=DO NOT CHECK READ STATUS ERRORS
0=CHECK READ STATUS ERRORS

SW10: 1=DO NOT PRINT ANY ERRORS
0=PRINT ALL ERRORS

SW9: 1=REWIND ALL AVAILABLE TAPES
0=DO NOT REWIND

SW8: 1=GENERATE RANDOM DATA
0=USED FIXED DATA

SW7: 1=GENERATE RANDOM CHARACTER COUNT
0=USE FIXED CHARACTER COUNT

SW6: 1=GENERATE RANDOM RECORD COUNT
0=USED FIXED RECORD COUNT

SW5: 1=YOZZLE ON CURRENT RECORD
0=DO NOT YOZZLE ON RECORD

SW4: 1=PRINT STATISTICS
0=DO NOT PRINT STATISTICS

SW3: 1=DO NOT READ
0=READ

SW2: NOT USED

SW1: 1=DISABLE WRITE AND READ RETRY OPTION
0=ENABLE WRITE AND READ RETRY OPTION

SW0: 1=DO NOT WRITE
0=WRITE

SWITCH EXPLANATION AND EXAMPLES:

SWO+SW3: THESE SWITCHES ARE USED TO CONTROL THE SEQUENCE OF MAG TAPE OPERATIONS PERFORMED ON EACH AVAILABLE UNIT. THE BLOCK OF DATA DESCRIBED THROUGH THE RESPONSES TO TELETYPE REQUESTS AT INITIAL START WILL BE EITHER WRITTEN OR READ FROM EACH AVAILABLE UNIT IN THE ORDER THAT THEY WERE ENTERED. THE SEQUENCE OF OPERATIONS IS CALLED A CYCLE, AND WILL BE PERFORMED CONTINUOUSLY UNTIL STOPPED BY THE OPERATOR. WHEN END OF TAPE IS REACHED, THE UNIT WILL BE REWOUND AND FLAGGED AS UNAVAILABLE FOR TEST UNTIL ALL UNITS HAVE REACHED EOT, AT WHICH TIME TESTING IS RESUMED ON ALL AVAILABLE UNITS.

EXAMPLES: SWO+SW3

- A. SWO=0, SW3=1 WRITE ONLY X RECORDS OF Y CHARACTERS
- B. SWO=1, SW3=0 READ ONLY X RECORDS OF Y CHARACTERS
- C. SWO=0, SW3=0 WRITE THEN BACKSPACE AND READ X RECORDS

SW1: SWITCH ONE(1), WHEN SET TO A ZERO (0), WILL CAUSE ANY DATA RELATED WRITE ERROR TO BE RETRIED. THE RETRY SCHEME CONSISTS OF REWRITING THE RECORD IN THE SAME SPOT ON THE TAPE FOUR (4) TIMES. IF ALL FOUR (4) REPEATS ARE SUCCESSFUL, THE RECORD IS CONSIDERED RECOVERED, AND A TAPE WRITE ERROR IS LOGGED. IF ANY OF THE FOUR (4) REPEATS IS UNSUCCESSFUL, A WRITE WITH EXTENDED INTERCORD GAP IS DONE, A SUSPECTED BAD TAPE SPOT LOGGED AT THIS CLOCK AND RECORD NUMBER, AND A SECOND RETRY OF FOUR REPEATS IS DONE. IF AFTER FOUR (4) RETRIES, THE RECORD CANNOT BE RECOVERED A NOTIFICATION IS PRINTED, AND TESTING IS RESUMED ON THE NEXT RECORD. IF 20(8) BAD TAPE SPOTS ARE FOUND, THE UNIT WILL BE REWOUND AND REMOVED FROM TESTING WITH AN APPROPRIATE MESSAGE PRINTED.

SWITCH ONE (1), WHEN SET TO A ZERO (0), WILL ALSO CAUSE ANY DATA RELATED READ ERROR TO BE RETRIED. THE RETRY SCHEME CONSISTS OF REREADING THE RECORD A MAXIMUM OF FOUR (4) TIMES. IF THE RECORD IS SUCCESSFULLY RECOVERED ON ANY OF THE REREADS IT IS CONSIDERED FOR STATISTICS PURPOSES TO BE A SOFT READ ERROR AND TESTING CONTINUES IF THE REREADS FAIL TO RECOVER THE RECORD, THE ERROR IS LOGGED AS A HARD READ ERROR.

SW4: SWITCH FOUR (4) WHEN SET WILL PRINT THE STATISTICS GATHERED FOR EACH UNIT. THE NUMBER WILL BE PRINTED AT THE END OF A BLOCK CYCLE.

SEE ITEM 10, PAGE 20 FOR FULL DETAILS.

- SW5: SWITCH FIVE (5) WHEN SET DURING A READ OPERATION WILL CAUSE THE PROGRAM TO CONTINUOUSLY READ THE CURRENT RECORD BY SPACING REVERSE OVER THE RECORD AND REREADING THAT RECORD. THIS TAPE MOVEMENT IS CALLED YOZZLING. THERE IS A SOFTWARE DELAY EXECUTED BETWEEN EACH SPACE/READ OF THE RECORD AND IT MAY BE VARIED BY TYPING CONTROL C ON THE TELETYPE DURING THE EXECUTION OF THE YOZZLE AND RESPONDING TO THE PRINTED REQUEST WITH A SIX (6) DIGIT VALUE. THE YOZZLE STALL IS PRESET TO A VALUE OF 1000 IN THE PROGRAM TO PREVENT EXCESSIVE TAPE WEAR, BUT MAY BE SET TO ANY VALUE THROUGH THE TELETYPE.
- SW6-8: THESE THREE (3) SWITCHES CONTROL THE RANDOMIZATION OF DATA AND BLOCK SIZE AND MAY BE SET AND RESET AT ANY TIME. THE ACTUAL CHANGE WILL TAKE PLACE BETWEEN BLOCK CYCLES.
- SW9: SWITCH NINE (9) WHEN SET WILL CAUSE ALL AVAILABLE TAPE UNITS TO BE REWOUND AT THE END OF THE CURRENT BLOCK CYCLE. TESTING WILL BE RESUMED AT A BLOCK COUNT OF ONE (1) WHEN ALL UNITS HAVE REACHED BOT.
- SW10-13: THESE SWITCHES ARE USED TO CONTROL THE ERROR HANDLING TO BE DONE ON THE TAPE OPERATION DESCRIBED BY SWITCHES 0+3.
- A. SWITCH TEN (10) WHEN SET TO A ONE WILL DISALLOW ANY ERROR PRINTOUTS MADE ON THE OPERATION IN PROGRESS. CATASTROPHIC FAILURES AND INFORMATION PRINTOUTS WILL STILL OCCUR. IE: UNIT NOT AVAILABLE, ILLEGAL BOT, DROP OR PICK OVERFLOW, AND EOT REWIND.
 - B. SWITCH ELEVEN (11) WHEN SET TO A ONE WILL DISALLOW THE CHECKING FOR STATUS ERRORS ON READ OPERATIONS.
 - C. SWITCH TWELVE (12) WHEN SET TO A ONE WILL DISALLOW THE CHECKING FOR STATUS ERRORS ON WRITE OPERATIONS.
 - D. SWITCH THIRTEEN (13) WHEN SET TO A ONE WILL DISALLOW THE CHECKING OF READ DATA. THIS SWITCH HAS NO EFFECT ON STATUS CHECKING.

SW14: SWITCH FOURTEEN (14) IS USED DURING A READ ONLY OPERATION; WHEN SET, THE BLOCK OF DATA BEING READ WILL CONTINUOUSLY BE READ AND SPACED OVER SO THAT TAPE WILL REMAIN AT THE SAME BLOCK. WHEN RESET, THE TAPE WILL BE ALLOWED TO MOVE FORWARD AND DATA BLOCKS WILL BE READ PROGRESSIVELY. THIS IS A BLOCK YOZZLE.

SW15: SWITCH FIFTEEN (15) WHEN SET TO A ONE, WILL CAUSE THE PROGRAM TO HALT ON ANY ERROR DETECTED BY THE OPERATION IN PROGRESS. IF BOTH SWITCH TEN (10) AND FIFTEEN (15) ARE SET, THE ACTUAL ERROR DETECTED WILL NOT BE PRINTED BUT WILL CAUSE A HALT. IF SWITCH TEN (10) IS RESET BEFORE PRESSING CONTINUE, THE ERROR WHICH CAUSED THE HALT WILL BE PRINTED BEFORE TESTING IS RESUMED.

9. ERROR PRINTOUTS

THERE ARE THREE TYPES OF ERROR PRINTOUTS MADE BY THE PROGRAM: OPERATION ERRORS, DATA ERRORS, AND CONDITION ERRORS. EACH ERROR MESSAGE PRINTED IS PRECEDED BY A HEADER WHICH CONTAINS THE UNIT NUMBER, BLOCK COUNT NUMBER, BAD RECORD NUMBER PLUS TOTAL NUMBER OF RECORDS, SIZE OF RECORD, AND TYPE OF OPERATION WHICH CAUSED ERROR.

A. OPERATION ERRORS:

THESE ARE ERRORS WHICH CAN OCCUR AS A DIRECT RESULT OF A TAPE OPERATION.

1. READ/WRITE STATUS ERRORS: THESE ARE INDICATED BY THE ERROR BIT (BIT 15) OF THE TAPE COMMAND REGISTER BEING SET TO A ONE.
2. RECORD LENGTH ERRORS: THESE ARE INDICATED BY A BYTE COUNT OTHER THAN ZERO (0) OR AN INCORRECT CURRENT MEMORY ADDRESS OR BOTH
3. TAPE POSITIONING ERRORS: THESE ARE INDICATED BY A SPACE COUNT OTHER THAN ZERO (0), NO BOT FOUND FROM A REWIND, OR NO TAPE UNIT READY AT THE END OF REWIND.

B. DATA ERRORS:

DATA ERRORS WILL OCCUR WHEN TAPE IS BEING READ AND THE DATA DOES NOT MATCH THE EXPECTED DATA.

BECAUSE DATA RECORDS CAN BE UP TO TWO THOUSAND CHARACTERS LONG, AN ERROR CONDITION WHICH WILL CAUSE THE ENTIRE RECORD TO READ INCORRECTLY COULD CAUSE A VERY LENGTHY PRINTOUT. THEREFORE, A COUNTER OF SUCCESSIVE BAD CHARACTERS IS EMPLOYED. IF TEN (10) CHARACTERS IN SUCCESSION ARE BAD, A NOTIFICATION IS PRINTED (BAD RECORD) AND THE NEXT TWENTY (20) CHARACTERS ARE SKIPPED BEFORE CHECKING IS RESUMED. IF THE BAD RECORD CONDITION OCCURS THREE (3) TIMES IN ONE RECORD, THE REST OF THE RECORD IS SKIPPED, DOWN TO THE LAST TEN (10) CHARACTERS, WHICH WILL BE CHECKED. THE SKIPPING AND RESUMPTION OF CHECKING WILL ONLY BE DONE ON RECORDS WHICH ARE LONG ENOUGH TO ALLOW IT.

C. CONDITION ERRORS: THESE ERRORS REFLECT THE STATE OF THE TAPE SYSTEM BEFORE AND AFTER AN OPERATION.

1. EOT: WHEN AN EOT (END OF TAPE) IS ENCOUNTERED DURING EITHER A READ OR A WRITE, THAT UNIT IS FLAGGED AS UNAVAILABLE FOR TESTING AND IS REWOUND UNTIL ALL AVAILABLE UNITS HAVE REACHED EOT. AT WHICH TIME TESTING IS RESUMED IN ALL AVAILABLE UNITS.
2. ILLEGAL BOT: WHEN A UNIT ENCOUNTERS BEGINNING OF TAPE (BOT) DURING A READ OPERATION THE ERROR IS PRINTED AND THE UNIT DROPPED FROM TESTING UNTIL ALL ARE RESTARTED ON THE NEXT PASS.
3. DROP DRIVE: UNIT BECOMES UNAVAILABLE DUE TO LOSE OF SELECT REMOTE, BOT DURING REWIND, OR NO TUR WHEN MAKING INITIAL SELECTION UNIT IS DROPPED, STATISTICS PRINTED, TESTING WILL RESUME AT BEGINNING OF NEXT PASS.
4. CONTROLLER NOT READY: BEFORE ANY OPERATION IS ATTEMPTED THE CONTROLLER IS CHECKED FOR READY. IF IT IS NOT READY, AN ERROR WILL BE PRINTED AND THE PROGRAM WILL STOP.
5. NO INTERRUPT RETURNED: EACH TAPE OPERATION SHOULD BE TERMINATED BY SETTING AN INTERRUPT IN THE CPU. IF NO INTERRUPT IS RETURNED WITHIN THE APPROPRIATE TIME, AN ERROR IS PRINTED.
6. NO MORE UNITS TO TEST: IF ALL UNITS HAVE BEEN DROPPED FOR CATASTROPHIC ERRORS, THE PROGRAM WILL STOP.

E. EXAMPLES:

GLOSSARY:

- BN = BLOCK NUMBER
- RN = RECORD NUMBER (X) OF A TOTAL OF (Y)
- RS = RECORD SIZE IN CHARACTERS PER RECORD
- WE = WRITE ERROR
- RE = READ ERROR
- SE = SPACE ERROR
- F = FORWARD
- CR = COMMAND REGISTER
- CS = STATUS REGISTER
- WC = BYTE COUNTER
- CA = CURRENT MEMORY ADDRESS POINTER AND EXPECTED VALUE
- CN = CHARACTER NUMBER
- G = GOOD DATA (SHOWN IN BIT FORMAT AS IN CORE)
- B = BAD DATA (SHOWN IN BIT FORMAT AS IN CORE)
- ERR AMT = NUMBER LEFT TO SPACE
- TM = TAPE MARK (OFTEN CALLED EOF FOR END OF FILE)
- LPC = LONGITUDINAL PARITY CHECK (RECEIVED - EXPECTED)
- PATRN = DATA PATTERN (R=RANDOM)

EXAMPLE 1

EXAMPLE 1: IN THIS EXAMPLE A TAPE VERTICAL PARITY ERROR WAS DETECTED DURING A WRITE OPERATION OF THE TWELVTH (12) RECORD OF THE BLOCK. THE WORD COUNT AND CURRENT MEMORY ADDRESS ARE CORRECT. THE RETRY OPTION WAS DISABLED.

UNIT NO. 3 *DEN 1 *PAR 0 *PATTRN 1
 BN 406*RN 12-200*RS 2000*WE
 COMD 1010001111000100
 STAT 0001000001000001
 WC 0
 CA 14436-14436

EXAMPLE 2

EXAMPLE 2: IN THIS EXAMPLE A RECORD LENGTH ERROR WAS DETECTED WHILE READING THE FIRST RECORD OF THE BLOCK. THE RETRY OPTION WAS DISABLED. THE WORD COUNT SHOWS A COUNT OF 20 CHARACTERS LEFT TO BE TRANSFERRED. THE CURRENT MEMORY ADDRESS REFLECTS THAT A SHORTAGE OF 20 CHARACTERS TRANSFERRED HAD OCCURRED. IN THIS EXAMPLE THE STATUS AND COMMAND REGISTERS DO NOT SHOW ANY ERROR, BUT THE LPC IS SHOWN TO BE INCORRECT.

UNIT NO. 7 *DEN 2 *PAR 0 *PATTRN 6
 BN 10*RN 1-100*RS 50*RE F***
 COMD 0100011111000100
 STAT 0000000001000001
 WC 20
 CA 12466-12506
 LPC 337 -147

EXAMPLE 3

EXAMPLE 3: IN THIS EXAMPLE THE TAPE UNIT WAS TRYING TO SPACE OVER THE 15 RECORDS IN THE BLOCK IN ORDER TO ESTABLISH PROPER POSITION TO BEGIN READING. THE OPERATION WAS TERMINATED BEFORE THE ENTIRE 15 RECORDS WERE TRAVERSED AND AN ERROR SHOWN BECAUSE THE TAPE IS NOT IN PROPER POSITION TO BEGIN READING.

UNIT NO. 0 *PATTRN R
 BN 2*RN 15-15*RS 23 *SE
 ERR AMT 4

EXAMPLE 4

EXAMPLE 4: IN THIS EXAMPLE UNIT NUMBER ONE (1) HAD BEEN REWOUND VIA CONSOLE SWITCH NINE (9) AND AT THE COMPLETION OF THE OPERATION BOT WAS NOT SET IN THE STATUS REGISTER.

UNIT NO. 1 *DEN 3 *PAR 0 *PATRN R
BN 3002*RN 65-65*RS 10
NO BOT ON REWIND-HALT

EXAMPLE 5

EXAMPLE 5: IN THIS EXAMPLE TWO BAD CHARACTERS WERE READ FROM TAPE IN THE FORWARD DIRECTION. THE FIRST (0) AND THE THIRTEENTH (13) CHARACTERS OF THE TOTAL NUMBER OF SIXTEEN (16) CHARACTERS IN THE BLOCK ARE BAD. CHARACTER NUMBER ZERO (0) HAS DROPPED BIT NUMBER FIVE (5) AND CHARACTER NUMBER TWELVE (12) HAS PICKED UP BIT NUMBER SEVEN (7).

UNIT NO. 5 *DEN 3 *PAR 0 *PATRN S
BN 12*RN 3-10*RS 15*DE-F**
CN 0
G: 10101010
B: 10001010
CN 12
G: 01010101
B: 11010101

EXAMPLE 6

EXAMPLE 6: IN THIS EXAMPLE UNIT NUMBER SIX (6) HAS REACHED END OF TAPE (EOT) FOR THE 1ST TIME AND WILL BE REWOUND. TESTING WILL RESTART ON UNIT NUMBER SIX (6) WHEN ALL UNITS HAVE REACHED EOT.

UNIT NO. 6 *DEN 3 *PAR 0 *PATRN R
BN 677 *RN 25-600*RS 1566
EOT NO. 1
UNIT WILL REWIND AND BE
RESTARTED ON BLOCK ONE
WHEN ALL AVAIL UNITS REACH EOT

EXAMPLE 7

EXAMPLE 7: IN THIS EXAMPLE UNIT NUMBER TWO (2) HAS ENCOUNTERED BEGINNING OF TAPE (BOT). DRIVE WILL BE DROPPED STATISTICS WILL BE PRINTED, TESTING RESUMED AT BEGINNING OF NEXT PASS.

UNIT NO. 2 *DEN 2 *PAR 0 *PATRN 2
BN 56*RN 2-4*RS 1200
ILLEGAL BOT

EXAMPLE 8

EXAMPLE 8: IN THIS EXAMPLE THE SELECTED UNIT (NUMBER 0) HAS BECOME UNAVAILABLE. UNIT WILL BE DROPPED STATISTICS WILL BE PRINTED, TESTING WILL RESUME AT BEGINNING OF NEXT PASS.

UNIT NO. 3 *DEN 1 *PAR 0 *PATRN 4
BN 1*RN 0-200*RS 66 NOT AVAIL
(OR LOST SELECT REMOTE, NO BOT ON REWIND)

EXAMPLE 9

EXAMPLE 9: IN THIS EXAMPLE THE WRITE OPERATION EXECUTED ON UNIT NUMBER SIX (6) WAS NOT COMPLETED AND NO INTERRUPT WAS RETURNED.

UNIT NO. 6 *DEN 2 *PAR 0 *PATRN R
BN 12*RN 3-4*RS 100*WE
NO INTERRUPT RETURNED

(PAGE 19)

SEG 0021

EXAMPLE 10

EXAMPLE: 10 THIS EXAMPLE SHOWS A READ ERROR WHICH RECOVERED ON THE SECOND RETRY. THIS ERROR WILL BE LOGGED AS A RDERR BUT WILL BE CATEGORIZED AS A SOFT ERROR. THE REGISTERS SHOW A PARITY ERROR WAS THE CAUSE OF THE ERROR.

```
UNIT NO. 1 *DEN 3 *PAR 1 *PATRN R
*BN 10 *RN 2-100 *RS 1117 *RE F***
CMD 1110100110000010
STAT 0011000001000001
WC 0
LPC 337-147
***ORIGINAL ERROR***
```

```
UNIT NO. 1 *DEN 3 *PAR 0 *PATRN R
*BN 10 *RN 2-100 *RS 1117 *RE F***
CMD 1110100110000010
STAT 0011000001000001
WC 0
LPC 337-147
READ FAILED--RETRY: 1
REREAD SUCCESSFUL--RETRY: 2
```

EXAMPLE 11

EXAMPLE 11: THIS EXAMPLE SHOWS A WRITE ERROR WHICH WAS NOT RECOVERED BY SUCCESSFULLY REWRITTING THE RECORD FOUR TIMES AT THAT LOCATION. THE RECORD WAS SUCCESSFULLY WRITTEN AFTER 3 INCHES OF TAPE WAS ERASED. THIS ERROR WILL BE LOGGED AS A BAD TAPE SPOT.

```
UNIT NO. 0 *DEN 3 *PAR 0 *PATRN R
*BN 2 *RN 370 -461 *RS 2407 *WE
CMD 1110000010000100
STAT 0011000001000001
WC 0
CA 25613 -25613
***ORIGINAL ERROR***
```

```
UNIT NO. 0 *DEN 3 *PAR 0 *PATRN R
*BN 2 *RN 370 -461 *RS 2407 *WE
CMD 1110000010000100
STAT 0011000001000001
WC 0
CA 25613 -25613
SUSPECT BAD TAPE
RETRY: 0
REPEAT: 0
RECOVERED
RETRY: 1
```

10. STATISTICS PRINTOUT

THE PROGRAM GATHERS A VARIETY OF STATISTICS DURING THE COURSE OF ITS TESTING. THE STATISTICS ARE KEPT ON A UNIT BY UNIT BASIS AND ARE SUMMARIZED IN A STATISTICS PRINTOUT. STATISTIC PRINTOUTS CAN BE PRINTED AT THE END OF EACH BLOCK CYCLE BY SETTING SWITCH FOUR (4) TO 1. A STATISTIC PRINTOUT IS AUTOMATICALLY PRINTED WHEN A UNIT REACHES EOT AND IS REWOUND.

HERE IS AN EXPLANATION OF THE STATISTIC SUMMARY.

DROPS: THE NUMBER OF BITS DROPPED ON A PER TRACK BASIS. DROPS ARE COLLECTED DURING THE DATA CHECK ROUTINE.

PICKS: THE NUMBER OF BITS PICKED ON A PER TRACK BASIS. DROPS ARE COLLECTED DURING THE DATA CHECK ROUTINE.

WTERR: THE NUMBER OF RECORDS IN WHICH A WRITE ERROR OCCURRED. IF WRITE RETRY WAS ENABLED, WTERR WILL CONTAIN ONLY THOSE RECORDS WHICH WERE NOT RECOVERED AFTER ONE RETRY.

RTRY: THE NUMBER OF RETRIES INITIATED UNDER THE WRITE RETRY OPTION. (SEE ITEM 8., SW1:)

RDERR: THE TOTAL NUMBER OF RECORDS IN WHICH A READ ERROR OCCURRED.

SOFT: THE NUMBER OF READ ERRORS WHICH WERE RECOVERED WITHIN A MAXIMUM OF FOUR REREADS OF A RECORD UNDER THE READ RETRY OPTION. (SEE ITEM 8., SW1:)
**NOTE: SOFT READ ERRORS ARE ONLY CATEGORIZED FOR THOSE READ ERRORS OCCURRING WHEN CONSOLE SWITCH 1 IS SET TO ZERO.

HARD: THE NUMBER OF READ ERRORS WHICH REMAINED UNRECOVERED UNDER THE READ RETRY SCHEME. (SEE ITEM 8., SW1:)
**NOTE: HARD READ ERRORS ARE ONLY CATEGORIZED FOR THOSE READ ERRORS OCCURRING WHEN CONSOLE SWITCH 1 IS SET TO ZERO.

DTERR: THE NUMBER OF DATA ERRORS FOUND FOR THIS UNIT.
**NOTE: DATA ERRORS ARE ONLY FOUND FOR THOSE RECORDS WHICH WERE READ WITH SWITCH 11 RESET TO ZERO.

BAD TAPE SPOTS: A COUNT OF THE NUMBER OF TAPE SPOTS
WHERE A RECORD COULD NOT BE REWRITTEN SUCCESSFULLY
UNDER THE WRITE RETRY OPTION (SEE ITEMS., SW1:)
FOLLOWING THE COUNT IS A LIST OF THE BAD TAPE
LOCATIONS IDENTIFIED BY THE BLOCK AND RECORD NUMBER
WHEN THE BAD TAPE SPOT WAS LOGGED.

EXAMPLE

DROPS: 0 0 0 0 7 0 0 0
PICKS: 0 0 0 2 0 0 0 0
WTERR: 3
RTRY: 4
RUERR: 6
SOFT: 1
HARD: 5
DTERR: 10
1 BAD TAPE SPOTS
0 *BN 16 *RN 41

11. AUTO SEQUENCE

THE AUTO SEQUENCE (START AT ADDRESS 240) WILL EXECUTE A PREDETERMINED TEST PLAN ON ALL AVAILABLE UNITS. THE ONLY OPERATOR RESPONSE REQUIRED IS TO THE TYPED REQUESTS FOR THE CONTROLLER ADDRESS AND VECTOR AND CONTINUOUS OR SINGLE CYCLE. ALL SWITCHES REMAIN ACTIVE AND MAY BE USED NORMALLY; HOWEVER, THE INTENT IS TO LEAVE ALL SWITCHES DOWN AND ALLOW FULL EXECUTION OF THE TEST PLAN FOR SYSTEM CHECKOUT.

SAMPLE START AT 240(8): AUTO SEQUENCE

LOAD ADDRESS 240(8), SET SWITCHES TO ZERO, PRESS START:

TM, A, B-11 AUTO SEQUENCE TEST
ENTER RESPONSES IN OCTAL

REGISTER START = 172520 (CR)
VECTOR = 224 (CR)
AUTO CONT: 0 (1)

THIS EXAMPLE SHOWS AN AUTO SEQUENCE START WITH THE CONTROLLER AT BUS ADDRESS 172520 AND A VECTOR OR 224. ALL AVAILABLE UNITS WILL BE TESTED CONTINUOUSLY.

AS EACH PASS IS COMPLETED A DIVIDER LINE OF ASTERISKS WILL BE PRINTED FOLLOWED BY AN END OF PASS MESSAGE INDICATING HOW MANY PASSES HAVE BEEN COMPLETED SINCE THE AUTO SEQUENCE WAS BEGUN. AT THE START OF EACH PASS THE UNITS BEING TESTED ARE PRINTED.

AUTO SEQUENCE TEST PLAN:

THE AUTO SEQUENCER WILL EXECUTE A PASS CONSISTING OF THE WRITING, READING, AND CHECKING OF SEVERAL DIFFERENT DATA PATTERNS. EACH PASS WILL START AT BOT AND PROCESS AN ENTIRE MAG TAPE BEFORE REWINDING

THE UNITS WILL BE SET UP TO WRITE 800 BPI IN NINE TRACK FORMAT. ODD PARITY WILL BE USED AND NO TAPE MARKS WILL BE WRITTEN.

THE DATA PATTERNS WILL BE AS FOLLOWS:

THREE FIXED DATA PATTERNS:

EACH PATTERN WILL BE USED FOR SIX BLOCKS.
EACH BLOCK CONSISTS OF (100) 4000 CHARACTER RECORDS.

PATTERN 3: WALKING ONE BIT
PATTERN 7: ALTERNATING ONE AND ZERO BITS
PATTERN 11: INCREMENTING CHARACTERS (000-377)

RANDOM DATA:

FOLLOWING THE FIXED DATA PATTERNS, RANDOM DATA WILL BE WRITTEN IN THE SAME BLOCK STRUCTURE UNTIL EOT IS REACHED.

IT IS IMPORTANT THAT THE TAPE USED FOR THE TEST BE OF SUFFICIENT LENGTH TO ACCOMODATE ALL OF THE FIXED DATA PATTERNS AND AT LEAST ONE RECORD OF RANDOM DATA; OTHERWISE, THE TAPE WILL BE REWOUND UNTIL ALL OF THE DATA PATTERNS HAVE BEEN TESTED.

12. TESTING PROCEDURES

AS PREVIOUSLY STATED THIS PROGRAM CONTAINS NO FIXED TESTS. THE ENTIRE TEST CYCLE TO BE EXECUTED IS DESCRIBED BY THE OPERATOR THROUGH RESPONSES TO TELETYPE REQUESTS FOR PARAMETERS AND CONSOLE SWITCH SETTINGS FOR OPERATION. THE OPERATION SELECTED WILL BE EXECUTED WITH THE PARAMETERS ENTERED CONTINUOUSLY ON EACH AVAILABLE UNIT, ONE BLOCK AT A TIME, UNTIL STOPPED BY THE OPERATOR. THE OPERATION MAY BE CHANGED DYNAMICALLY BY CHANGING THE CONSOLE SWITCHES AT ANY TIME. THE PROGRAM WILL ATTEMPT TO PERFORM ANY OPERATION SET AND THEREFORE CAUTION SHOULD BE TAKEN TO ASSURE THAT UNIT IS CAPABLE OF PERFORMING AS REQUESTED. FOR INSTANCE, ONE SHOULD NOT ATTEMPT TO PERFORM READ OPERATIONS ON A TAPE WHICH HAS NOT BEEN WRITTEN AS THE DATA, IF ANY, IS UNPREDICTABLE. HOWEVER, IF A TAPE HAS BEEN WRITTEN WITH THIS PROGRAM, IT CAN BE READ AS OFTEN AS DESIRED WITHOUT BEING REWRITTEN. THIS IS A GOOD PROCEDURE TO USE FOR TESTING TAPE COMPATABILITY. SCOPING OF TAPE UNITS BECOMES SIMPLE; BY SETTING THE DESIRED OPERATION AND ITS PARAMETER, A UNIT MAY BE CONTINUOUSLY EXERCISED IN ANY MANNER DESIRED. BY USING THE VARIOUS ERROR CONTROL SWITCHES AND ENTERING THE NEEDED STALL, ANY FUNCTION CAN BE SCOPED RATHER EASILY. RELIABILITY TESTING CAN BE PERFORMED BY USE OF THE RANDOMIZATION CAPABILITY. PERHAPS A CYCLE OF RANDOM TESTING MIGHT BE SET UP AND ALLOWED TO RUN FOR SOME PERIOD OF TIME, THE STATISTICAL COLLECTION OF DROPS AND PICKS IS THEN SIGNIFICANT. INTERMITTANT PROBLEMS CAN BE FOUND BY SETTING THE DESIRED OPERATION IN MOTION AND DISALLOWING ERROR PRINTOUTS WHILE ALLOWING A HALT ON ERROR. THE ERROR THAT CAUSED THE HALT CAN BE PRINTED BY RESETTING CONSOLE SWITCH TEN AND PRESSING CONTINUE. IF SOME PARTICULAR DATA PATTERN SHOULD BE CAUSING DATA ERROR, USE OF THE YOZZLE SWITCH AND ITS ASSOCIATED STALL CAN BE USED TO ALLOW SCOPING OF THIS PARTICULAR RECORD.

AS YOU SEE, THERE ARE MYRIAD TESTING PROCEDURES WHICH COULD BE PERFORMED. THE PARAMETERS, TAPE OPERATIONS, ERROR EXAMINATION AND REPORTING ARE ALL AT YOUR DISCRETION.

TRY IT, YOU'LL LIKE IT.

13. LISTING

803

TM, A, B-11 TSO3 OR TUIO, N, W MULTIDRIVE DATA RELIABILITY EXERCISER
DZTMHE.P11 20-DEC-76 12:27

MACY11 27(1006) 20-DEC-76 12:35

SEQ 0027

5619

ACT11 HOOKS

TABLE OF CONTENTS


```

5539 .TITLE TM, A, B-11 TS03 OR TU10, N, W MULTIDRIVE DATA RELIABILITY EXERCISER
5540 ;MAINDEC-11-DZTMH-E-D
5541 ;FEB 1977
5542 ;R. B. BARNES/RON PLATUKIS/R. SOLER
5543 ;ENABLE ABS,AMA
5544
5545 ;CONSOLE SWITCHES*****
5546
5547 ;SW15: 1=STOP ON ERROR
5548 ;       0=CONTINUE ON ERROR
5549 ;
5550 ;SW14: 1=YOZZLE ON CURRENT BLOCK
5551 ;       0=DO NOT YOZZLE ON BLOCK
5552 ;
5553 ;SW13: 1=DO NOT CHECK DATA
5554 ;       0=CHECK DATA
5555 ;
5556 ;SW12: 1=DO NOT CHECK WRITE ERRORS
5557 ;       0=CHECK WRITE ERRORS
5558 ;
5559 ;SW11: 1=DO NOT CHECK READ ERRORS
5560 ;       0=CHECK READ ERRORS
5561 ;
5562 ;SW10: 1=DO NOT PRINT ERRORS
5563 ;       0=PRINT ERRORS
5564 ;
5565 ;SW9:  1=REWIND TAPE
5566 ;       0=DO NOT REWIND
5567 ;
5568 ;SW8:  1=USE RANDOM DATA
5569 ;       0=USE FIXED DATA PATTERN
5570 ;
5571 ;SW7:  1=USE RANDOM CHARACTER COUNT
5572 ;       0=USE FIXED CHAR COUNT
5573 ;
5574 ;SW6:  1=USE RANDOM RECORD COUNT
5575 ;       0=USE FIXED RECORD COUNT
5576 ;
5577 ;SW5:  1=YOZZLE ON CURRENT RECORD
5578 ;       0=DO NOT YOZZLE ON RECORD
5579 ;
5580 ;SW4:  1=PRINT DROPS AND PICKS
5581 ;       0=DO NOT PRINT DROPS AND PICKS
5582 ;
5583 ;SW3:  1=DO NOT READ FORWARD
5584 ;       0=READ FORWARD
5585 ;
5586 ;SW2:  NOT USED
5587 ;
5588 ;SW1:  1=INHIBIT WRITE AND READ RETRY
5589 ;       0=ENABLE WRITE AND READ RETRY
5590 ;
5591 ;SW0:  1=DO NOT WRITE
5592 ;       0=WRITE
5593 ;
5594

```

```

5596
5597 ;REGISTER EQUIVS*****
5598
5599 000000 R0=%0
5600 000001 R1=%1
5601 000302 R2=%2
5602 000003 R3=%3
5603 000004 R4=%4
5604 000005 R5=%5
5605 000006 SP=%6
5606 000007 PC=%7
5607 000240 NOP=240
5608
5609 ;TRAP CATCHERS*****
5610
5611 000000 .=0
5618 000042 .=42
5619 .SBTTL ACT11 HOOKS
(1)
(2) ;:*****
(1) ;HOOKS REQUIRED BY ACT11
(1) 000042 $SVPC=. ;SAVE PC
(1) 000046 .=46
(1) 000046 004650 $ENDAD ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
(1) 000052 000052 .=52 ;;2)SET LOC.52 TO ZERO
(1) 000052 000000 .WORD 0 ;: RESTORE PC
(1) 000042 .=$SVPC
5620
5621 ;TTY INTERRUPT VECTOR*****
5622
5623 000060 .=60
5624 000060 017404 TTINT ;TTY INTERRUPT HANDLER ADDRESS
5625 000062 000000 0
5626
5627
5628 ;SOFTWARE SWITCH REGISTER LOCATIONS*****
5629
5630 000174 .=174
5631 000174 000000 DISPREG:0
5632 000176 000000 SWREG: 0
5633
5634 ;START ADDRESS*****
5635
5636 000200 .=200
5637 000200 000137 002772 JMP START ;ENTER PARAMETERS VIA TTY
5638
5639 000204 .=204
5640 000204 000137 003124 JMP STARTA ;USE FIXED PARAMETERS; HOLD DATA
5641
5642 000210 .=210
5643 000210 005037 013304 CHAIN: CLR RDFL
5644 000214 000137 003142 JMP STARTE ;USE FIXED PARAMETERS; NEW DATA
5645

```

E03

```
5647  
5648 ;MAG TAPE INTERRUPT VECTOR*****  
5649  
5650 . =224  
5651 000224 017466 MTINT ;MAG TAPE INTERRUPT HANDLER ADDRESS  
5652 000226 000340 340  
5653  
5654 :AUTO SEQUENCE START*****  
5655 . =240  
5656 000240 005237 021624 INC ASEQF ;SET AUTO SEQUENCE FLAG  
5657 000244 000137 003106 JMP STAUT ;GO TO START OF AUTO SEG
```

```

5659          000600          . =600
5660          ;CONSTANTS*****
5661
5662 000600 172520 MTS: 172520 ;TAPE STATUS REGISTER
5663 000602 172522 MTC: 172522 ;TAPE COMMAND REGISTER
5664 000604 172524 MWC: 172524 ;TAPE CHARACTER COUNT REGISTER
5665 000606 172526 MDA: 172526 ;TAPE DATA ADDRESS REGISTER
5666 000610 172530 MTD: 172530 ;TAPE DATA BUFFER
5667 000612 172532 MTRD: 172532 ;TAPE READ LINES
5668 000614 000224 VECT: 224 ;INTERRUPT VECTOR ADDRESS
5669 000616 000000 UDES: 0 ;UNIT DESCRIPTION (PARITY, DENSITY, UNIT, TRACK)
5670 000620 000100 RCNT: 100 ;RECORD COUNTER
5671 000622 177600 CARCNT: 177600 ;NUMBER OF CHAR (2 - 4000) OCTAL IN TWOS COMPLEMENT
5672 000624 000001 PATRN: 1 ;DATA PATTERN SELECTOR (0 - 20) OCTAL
5673 000626 000002 RDCMD: 2 ;READ COMMAND
5674 000630 000001 SPFLG: 1 ;SINGLE PASS FLAG
5675 000632 000001 RSTAL: 1 ;READ STALL
5676 000634 000001 WSTAL: 1 ;WRITE STALL
5677 000636 000001 TSTAL: 1 ;TURN AROUND STAL
5678 000640 001000 YSTAL: 1000 ;YOZZLE STAL
5679 000642 000100 RCSAV: 100 ;RECORD COUNT SAVE
5680 000644 177600 CCSAV: -200 ;CHARACTER COUNT SAVE
5681 000646 000000 TMEX: 0 ;TAPE MARK FLAG: 1=TM 0=NO TM
5682 000650 177776 PSW: 177776 ;PROCESSOR STATUS
5683 000652 177570 SWR: 177570 ;CONSOLE SWITCHES
5684 000654 177570 DISPLAY: 177570
5685 000656 177560 TKS: 177560 ;TTY READ STATUS REGISTER
5686 000660 177562 TKB: 177562 ;TTY READ BUFFER
5687 000662 177564 TPS: 177564 ;TTY PUNCH STATUS REGISTER
5688 000664 177566 TPB: 177566 ;TTY PUNCH OUTPUT REGISTER
5689 000666 177550 PRS: 177550 ;H/S READER STATUS REGISTER
5690 000670 177552 PRB: 177552 ;H/S READER BUFFER
5691 000672 153624 RANBAS: 153624 ;RANDOM NUMBER GENERATOR BASE
5692 000674 172520 REGST: 172520 ;STARTING REGISTER ADDRESS
5693 000676 032561 RANSAB: 032561 ;RANDOM NUMBER BUFFER
5694
    
```

5696
5697
5698
5699 000700 000000
5700 000702 000000
5701 000704 000000
5702 000706 000000
5703 000710 000000
5704 000712 000000
5705 000714 000000
5706 000716 000000
5707 000720 000000
5708 000722 000000
5709 000724 000000
5710 000726 000000
5711 000730 000000
5712 000732 000000
5713 000734 000000
5714 000736 000000
5715 000740 000000
5716 000742 000000
5717 000744 000000
5718 000746 000000
5719 000750 000000
5720 000752 000000
5721 000754 000000
5722 000756 000000
5723 000760 000000
5724 000762 000000
5725 000764 000000
5726 000766 000000
5727 000770 000000
5728 000772 000000
5729 000774 000000
5730 000776 000000
5731 001000 000000
5732 001002 000000
5733 001004 000000
5734 001006 000000
5735 001010 000000
5736

:FLAGS AND COUNTERS*****

TINF: 0 :TTY ENTERY FLAG
TOB: 0 :TTY OUTPUT BUFFER
TIB: 0 :TTY INPUT BUFFER
TEMP1: 0 :TEMP STORAGE
TEMP2: 0 :TEMP STORAGE
TEMP3: 0 :TEMP STORAGE
TEMP4: 0 :TEMP STORAGE
EMADDR: 0 :ERROR MSG ADDRESS STORAGE
BLCNTR: 0 :BLOCK COUNTER
BBC: 0 :BAD RECORD COUNTER
RTRN: 0 :INTERRUPT RETURN STORAGE
HDRFL: 0 :HEADER FLAG
STAL: 0 :DELAY STORAGE
PFLG: 0 :PRINT FLAG
UNP: 0 :UNIT TABLE POINTER
BCNT: 0 :BIT COUNTER
ERSAV: 0 :STATUS STORAGE
SERFL: 0 :STATUS ERROR FLAG
DERFL: 0 :DATA ERROR FLAG
BTFLG: 0 :BAD TAPE FLAG
RPCNT: 0 :REPEAT COUNTER
RTCNT: 0 :RETRY COUNTER
RTYFL: 0 :RETRY FLAG
TMFLG: 0 :TM FLAG
EOTREC: 0 :END OF TAPE RECORD
BTPT: 0 :BAD TAPE POINTER
ERTFL: 0 :ERASE T F FLAG
BDPP: 0 :DROP PC IER
BPKP: 0 :PICK POINTER
BTSTF: 0 :BAD TAPE STATISTICS FLAG
RRTYFL: 0 :READ RETRY FLAG
SEQCT: 0 :AUTO SEQ PASS COUNT
COUNT: 0
TEMPST: 0
RDSW: 0
DICTR: 0 :DROPPED UNIT COUNTER
STCDFL: 0 :7 TRK CORE DUMP FLAG

```

5738
5739 ;UNIT ORDER AND DESCRIPTION TABLE *****
5740
5741 001012 060000 UN1: 60000 ;THIS TABLE IS LOADED
5742 001014 177777 UN2: -1 ;WITH UNIT NUMBERS AND
5743 001016 177777 UN3: -1 ;THEIR DESCRIPTIONS IN
5744 001020 177777 UN4: -1 ;THE ORDER THAT THEY
5745 001022 177777 UN5: -1 ;WILL BE TESTED
5746 001024 177777 UN6: -1
5747 001026 177777 UN7: -1
5748 001030 177777 UN8: -1
5749 001032 177777 UNX: -1

```

```

5750
5751 ;UNIT DROPS AND PICKS COUNTERS*****
5752
5753 001034 001254 PIK1: BP00
5754 001036 001274 PIK2: BP10
5755 001040 001314 PIK3: BP20
5756 001042 001334 PIK4: BP30
5757 001044 001354 PIK5: BP40
5758 001046 001374 PIK6: BP50
5759 001050 001414 PIK7: BP60
5760 001052 001434 PIK8: BP70
5761 001054 001454 DRP1: BD00
5762 001056 001474 DRP2: BD10
5763 001060 001514 DRP3: BD20
5764 001062 001534 DRP4: BD30
5765 001064 001554 DRP5: BD40
5766 001066 001574 DRP6: BD50
5767 001070 001614 DRP7: BD60
5768 001072 001634 DRP8: BD70

```

```

5769
5770 ;UNIT WRITE ERRORS*****
5771
5772 001074 000000 WTER1: 0
5773 001076 000000 WTER2: 0
5774 001100 000000 WTER3: 0
5775 001102 000000 WTER4: 0
5776 001104 000000 WTER5: 0
5777 001106 000000 WTER6: 0
5778 001110 000000 WTER7: 0
5779 001112 000000 WTER8: 0

```

```

5780 ;UNIT READ ERRORS*****
5781
5782 001114 000000 RDER1: 0
5783 001116 000000 RDER2: 0
5784 001120 000000 RDER3: 0
5785 001122 000000 RDER4: 0
5786 001124 000000 RDER5: 0
5787 001126 000000 RDER6: 0
5788 001130 000000 RDER7: 0
5789 001132 000000 RDER8: 0
5790
5791

```

;UNIT DATA ERRORS*****

5793		
5794		
5795	001134	000000
5796	001136	000000
5797	001140	000000
5798	001142	000000
5799	001144	000000
5800	001146	000000
5801	001150	000000
5802	001152	000000

DATER1:	0
DATER2:	0
DATER3:	0
DATER4:	0
DATER5:	0
DATER6:	0
DATER7:	0
DATER8:	0

;UNIT RETRY COUNTERS*****

5803		
5804		
5805		
5806	001154	000000
5807	001156	000000
5808	001160	000000
5809	001162	000000
5810	001164	000000
5811	001166	000000
5812	001170	000000
5813	001172	000000

RTY1:	0
RTY2:	0
RTY3:	0
RTY4:	0
RTY5:	0
RTY6:	0
RTY7:	0
RTY8:	0

;UNIT SOFT READ ERRORS*****

5814		
5815		
5816		
5817	001174	000000
5818	001176	000000
5819	001200	000000
5820	001202	000000
5821	001204	000000
5822	001206	000000
5823	001210	000000
5824	001212	000000

GDRTY1:	0
GDRTY2:	0
GDRTY3:	0
GDRTY4:	0
GDRTY5:	0
GDRTY6:	0
GDRTY7:	0
GDRTY8:	0

;UNIT HARD READ ERRORS*****

5825		
5826		
5827		
5828	001214	000000
5829	001216	000000
5830	001220	000000
5831	001222	000000
5832	001224	000000
5833	001226	000000
5834	001230	000000
5835	001232	000000

BDRTY1:	0
BDRTY2:	0
BDRTY3:	0
BDRTY4:	0
BDRTY5:	0
BDRTY6:	0
BDRTY7:	0
BDRTY8:	0

;UNIT EOT COUNTERS*****

5836		
5837		
5838		
5839	001234	000000
5840	001236	000000
5841	001240	000000
5842	001242	000000
5843	001244	000000
5844	001246	000000
5845	001250	000000
5846	001252	000000

EOTCT1:	0
EOTCT2:	0
EOTCT3:	0
EOTCT4:	0
EOTCT5:	0
EOTCT6:	0
EOTCT7:	0
EOTCT8:	0

5847

```

5849
5850                                     :DROPS + PICKS PER CHANNEL PER UNIT.*****
5851
5852 001254 000000 BP00: 0
5853         001274         .=.+16
5854 001274 000000 BP10: 0
5855         001314         .=.+16
5856 001314 000000 BP20: 0
5857         001334         .=.+16
5858 001334 000000 BP30: 0
5859         001354         .=.+16
5860 001354 000000 BP40: 0
5861         001374         .=.+16
5862 001374 000000 BP50: 0
5863         001414         .=.+16
5864 001414 000000 BP60: 0
5865         001434         .=.+16
5866 001434 000000 BP70: 0
5867         001454         .=.+16
5868 001454 000000 B000: 0
5869         001474         .=.+16
5870 001474 000000 B010: 0
5871         001514         .=.+16
5872 001514 000000 B020: 0
5873         001534         .=.+16
5874 001534 000000 B030: 0
5875         001554         .=.+16
5876 001554 000000 B040: 0
5877         001574         .=.+16
5878 001574 000000 B050: 0
5879         001614         .=.+16
5880 001614 000000 B060: 0
5881         001634         .=.+16
5882 001634 000000 B070: 0
5883         001654         .=.+16
5884
5885                                     ;UNIT BAD TAPE COUNTER: 16 PER DRIVE
5886
5887 001654 000000 BT00: 0
5888         001760         .=.+102
5889 001760 000000 BT01: 0
5890         002064         .=.+102
5891 002064 000000 BT02: 0
5892         002170         .=.+102
5893 002170 000000 BT03: 0
5894         002274         .=.+102
5895 002274 000000 BT04: 0
5896         002400         .=.+102
5897 002400 000000 BT05: 0
5898         002504         .=.+102
5899 002504 000000 BT06: 0
5900         002610         .=.+102
5901 002610 000000 BT07: 0
5902         002714         .=.+102

```


K03

5904
5905
5906
5907 002714 001654
5908 002716 001760
5909 002720 002064
5910 002722 002170
5911 002724 002274
5912 002726 002400
5913 002730 002504
5914 002732 002610
5915
5916
5917
5918 002734 002734
5919 002736 012552
5920 002740 012754
5921 002742 012776
5922 002744 013004
5923 002746 013032
5924 002750 013044
5925 002752 013054
5926 002754 013064
5927 002756 013074
5928 002760 013104
5929 002762 013126
5930 002764 013152
5931 002766 013162
5932 002770 013172
5933
5934

;UNIT BAD TAPE POINTERS*****

BTADDR: BT00
BT01
BT02
BT03
BT04
BT05
BT06
BT07

;DATA PATTERN GENERATORS*****

DATBL: . ;ENTRY TABLE
DATA0: DAT0 ;EXTERNAL INPUT FROM H/S READER
DATA1: DAT1 ;ALL ONES
DATA2: DAT2 ;ALL ZEROS
DATA3: DAT3 ;WALKING ONE
DATA4: DAT4 ;WALKING ZERO
DATA5: DAT5 ;ALTERNATING ONE/ZERO
DATA6: DAT6 ;ALTERNATING ZERO/ONE
DATA7: DAT7 ;ALTERNATING ONE/ZERO IN ALTERNATING CHARACTERS
DATA10: DAT10 ;ALTERNATING ZERO/ONE IN ALTERNATING CHARACTERS
DATA11: DAT11 ;ALL BITS 0-377
DATA12: DAT12 ;ALL BITS 377-0
DATA13: DAT13 ;ALTERNATING CHARACTERS 0 AND 377
DATA14: DAT14 ;ALTERNATING CHARACTERS 377 AND 0
DATA15: DAT15 ;WALKING ZERO REPEATED FOUR TIMES

5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946
5947
5948
5949
5950
5951
5952
5953
5954
5955
5956
5957
5958
5959
5960
5961
5962
5963
5964
5965
5966
5967
5968
5969
5970
5971
5972
5973
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986
5987
5988
5989
5990
5991

002772 005037 021624
002776 012737 177570 000652
003004 005737 000042
003010 001436
003012 012706 000500
003016 012704 023144
003022 004737 020530
003026 122737 000004 000041
003034 001006
003036 012704 026116
003042 004737 020530
003046 000137 004622
003052 012737 000176 000072
003060 012700 001014
003064 022720 177777
003070 001404
003072 062737 000401 004715
003100 000771
003102 000137 000210
003106 012737 000001 000700
003114 005037 013304
003120 000137 003146
003124 005037 000700
003130 012706 000500
003134 004737 022040
003140 000451
003142 005037 000700
003146 012700 000702
003152 012701 000044
003156 005020
003160 005301
003162 001375
003164 012700 000510
003170 012701 001074
003174 005021
003176 005300

```
.EVEN
*****
PROGRAM START AND SEQUENCE FORMATTER:
THIS ROUTINE IS USED TO PERFORM ALL HOUSEKEEPING,
DECIDE WHICH TRANSPORT TO TEST AND ITS AVAILABILITY,
LOAD THE WRITE BUFFER WITH THE SELECTED DATA PATTERN,
GENERATE ANY RANDOM NUMBER AND THEN EXECUTE
THE TEST CYCLE REQUESTED BY THE SWITCH SETTING.
AT THE END OF THE TEST CYCLE THE NEXT UNIT IS SELECTED
AND CHECKED FOR AVAILABILITY AND THE TEST CYCLE IS
EXECUTED ON IT.
THE NUMBER OF BITS DROPPED OR PICKED MAY BE PRINTED
AT THE END OF EACH TEST CYCLE VIA CONSOLE SWITCH FOUR (4).
*****
START: CLR ASEQF ;CLEAR AUTO SEQ FLAG
MOV #177570,SWR ;PRESET FOR CONSOLE SWITCHES
TST @#42 ;SEE IF CHAIN MODE
BEQ STAUT ;IF NOT: BR
MOV #500,SP ;SET UP STACK POINTER
MOV #MSG31,R4
JSR PC,TTOUT ;PRINT TITLE
CMPB #4,@#41 ;SEE IF LOAD MEDIUM
BNE 1$ ;IF NOT: BR
MOV #MSG97,R4
JSR PC,TTOUT ;PRINT NO TEST LOAD MEDIUM
JMP REOTB ;END TEST
1$: MOV #176,SWR ;SET FOR SOFTWARE SWITCHES
MOV #UN2,RO ;SET UNIT POINTER
2$: CMP #-1,(RO)+ ;SEE IF END OF UNITS
BEQ 3$ ;IF SO: BR
ADD #401,REOTC ;ELSE BUMP UNIT EOT COUNTER
BR 2$
3$: JMP CHAIN ;GO DO CHAIN START
STAUT: MOV #1,TINF ;SET TTY ENTRY FLAG
CLR ROFL ;CLEAR RANDOM DATA FLAG
JMP STARTB
STARTA: CLR TINF ;CLEAR TTY ENTRY FLAG
MOV #500,SP ;SET STACK POINTER
JSR PC,SUSWR ;CHECK FOR SOFTSWR
BR STAUTO
STARTE: CLR TINF ;CLEAR INPUT FLAG
STARTB: MOV #TOB,RO
MOV #44,R1
STARTD: CLR (RO)+ ;CLEAR FLAGS AND COUNTERS
DEC R1
BNE STARTD
MOV #510,RO ;SET SIZE OF TABLE
MOV #WTER1,R1 ;SET START OF TABLE
STARTC: CLR (R1)+ ;CLEAR STATISTICS TABLES
DEC RO
```

5992	003200	001375			BNE	STARTC		; CLEAR ALL
5993	003202	012737	177777	012750	MOV	#-1, PATS		; RESET PATTERN
5994	003210	012737	177777	012752	MOV	#-1, PARS		; RESET PARITY
5995	003216	012737	000001	000720	MOV	#1, BLCNTR		; PRESET BLOCK COUNTER
5996	003224	005077	175352		CLR	QMTC		
5997	003230	052777	010000	175344	BIS	#10000, QMTC		; POWER CLEAR CONTROLLER
5998	003236	012706	000500		STARTD: MOV	#500, SP		
5999	003242	004737	022040		JSR	PC, SUSWR		; CHECK FOR SORTSWR
6000	003246	012777	000340	175374	1\$: MOV	#340, QPSW		
6001	003254	004737	010722		JSR	PC, TINP		; GO GET PARAMETERS FROM TTY
6002	003260	004737	004110		JSR	PC, RANSET		; GO RESET BASE
6003	003264	005000			STARTO: CLR	RO		; POINT TO FIRST ENTRY
6004	003266	022737	000176	000652	CMP	#SWREG, SWR		; TEST FOR SOFTSWR
6005	003274	001005			BNE	STAROA		
6006	003276	005737	000042		TST	Q#42		; SEE IF CHAIN MODE
6007	003302	001002			BNE	STAROA		; IF SO: BR
6008	003304	004737	022164		JSR	PC, CNTLU		; ASK FOR CONTROL SETTINGS
6009	003310	005160	001012		STAROA: COM	UNI(RO)		; SEE IF LAST ENTRY
6010	003314	001411			BEQ	STAROB		; IF SO: BR
6011	003316	005160	001012		COM	UNI(RO)		
6012	003322	042760	100200	001012	BIC	#100200, UNI(RO)		; CLEAR EOT/DROPPED FLAG
6013	003330	062700	000002		ADD	#2, RO		; POINT TO NEXT UNIT ENTRY
6014	003334	000137	003310		JMP	STAROA		; CONTINUE CLEARING
6015	003340	005160	001012		STAROB: COM	UNI(RO)		
6016	003344	013703	004716		MOV	REOTC, R3		
6017	003350	000303			SWAB	R3		
6018	003352	110337	004716		MOV	R3, REOTC		; RESTORE EOT CNTR
6019	003356	012777	000100	175272	START1: MOV	#100, QTKS		; SET TTY INTERRUPT ENABLE
6020	003364	013700	000734		MOV	UNP, RO		; RO = UNIT TABLE POINTER
6021	003370	005160	001012		COM	UNI(RO)		
6022	003374	001407			BEQ	STAR1B		; IF LAST UNIT IN STRING: BR
6023	003376	005160	001012		COM	UNI(RO)		
6024	003402	016037	001012	000616	STAR1A: MOV	UNI(RO), UDES		; LOAD NEXT UNIT DESCRIPTION
6025	003410	000137	003542		JMP	START4		
6026	003414	005237	000720		STAR1B: INC	BLCNTR		; BUMP BLOCK COUNTER
6027	003420	005737	021624		TST	ASEQF		; SEE IF AUTO SEQ
6028	003424	001414			BEQ	STAR1C		; IF NOT: BR
6029	003426	023737	000720	021622	CMP	BLCNTR, ABLCNT		; SEE IF DONE SEQ
6030	003434	001010			BNE	STAR1C		; IF NOT: BR
6031	003436	005160	001012		COM	UNI(RO)		; RESET UNIT TABLE TERMINATOR
6032	003442	012737	000001	000720	MOV	#1, BLCNTR		; RESET BLOCK COUNTER
6033	003450	005037	000734		CLR	UNP		; RESET UNIT POINTER
6034	003454	000207			RTS	PC		; RETURN TO AUTO SEQ
6035	003456	005037	000734		STAR1C: CLR	UNP		
6036	003462	005160	001012		COM	UNI(RO)		
6037	003466	005000			CLR	RO		
6038	003470	016037	001012	000616	MOV	UNI(RO), UDES		; LOAD FIRST UNIT DESCRIPTION
6039	003476	032777	000200	175146	BIT	#200, QSWR		; SEE IF RANDOM RECORD SIZE
6040	003504	001402			BEQ	START2		; IF NOT: BR
6041	003506	004737	010644		JSR	PC, CCNTR		; GO GENERATE RANDOM CHAR COUNT
6042	003512	032777	000400	175132	START2: BIT	#400, QSWR		; SEE IF RANDOM DATA
6043	003520	001402			BEQ	START3		; IF NOT: BR
6044	003522	004737	013236		JSR	PC, DATR		; GO GENERATE RANDOM DATA
6045	003526	032777	000100	175116	START3: BIT	#100, QSWR		; SEE IF RANDOM RECORD COUNT
6046	003534	001402			BEQ	START4		; IF NOT: BR
6047	003536	004737	010676		JSR	PC, RCNTR		; GO GENERATE RANDOM RECORD COUNT

N03

TM,A,B-11 TS03 OR TU10,N,W MULTIDRIVE DATA RELIABILITY EXERCISER
 02TMHE.P11 20-DEC-76 12:27 ACT11 HOOKS

MACY11 27(1006) 20-DEC-76 12:35 PAGE 68-2

SEQ 0039

6048	003542	032760	100000	001012	START4:	BIT	#100000,UNI(RO)	;SEE IF UNIT REACHED EOT OR DROPPED
6049	003550	001404				BEQ	STAR40	;IF NOT: BR
6050	003552	062737	000002	000734		ADD	#2,UNP	;POINT TO NEXT UNIT
6051	003560	000676				BR	START1	
6052	003562	013777	000616	175012	STAR40:	MOV	UDES,AMTC	;SET UNIT NUMBER
6053	003570	004737	021226			JSR	PC,STDLY	;GO AWAIT ASSURED STATUS
6054	003574	032777	000001	174776		BIT	#1,AMTS	;SEE IF TUR
6055	003602	001030				BNE	STAR46	;IF SO: BR
6056	003604	032777	000002	174766		BIT	#2,AMTS	;SEE IF REWINDING
6057	003612	001414				BEQ	STAR45	;IF NOT: BR
6058	003614	004737	017506			JSR	PC,PAPRT	;PRINT HEADER
6059	003620	012704	025475			MOV	#MSG89,R4	
6060	003624	004737	020530			JSR	PC,TTOUT	;PRINT REWIND MSG
6061	003630	032777	000001	174742	STAR44:	BIT	#1,AMTS	
6062	003636	001774				BEQ	STAR44	;AWAIT REWIND DONE
6063	003640	000137	003664			JMP	STAR46	
6064	003644	004737	017506		STAR45:	JSR	PC,PAPRT	;PRINT HEADER
6065	003650	012704	023701			MOV	#MSG49,R4	
6066	003654	004737	020530			JSR	PC,TTOUT	;PRINT NOT AVAIL
6067	003660	000137	020050			JMP	DRPDRV	;GO DROP DRIVE
6068	003664	005037	001010		STAR46:	CLR	STCDFL	;CLEAR 7 TRK CORE DUMP FLAG
6069	003670	032777	000020	174702		BIT	#20,AMTS	;SEE IF 7 TRK
6070	003676	001411				BEQ	1\$;IF NOT: BR
6071	003700	013704	000616			MOV	UDES,R4	;GET UNIT DESCRIPTION
6072	003704	042704	117777			BIC	#117777,R4	;MASK DENSITY
6073	003710	022704	060000			CMP	#60000,R4	;SEE IF CORE DUMP
6074	003714	001002				BNE	1\$;IF NOT: BR
6075	003716	005237	001010			INC	STCDFL	;ELSE SET FLAG
6076	003722	004737	012370		1\$:	JSR	PC,DSUP	;GO SET UP WRITE DATA
6077	003726	004737	004720			JSR	PC,RWNO	;REWIND
6078	003732	004737	005254			JSR	PC,WRITE	;WRITE
6079	003736	013737	000636	000730		MOV	T\$AL,STAL	;SET TURN AROUND DELAY
6080	003744	004737	010634			JSR	PC,STALL	;DELAY
6081	003750	004737	006626			JSR	PC,RSEQ	;GO TO READ SEQUENCER
6082	003754	013737	000636	000730		MOV	T\$AL,STAL	;SET TURN AROUND DELAY
6083	003762	004737	010634			JSR	PC,STALL	;DELAY
6084	003766	032777	000020	174656		BIT	#20,ASWR	;SEE IF SHOULD PRINT DROPS AND PICK
6085	003774	001410				BEQ	START5	;IF NOT: BR
6086	003776	012700	000001			MOV	#1,RO	;SET RECORD COUNTER TO 1
6087	004002	005237	000772			INC	BT\$TF	;SET FOR STAT PRINT ONLY
6088	004006	004737	015370			JSR	PC,PRSTAT	;PRINT STATISTICS
6089	004012	005037	000772			CLR	BT\$TF	;CLEAR FLAG
6090	004016	017700	174630		START5:	MOV	ASWR,RO	;LOAD SWR
6091	004022	042700	177762			BIC	#177762,RO	;MASK READ/WRITE SWITCHES
6092	004026	022700	000015			CMP	#15,RO	;SEE IF HAVE READ OR WRITE
6093	004032	001424				BEQ	START8	;IF NOT: BR
6094	004034	032777	000001	174536	START6:	BIT	#1,AMTS	;SEE IF HAVE UNIT READY
6095	004042	001013				BNE	START7	;IF SO: BR
6096	004044	005337	000730			DEC	STAL	
6097	004050	001371				BNE	START6	;DELAY FOR TUR
6098	004052	004737	017506			JSR	PC,PAPRT	;PRINT HEADER
6099	004056	012704	023701			MOV	#MSG49,R4	
6100	004062	004737	020530			JSR	PC,TTOUT	;PRINT NOT AVAIL
6101	004066	000137	020050			JMP	DRPDRV	;GO DROP DRIVE
6102	004072	062737	000002	000734	START7:	ADD	#2,UNP	;POINT TO NEXT UNIT
6103	004100	005077	174476			CLR	AMTC	

B04

SEQ 0040

```
6104 004104 000137 003356      START8: JMP      START1      ;CONTINUE
6105
6106                          ;RANDOM BASE RESET*****
6107
6109 004110 012737 153624 000672 RANSET: MOV      #153624, RANBAS ;RESET BASE
6109 004116 012737 032561 000676      MOV      #32561, RANSV  ;RESET BUFFER
6110 004124 013737 000642 000620      MOV      RCSAV, RCNT   ;RESET RECORD COUNT
6111 004132 013737 000644 000622      MOV      CCSAV, CARCNT ;RESET CHAR COUNT
6112 004140 000207
```

```

6114 ;*****
6115 ;REWIND FROM EOT:
6116 ;
6117 ;WHEN ANY TRANSPORT BEING TESTED REACHES END OF TAPE
6118 ;DURING A READ OR WRITE OPERATION, IT WILL BE REWOUND
6119 ;AND FLAGGED AS UNAVAILABLE UNTIL ALL AVAILABLE UNITS
6120 ;HAVE REACHED EOT AT WHICH TIME ALL TESTING WILL BE RESUMED
6121 ;AT A BLOCK COUNT OF ONE (1). A MESSAGE WILL BE
6122 ;PRINTED ON THE SUPERVISORS CONSOLE AS EACH UNIT REACHES
6123 ;EOT AND IS REWOUND.
6124 ;*****
6125
6126 004142 013777 000616 174432 REOT: MOV UDES,AMTC ;LOAD COMMAND REGISTER
6127 004150 032777 000010 174422 REOT1: BIT #10,AMTS
6128 004156 001374 BNE REOT1 ;AWAIT SETTLE DOWN RESET
6129 004160 052777 000017 174414 BIS #17,AMTC ;START REWIND
6130 004166 004737 017506 JSR PC,PAPRT ;PRINT HEADER
6131 004172 032737 000004 000746 BIT #4,BTFLG ;ERROR DURING RETRY?
6132 004200 001405 BEQ 1$ ;IF NOT: BR
6133 004202 012704 025466 MOV #MSG88,R4
6134 004206 004737 020530 JSR PC,TTOUT ;PRINT RETRY
6135 004212 000404 BR 2$
6136 004214 032737 000002 000746 1$: BIT #2,BTFLG ;BACKSPACE ERROR
6137 004222 001405 BEQ REOT1C ;IF NOT: BR
6138 004224 012704 024304 2$: MOV #MSG61,R4 ;POINT TO BACKSPACE ERROR MESSG.
6139 004230 005037 000746 CLR BTFLG ;CLEAR BAD TAPE FLAG
6140 004234 000437 BR REOT1B
6141 004236 005737 000746 REOT1C: TST BTFLG ;TEST BAD TAPE FLAG
6142 004242 001405 BEQ REOT1D ;IF NOT: BR
6143 004244 012704 024114 MOV #MSG59,R4 ;SET UP BAD TAPE MESSAGE
6144 004250 005037 000746 CLR BTFLG ;CLEAR BAD TAPE FLAG
6145 004254 000427 BR REOT1B
6146 004256 005737 021624 REOT1D: TST ASEQF ;IS IT AUTO SEQ?
6147 004262 001406 BEQ REOT1A ;IF NOT: BR
6148 004264 005737 000624 TST PATRN ;IS IT RANDOM DATA?
6149 004270 100403 BMI REOT1A ;IF SO: BR
6150 004272 012704 025332 MOV #MSG87,R4 ;PRINT EARLY ASEQ EOT MESSG.
6151 004276 000416 BR REOT1B
6152 004300 012704 022572 REOT1A: MOV #MSG20,R4
6153 004304 004737 020530 JSR PC,TTOUT ;PRINT EOT MESSAGE
6154 004310 013704 000734 MOV UNP,R4
6155 004314 005264 001234 INC EOTCT1(R4) ;BUMP EOT COUNTER
6156 004320 016403 001234 MOV EOTCT1(R4),R3
6157 004324 004737 020716 JSR PC,OCTP ;PRINT EOT COUNT
6158 004330 012704 022606 MOV #MSG20A,R4
6159 004334 004737 020530 REOT1B: JSR PC,TTOUT ;PRINT REWIND MSG
6160 004340 004737 015400 JSR PC,PRSTA2 ;PRINT STATS WITHOUT HEADER
6161 004344 032777 000200 174230 REOT2: BIT #200,AMTC
6162 004352 001774 BEQ REOT2 ;AWAIT CUR *
6163 004354 105337 004716 DECB REOTC ;SEE IF LAST UNIT TO REACH EOT
6164 004360 001410 BEQ REOT3 ;IF SO: BR
6165 004362 013700 000734 MOV UNP,R0
6166 004366 052760 100000 001012 BIS #100000,UN1(R0) ;SET EOT FLAG
6167 004374 005726 TST (SP)+
6168 004376 000137 004072 JMP START7 ;GO TO NEXT UNIT
6169 004402 000337 004716 REOT3: SWAB REOTC

```

```

6170 004406 013700 004716      MOV      REOTC,RO
6171 004412 000337 004716      SWAB    REOTC
6172 004416 110037 004716      MOVB   RO,REOTC      ;RESTORE EOT UNIT COUNTER
6173 004422 005037 000734      CLR    UNP
6174 004426 013700 000734      MOV    UNP,RO      ;POINT TO FIRST UNIT
6175 004432 016037 001012 000616 REOT4:  MOV    UN1(RO),UDES ;LOAD UNIT DESCRIPTION
6176 004440 032737 000200 000616      BIT    #200,UDES    ;SEE IF UNIT IS DROPPED
6177 004446 001034      BNE    REOT6A      ;IF SO: BR
6178 004450 013777 000616 174124      MOV    UDES,AMTC   ;LOAD COMMAND REGISTER
6179 004456 032777 000002 174114 REOT5:  BIT    #2,AMTS
6180 004464 001374      BNE    REOT5      ;AWAIT RWS RESET
6181 004466 032777 000040 174104      BIT    #40,AMTS    ;SEE IF HAVE BOT
6182 004474 001012      BNE    REOT6      ;IF SO: BR
6183 004476 012700 000001      MOV    #1,RO
6184 004502 004737 017506      JSR    PC,PPRPT    ;PRINT HEADER
6185 004506 012704 023653      MOV    #MSG48,R4
6186 004512 004737 020530      JSR    PC,TTOUT    ;PRINT BOT ERROR
6187 004516 000137 020050      JMP    DRPDRV      ;GO DROP DRIVE
6188 004522 032777 000010 174050 REOT6:  BIT    #10,AMTS    ;SEE IF SWDN IS RESET
6189 004530 001374      BNE    REOT6      ;IF NOT: AWAIT SWDN RESET
6190 004532 042760 100200 001012      BIC    #100200,UN1(RO) ;CLEAR EOT/DROPPED FLAG
6191 004540 062737 000002 000734 REOT6A:  ADD    #2,UNP
6192 004546 013700 000734      MOV    UNP,RO      ;POINT TO NEXT UNIT
6193 004552 005160 001012      COM    UN1(RO)     ;SEE IF LAST UNIT
6194 004556 001404      BEQ    REOT7      ;IF SO: BR
6195 004560 005160 001012      COM    UN1(RO)
6196 004564 000137 004432      JMP    REOT4      ;DO NEXT UNIT
6197 004570 005160 001012      COM    UN1(RO)
6198 004574 012737 000001 000720 REOT7:  MOV    #1,BLCNTR  ;SET TO BLOCK COUNT 1
6199 004602 005037 000734      CLR    UNP
6200 004606 005000      CLR    RO          ;SET TO RESTART WITH FIRST UNIT
6201 004610 005726      TST    (SP)+       ;RESET STACK
6202 004612 005737 021624      TST    ASEQF      ;SEE IF AUTO SEQ
6203 004616 001401      BEQ    REOT8      ;IF NOT: BR
6204 004620 000207      RTS    PC          ;RETURN
6205 004622 012704 023473      MOV    #MSG39,R4
6206 004626 004737 020530      JSR    PC,TTOUT    ;PRINT END OF PASS
6207 004632 005737 000630      TST    SPFLG      ;SEE IF SINGLE PASS
6208 004636 001412      BEQ    REOTX      ;IF NOT: BR
6209 004640 013704 000042      MOV    #42,R4
6210 004644 001405      BEQ    HERE      ;IF NOT CHAIN MODE: BR
6211 004646 000005      RESET
6212 004650 004714      SENDAD: JSR    PC,(R4)
6213 004652 000240      NOP
6214 004654 000240      NOP
6215 004656 000240      NOP
6216 004660 000240      HERE:  NOP
6217 004662 000000      REOT10: HALT
6218 004664 012706 000500      REOTX:  MOV    #500,SP   ;RESET STACK
6219 004670 004737 004110      JSR    PC,RANSET  ;GO RESET RANDOM BASE
6220 004674 012737 177777 012750      MOV    #-1,PATS   ;PRESET PATTERN
6221 004702 005037 013304      CLR    RDFL       ;CLEAR RANDOM DATA FLAG
6222 004706 005037 001006      CLR    DUCTR      ;CLEAR DROPPED UNITER COUNTER
6223 004712 000137 003264      JMP    STAUTO     ;RESTART AT BLOCK NUMBER ONE
6224 004716 000401      REOTC:  401      ;EOT UNIT COUNTER(DEFAULT TO ONE UNIT)

```

```

6226
6227
6228
6229
6230
6231
6232
6233
6234
6235 004720 032777 001000 173724 RWND: BIT #1000, @SWR ;SEE IF SHOULD REWIND
6236 004726 001001 BNE RWNDA ;IF SO: BR
6237 004730 000207 RTS PC ;ELSE EXIT
6238 004732 005037 000734 RWNDA: CLR UNP ;CLEAR POINTER
6239 004736 000337 004716 SWAB REOTC
6240 004742 013700 004716 MOV REOTC, RO
6241 004746 000337 004716 SWAB REOTC
6242 004752 110037 004716 MOVB RO, REOTC ;RESTORE EOT UNIT COUNTER
6243 004756 013700 000734 RWND0: MOV UNP, RO ;POINT TO UNIT ENTRY
6244 004762 005160 001012 COM UN1(RO) ;SEE IF LAST ENTRY
6245 004766 001424 BEQ RWND2 ;IF SO: BR
6246 004770 005160 001012 COM UN1(RO)
6247 004774 016037 001012 000616 MOV UN1(RO), UDES ;SET UNIT DESCRIPTION
6248 005002 013777 000616 173572 MOV UDES, @MTC ;LOAD COMMAND REGISTER
6249 005010 052777 000017 173564 BIS #17, @MTC ;START REWIND
6250 005016 032777 000200 173556 RWND1: BIT #200, @MTC
6251 005024 001774 BEQ RWND1 ;AWAIT CUR
6252 005026 062737 000002 000734 ADD #2, UNP ;BUMP POINTER
6253 005034 000137 004756 JMP RWND0 ;DO NEXT UNIT
6254 005040 005160 001012 RWND2: COM UN1(RO)
6255 005044 005037 000734 CLR UNP ;CLEAR POINTER
6256 005050 013700 000734 RWND3: MOV UNP, RO ;POINT TO UNIT ENTRY
6257 005054 005160 001012 COM UN1(RO) ;SEE IF LAST ENTRY:
6258 005060 001452 BEQ RWNDX ;IF SO: BR
6259 005062 005160 001012 COM UN1(RO)
6260 005066 016037 001012 000616 MOV UN1(RO), UDES ;SET UNIT DESCRIPTION
6261 005074 032737 000200 000616 BIT #200, UDES ;SEE IF UNIT IS DROPPED
6262 005102 001403 BEQ 1$ ;IF NOT: BR
6263 005104 005337 004716 DEC REOTC ;ELSE DECREMENT EOT UNIT CNTR
6264 005110 000417 BR RWND5
6265 005112 013777 000616 173462 1$: MOV UDES, @MTC ;LOAD COMMAND REGISTER
6266 005120 032777 000002 173452 RWND4: BIT #2, @MTC
6267 005126 001374 BNE RWND4 ;AWAIT RWS RESET
6268 005130 032777 000040 173442 BIT #40, @MTC ;SEE IF HAVE BOT
6269 005136 001411 BEQ RWND6 ;IF NOT: BR
6270 005140 032777 000010 173432 1$: BIT #10, @MTC ;SEE IF SDWN SET
6271 005146 001374 BNE 1$ ;IF SO AWAIT RESET
6272 005150 062737 000002 000734 RWND5: ADD #2, UNP ;BUMP POINTER
6273 005156 000137 005050 JMP RWND3 ;DO NEXT UNIT
6274 005162 012700 000001 RWND6: MOV #1, RO
6275 005166 004737 017506 JSR PC, PAPRT ;PRINT HEADER
6276 005172 012704 023653 MOV #MSG48, R4
6277 005176 004737 020530 JSR PC, TTOUT ;PRINT NO BOT
6278 005202 000137 020050 JMP DRPDRV ;GO DROP DRIVE
6279 005206 005160 001012 RWNDX: COM UN1(RO)
6280 005212 005000 CLR RO
6281 005214 010037 000734 1$: MOV RO, UNP

```


F04

6282	005220	016037	001012	000616	MOV	UNI(RO).UDES	
6283	005226	032737	100200	000616	BIT	#100200.UDES	:SEE IF UNIT DROPPED
6284	005234	001403			BEQ	2\$:IF NOT: BR
6285	005236	062700	000002		ADD	#2,RO	
6286	005242	000764			BR	1\$	
6287	005244	012737	000001	000720	MOV	#1,BLCNTR	
6288	005252	000207			RTS	PC	

6290
6291
6292
6293
6294
6295
6296
6297
6298
6299
6300
6301
6302
6303
6304
6305
6306
6307
6308
6309
6310
6311

```

*****
WRITE ROUTINE:
THIS ROUTINE IS USED TO WRITE ONTO TAPE THE BLOCK
OF DATA DESCRIBED BY THE OPERATOR AND SET UP
IN THE SEQUENCE FORMATTER. THE TAPE UNIT TO BE USED
HAS BEEN ASSIGNED BY THE SEQUENCE FORMATTER AND
ITS PARAMETERS SET IN A UNIT DESCRIPTION WORD.
AS EACH RECORD OF THE BLOCK IS WRITTEN, IT IS CHECKED
FOR STATUS ERRORS, WORD COUNT ZERO, AND CORRECT CURRENT
MEMORY ADDRESS. IF THE WRITE OPERATION CAUSES THE SELECTED
UNIT TO REACH END OF TAPE (EOT), THE UNIT IS REWOUND
AND FLAGGED AS UNAVAILABLE FOR TESTING UNTIL ALL AVAILABLE
UNITS HAVE REACHED EOT AT WHICH TIME ALL UNITS WILL
BE RESTARTED AT A BLOCK COUNT OF ONE (1).
ERROR CHECKING MAY BE DISALLOWED VIA CONSOLE SWITCH
TWELVE (12).
WRITING TO TAPE MAY BE DISALLOWED VIA CONSOLE SWITCH
ZERO (0).
*****

```

```

6312 005254 032777 000001 173370 WRITE: BIT #1,JSWR ;SEE IF SHOULD WRITE
6313 005262 001076 BNE W3AO ;IF NOT: BR
6314 005264 012737 022452 000716 MOV #MSG5,EMADDR ;SET ERROR MSG ADDRESS
6315 005272 005077 173304 CLR @MTC
6316 005276 005077 173276 CLR @MTS
6317 005302 005037 000760 CLR EOTREC ;CLEAR EOT FLAG
6318 005306 013700 000620 MOV RCNT,R0 ;R0=RECORD COUNT
6319 005312 013777 000622 173264 W0: MOV CARCNT,@MWC ;LOAD CHAR COUNT
6320 005320 012777 026176 173260 MOV #WDATA,@MDA ;SET DATA ADDR
6321 005326 005737 000764 TST ERTFL ;SEE IF SHOULD ERASE
6322 005332 001406 BEQ W0A ;IF NOT: BR
6323 005334 112777 000014 173240 MOVB #14,@MTC ;SET OP-CODE: WRITE W/EXTENDED IRG
6324 005342 005037 000764 CLR ERTFL ;CLEAR ERASE FLAG
6325 005346 000403 BR W0B
6326 005350 112777 000004 173224 W0A: MOVB #4,@MTC ;SET WRITE OP COMMAND
6327 005356 012737 005370 000724 W0B: MOV #W1A,RTRN ;SET RETURN ADDRESS
6328 005364 000137 017006 JMP TAPG ;GO EXECUTE COMMAND
6329 005370 005737 017402 W1A: TST WEOTF ;SEE IF EOT FOUND
6330 005374 001413 BEQ W1 ;IF NOT: BR
6331 005376 005037 017402 CLR WEOTF ;CLEAR WRITE EOT FLAG
6332 005402 013701 000620 MOV RCNT,R1 ;BUILD SHORTENED RECORD COUNT
6333 005406 160001 SUB R0,R1
6334 005410 005201 INC R1
6335 005412 010137 000760 MOV R1,EOTREC
6336 005416 052737 100000 000760 BIS #10000,EOTREC ;SET EOT FLAG
6337 005424 032777 010000 173220 W1: BIT #10000,@JSWR ;SEE IF SHOULD CHECK ERRORS
6338 005432 001002 BNE W3 ;IF NOT: BR
6339 005434 004737 016036 JSR PC,ERCHK ;GO CHECK ERRORS
6340 005440 013737 000634 000730 W3: MOV #STAL,STAL ;SET DELAY
6341 005446 004737 010634 JSR PC,STALL ;DELAY
6342 005452 005737 000754 TST RTYFL ;SEE IF RETRY
6343 005456 001401 BEQ W3A ;IF NOT: BR
6344 005460 000207 W3A: RTS PC ;ELSE RETURN TO RETRY ROUTINE
6345 005462 005737 000742 W3A: TST SERFL ;SEE IF WRITE ERROR

```

H04

6346	005466	001453		BEQ	W3D	;IF NOT: BR
6347	005470	013704	000734	MOV	UNP,R4	;BUMP WRITE ERROR
6348	005474	005264	001074	INC	WTER1(R4)	
6349	005500	005037	000742	CLR	SERFL	;CLEAR STATUS ERROR FLAG
6350	005504	032777	000002	BIT	#2,@SWR	;SEE IF RETRY -- SW1
6351	005512	001041		BNE	W3D	;IF NOT: BR
6352	005514	042737	072521	BIC	#072521,ERSAV	;MASK UNRECOVERABLE ERROR
6353	005522	005737	000740	TST	ERSAV	;SEE IF RETRYABLE ERROR
6354	005526	001411		BEQ	W3B	;IF SO: BR
6355	005530	012704	023733	MOV	#MSG52,R4	
6356	005534	004737	020530	JSR	PC,TTOUT	;PRINT NON-RETRYABLE ERROR FLAG
6357	005540	012704	022452	MOV	#MSG5,R4	
6358	005544	004737	020530	JSR	PC,TTOUT	;PRINT WRITE ERROR TAG
6359	005550	000422		BR	W3D	
6360	005552	013704	000734	MOV	UNP,R4	
6361	005556	005264	001154	INC	RTY1(R4)	;BUMP RETRY CNTR
6362	005562	032777	002000	BIT	#2000,@SWR	;SEE IF PRINT ERRORS
6363	005570	001004		BNE	W3C	;IF NOT: BR
6364	005572	012704	023753	MOV	#MSG53,R4	
6365	005576	004737	020530	JSR	PC,TTOUT	;PRINT ORIGINAL ERROR TAG
6366	005602	005037	000752	CLR	RTCNT	;CLEAR RETRY NUMBER
6367	005606	005037	000750	CLR	RPCNT	;CLEAR REPEAT COUNTER
6368	005612	004737	006142	JSR	PC,WRTY	;GO RETRY WRITE ERROR
6369	005616	005037	000754	CLR	RTYFL	;CLEAR RETRY FLAG
6370	005622	005737	000760	TST	EOTREC	;WAS EOT REACHED?
6371	005626	100403		BMI	WEX	;IF SO: BR
6372	005630	005300		DEC	RO	;SEE IF DONE ALL
6373	005632	001227		BNE	W0	;IF NOT: BR
6374	005634	005200		INC	RO	;ADJUST FOR REC NO. IN HEADER
6375	005636	005737	000646	TST	TMEX	;SEE IF TM
6376	005642	001402		BEQ	WEX1	;IF NOT: BR
6377	005644	004737	005712	JSR	PC,WTM	;WRITE TM
6378	005650	005037	000754	CLR	RTYFL	;CLEAR RETRY FLAG
6379	005654	005037	000756	CLR	TMFLG	;CLEAR TM FLAG
6380	005660	005737	000760	TST	EOTREC	;TEST FOR EOT
6381	005664	100401		BMI	W4	;IF SO: BR
6382	005666	000207		RTS	PC	;EXIT
6383	005670	017704	172756	MOV	@SWR,R4	
6384	005674	042704	177767	BIC	#177767,R4	;CHECK IF WRITE ONLY
6385	005700	022704	000010	CMP	#10,R4	
6386	005704	001370		BNE	WEX2	;IF NOT: BR
6387	005706	000137	004142	JMP	REOT	;GO REWIND ALL AVAIL TAPES
6388						

6390
6391
6392
6393
6394
6395
6396
6397
6398
6399
6400
6401
6402
6403
6404
6405
6406
6407
6408
6409
6410
6411
6412
6413
6414
6415
6416
6417
6418
6419
6420
6421
6422
6423
6424
6425
6426
6427
6428
6429
6430
6431
6432
6433
6434
6435
6436
6437
6438
6439
6440

005712 012737 024455 000716 WTM:
005720 005300
005722 005237 000756
005726 005077 172652
005732 012777 026176 172646
005740 012777 000006 172634
005746 012737 005760 000724
005754 000137 017006
005760 032777 010000 172664 WTM0:
005766 001062
005770 004737 016036
005774 005737 000742
006000 001455
006002 005737 000754
006006 001401
006010 000207
006012 013704 000734 WTM1:
006016 005264 001074
006022 032777 000002 172622
006030 001041
006032 042737 147377 000740
006040 005737 000740
006044 001411
006046 012704 023733
006052 004737 020530
006056 012704 024455
006062 004737 020530
006066 000207
006070 005037 000750 WTM2:
006074 013704 000734
006100 005264 001154
006104 005037 000752
006110 032777 002000 172534
006116 001004
006120 012704 023753
006124 004737 020530
006130 004737 006142 WTM3:
006134 005037 000756 WTM4:
006140 000207

```
*****  
:WRITE TAPE MARK  
:THIS ROUTINE, ENABLED THRU TELETYPE RESPONSE  
:AT PROGRAM START-UP, WILL WRITE A TAPE MARK  
:FOLLOWING THE WRITING OF EACH BLOCK OF DATA.  
:THIS OPTION INCREASES THE BLOCK SIZE BY ONE RECORD;  
:A BLOCK OF 100 RECORDS WILL HAVE A TAPE MARK  
:WRITTEN AS RECORD 101.  
*****  
;POINT TO TM ERROR MSG  
;SET TM FLAG  
;CLEAR BYTE COUNTER  
;SET TM OP CODE  
;SAVE RETURN ADDRESS  
;EXECUTE TM COMMAND  
;SEE IF SHOULD CHECK ERRORS  
;IF NOT: BR  
;CHECK FOR ERRORS  
;SEE IF STATUS ERROR  
;IF NOT: BR  
;SEE IF RETRY  
;IF NOT: BR  
;ELSE RETURN TO RETRY ROUTINE  
;BUMP WRITE ERROR  
;SEE IF SHOULD RETRY  
;IF NOT: BR  
;MASK UNRECOVERABLE ERROR  
;SEE IF RECOVERABLE  
;IF SO: BR  
;PRINT UNRETRYABLE TAG  
;PRINT TM ERROR TAG  
;CLEAR REPEAT CNTR  
;BUMP RETRY CNTR  
;CLEAR RETRY LOOP CNTR  
;SEE IF PRINT ERRORS  
;IF NOT: BR  
;PRINT ORIGINAL ERROR TAG  
;GO DO RETRY  
;CLEAR TM FLAG  
;EXIT
```

```

6442 ;*****
6443 ;WRITE ERROR RETRY
6444 ;
6445 ;*****
6446
6447 006142 012737 000001 000754 WRTY: MOV #1,RTYFL ;SET RETRY FLAG
6448 006150 004737 006504 WRTY0: JSR PC,WRTSB ;GO SPACE BACK FOR REPEAT
6449 006154 005737 000756 TST TMFLG ;SEE IF A TM
6450 006160 001003 BNE WRTYTM ;IF SO: BR
6451 006162 004737 005312 JSR PC,W0 ;REWRITE RECORD
6452 006166 000402 BR WRTYR ;CONTINUE
6453 006170 004737 005712 WRTYTM: JSR PC,WTM ;GO WRITE TM AGAIN
6454 006174 005737 000742 WRTYR: TST SERFL ;REWRITE GOOD?
6455 006200 001027 BNE WRTY2 ;IF NOT: BR
6456 006202 005237 000750 INC RPCNT ;BUMP REPEAT COUNTER
6457 006206 022737 000004 000750 CMP #4,RPCNT ;SEE IF FOUR GOOD REPEATS
6458 006214 001355 BNE WRTY0 ;IF NOT: DO ANOTHER
6459 006216 032777 002000 172426 BIT #2000,JSWR ;SEE IF PRINT
6460 006224 001014 BNE WRTY1 ;IF NOT: BR
6461 006226 012704 024001 MOV #MSG54,R4
6462 006232 004737 020530 JSR PC,TTOUT ;PRINT RECOVERED MESSAGE
6463 006236 012704 024014 MOV #MSG55,R4
6464 006242 004737 020530 JSR PC,TTOUT ;PRINT RETRY TAG
6465 006246 013703 000752 MOV RTCNT,R3
6466 006252 004737 020716 JSR PC,OC1P ;PRINT RETRY NUMBER
6467 006256 000207 WRTY1: RTS PC ;RESUME TESTING
6468 006260 032777 002000 172364 WRTY2: BIT #2000,JSWR ;SEE IF PRINT
6469 006266 001024 BNE WRTY3 ;IF NOT: BR
6470 006270 012704 024025 MOV #MSG56,R4
6471 006274 004737 020530 JSR PC,TTOUT ;PRINT BAD TAPE SUSPECT
6472 006300 012704 024014 MOV #MSG55,R4
6473 006304 004737 020530 JSR PC,TTOUT ;PRINT RETRY TAG
6474 006310 013703 000752 MOV RTCNT,R3
6475 006314 004737 020716 JSR PC,OC1P ;PRINT RETRY NUMBER
6476 006320 012704 024047 MOV #MSG57,R4
6477 006324 004737 020530 JSR PC,TTOUT ;PRINT REPEAT TAG
6478 006330 013703 000750 MOV RPCNT,R3
6479 006334 004737 020716 JSR PC,OC1P ;PRINT REPEAT NUMBER
6480 006340 005737 000752 WRTY3: TST RTCNT ;SEE IF FIRST RETRY
6481 006344 001004 BNE WRTY3A ;IF NOT: BR
6482 006346 013704 000734 MOV UNP,R4
6483 006352 005364 001074 DEC WERR1(R4) ;DECREMENT WRITE ERROR CNTR
6484 006356 013704 000734 WRTY3A: MOV UNP,R4 ;GET UNIT NUMBER
6485 006362 016437 002714 000762 MOV BTADDR(R4),BTPT ;GET ADDRESS OF UNIT BAD TAPE CNTR
6486 006370 017704 172366 MOV BTPT,R4 ;GET COUNTER
6487 006374 005724 TST (R4)+ ;SET POINTER OFFSET
6488 006376 010477 172360 MOV R4,BTPT
6489 006402 013703 000762 MOV BTPT,R3
6490 006406 060304 ADD R3,R4 ;SET ABSOLUTE POINTER
6491 006410 013714 000720 MOV BLCNTR,(R4) ;SET BLOCK NUMBER
6492 006414 062704 000040 ADD #40,R4 ;ADD RCNT OFFSET
6493 006420 013714 000620 MOV RCNT,(R4)
6494 006424 160014 SUB RO,(R4) ;SET RECORD NUMBER
6495 006426 005214 INC (R4) ;CORRECT RECORD NUMBER
6496 006430 022777 000040 172324 CMP #40,BTPT ;SEE IF TOO MANY BAD SPOTS
6497 006436 001002 BNE WRTY4 ;IF NOT: BR

```

K04

```

6498 006440 000137 006570          JMP      BTOV          ;ELSE GO TO BAD TAPE OVERFLOW
6499 006444 005237 000752          WRTY4: INC      RTCNT      ;BUMP RETRY COUNTER
6500 006450 022737 000004 000752    CMP      #4,RTCNT      ;SEE IF DONE 4 RETRIES
6501 006456 001410                    BEQ      WRTY5          ;IF SO: BR
6502 006460 013704 000734          MOV      UNP,R4
6503 006464 005264 001154          INC      RTY1(R4)      ;BUMP RETRY COUNTER
6504 006470 005237 000764          INC      ERTFL         ;SET ERASE FLAG
6505 006474 000137 006150          JMP      WRTY0         ;DO NEXT RETRY
6506 006500 000137 006614          WRTY5: JMP      BTUR          ;ELSE GO TO BAD TAPE UNRECOVERABLE
6507
6508                                     ;WRITE RETRY BACKSPACE-ERASE SUBROUTINE
6509
6510 006504 005037 000742          WRTSB: CLR      SERFL          ;CLEAR FLAG
6511 006510 012777 177777 172066    MOV      #-1,AMWC      ;SET FOR 1 RECORD
6512 006516 012737 024551 000716    MOV      #MSG69,EMADDR
6513 006524 004737 010466          JSR      PC,SPBK       ;DO SPACE BACK
6514 006530 012737 022452 000716    MOV      #MSG5,EMADDR
6515 006536 032737 000002 000746    JTY      #2,BTFLG      ;SEE IF ERROR ON BACKSPACE
6516 006544 001410                    BEQ      WRTSBO        ;IF NOT: BR
6517 006546 005037 000754          CLR      RTYFL
6518 006552 022626                    CMP      (SP)+,(SP)+   ;RESET STACK
6519 006554 052737 000004 000746    BIS      #4,BTFLG      ;MARK RETRY ERROR
6520 006562 000137 004142          JMP      REOT          ;REWIND AND REMOVE FROM TESTING
6521 006566 000207          WRTSBO: RTS      PC          ;RETURN
6522
6523                                     ;BAD TAPE OVERFLOW SUBROUTINE*****
6524
6525 006570 013704 000734          BTOV:  MOV      UNP,R4
6526 006574 005264 001154          INC      RTY1(R4)      ;BUMP RETRY COUNTER
6527 006600 012737 000001 000746    MOV      #1,BTFLG      ;SET BAD TAPE OVERFLOW FLAG
6528 006606 005726                    TST      (SP)+         ;RESET STACK
6529 006610 000137 004142          JMP      REOT          ;GO REWIND AND REMOVE FROM TESTING
6530
6531                                     ;BAD TAPE UNRECOVERABLE SUBROUTINE*****
6532
6533 006614 012704 024061          BTUR:  MOV      #MSG58,R4
6534 006620 004737 020530          JSR      PC,TTOUT      ;PRINT UNRECOVERABLE BAD SPOT MSG
6535 006624 000207          RTS      PC          ;RESUME TESTING

```

6537
6538
6539
6540
6541
6542
6543
6544
6545
6546
6547
6548
6549
6550
6551
6552
6553
6554
6555
6556
6557
6558
6559
6560
6561
6562
6563
6564
6565
6566
6567

```
*****  
; READ SEQUENCER:  
; THIS ROUTINE IS USED TO DETERMINE THE SEQUENCE  
; IN WHICH READ TAPE OPERATIONS ARE TO BE PERFORMED.  
; SWITCH THREE (3) DISALLOWS READING.  
; IF THE PROGRAM IS BEING RUN IN THE READ ONLY MODE,  
; CONSOLE SWITCH ZERO (0) SET TO A ONE (1), THEN SETTING  
; CONSOLE SWITCH FOURTEEN (14) WILL CAUSE READING OF  
; THE SAME BLOCK OF DATA CONTINUOUSLY,  
; WHEN SET TO A ONE (1), AND ALLOW TAPE  
; TO READ BLOCKS PROGRESSIVELY WHEN SET TO A ZERO (0).  
*****
```

```
006626 032777 000010 172016 RSEQ: BIT #10, QSWR ; SEE IF SHOULD READ FORWARD  
006634 001031 BNE RSEX ; IF NOT: BR  
006636 032777 000001 172006 BIT #1, QSWR ; SEE IF WRITE  
006644 001404 BEQ RSFROA ; IF SO: BR  
006646 032777 040000 171776 BIT #40000, QSWR ; SEE IF SHOULD REMAIN IN PLACE  
006654 001410 BEQ RSFRO ; IF NOT: BR  
006656 004737 010132 RSFROA: JSR PC, BKSP ; GO BACKSPACE TO START  
006662 032737 000002 000746 BIT #2, BTFLG ; ERROR ON BACKSPACE?  
006670 001402 BEQ RSFRO ; IF NOT: BR  
006672 000137 004142 JMP REOT ; REWIND AND REMOVE FROM TESTING  
006676 012737 000002 000626 RSFRO: MOV #2, RUCMD ; LOAD READ FORWARD COMMAND  
006704 004737 006722 JSR PC, READ ; GO READ FORWARD  
006710 032777 040000 171734 BIT #40000, QSWR ; SEE IF SHOULD READ SAME BLOCK  
006716 001357 BNE RSFROA ; IF SO: BR  
006720 000207 RSEX: RTS PC ; EXIT
```

6569
6570
6571
6572
6573
6574
6575
6576
6577
6578
6579
6580
6581
6582
6583
6584
6585
6586
6587
6588
6589
6590
6591
6592
6593
6594
6595
6596
6597
6598
6599
6600
6601
6602
6603
6604
6605
6606
6607
6608
6609
6610
6611
6612
6613
6614
6615
6616
6617
6618
6619
6620
6621
6622
6623
6624

006722 013700 000620
006726 012737 022457 000716
006734 005037 000756
006740 052777 040000 171644
006746 005077 171630
006752 005077 171622
006756 013777 000622 171620
006764 012777 032210 171614
006772 053777 000626 171602
007000 012737 007012 000724
007006 000137 017006
007012 032777 002000 171560
007020 001405
007022 052737 100000 000760
007030 000137 007350
007034 032777 000040 171536
007042 001411
007044 004737 017506
007050 012704 022725
007054 004737 020530
007060 000240
007062 000137 020050
007066 032777 004000 171556
007074 001037
007076 004737 016036
007102 005737 000742
007106 001432
007110 013704 000734
007114 005264 001114
007120 032777 000002 171524
007126 001022
007130 017737 171444 000740
007136 042737 073525 000740
007144 001411
007146 012704 C23733
007152 004737 020530

READ: MOV RCNT, R0 ;LOAD REC CNTR
MOV #MSG6, EMADDR ;SET ERROR MSG ADDRESS
CLR TMFLG ;CLEAR TM FLAG
BIS #40000, AMTRD ;SET TO READ LPC ON READ
RDO: CLR AMTC
CLR AMTS
RD1: MOV CARCNT, AMWC ;LOAD CHAR CNTR
RD1A: MOV #RDATA, AMDA ;LOAD DATA ADDR
BIS RDCMD, AMTC ;LOAD READ OP COMMAND
MOV #RD2, ATRN ;SET INTERRUPT RETURN ADDRESS
JMP TAPG ;GO EXECUTE TAPE COMMAND
RD2: BIT #2000, AMTS ;SEE IF AT EOT
BEQ RD3 ;IF NOT: BR
BIS #100000, EOTREC ;MARK EOT FOUND
JMP RDEX ;GO REWIND
RD3: BIT #40, AMTS ;SEE IF AT LOAD POINT
BEQ RD4 ;IF NOT: BR
JSR PC, PAPRT ;PRINT CYCLE NUMBER
MOV #MSG22, R4
JSR PC, TOUT ;PRINT BOT ERROR
NOP
JMP DRPDRV ;DROP DRIVE
RD4: BIT #4000, ASWR ;SEE IF SHOULD CHECK ERRORS
BNE R05 ;IF NOT: BR
JSR PC, ERCHK ;GO CHECK ERRORS
TST SERFL ;SEE IF STATUS ERROR
BEQ R05 ;IF NOT: BR
MOV UNP, R4
INC RDER1(R4) ;BUMP READ ERROR
BIT #2, ASWR ;SEE IF SHOULD DO READ RETRY
BNE R05 ;IF NOT: BR
MOV AMTS, ERSV
BIC #073525, ERSV ;MASK NON-RETRYABLE ERRORS
BEQ RD4A ;IF RETRYABLE: BR
MOV #MSG52, R4
JSR PC, TOUT ;PRINT NON-RETRYABLE MESSG.

;READ ROUTINE:
;THIS ROUTINE PERFORMS THE READ OPERATION DETERMINED
;BY THE READ SEQUENCE ROUTINE ONE RECORD AT A TIME.
;AT THE END OF EACH READ OPERATION THE STATUS REGISTER
;IS SCANNED FOR EITHER END OF TAPE OR BEGINNING OF TAPE.
;IF EOT WAS REACHED, CONTROL WILL BE PASSED TO
;THE EOT SUBROUTINE TO REWIND THE UNIT AND FLAG IT
;UNAVAILABLE UNTIL ALL UNITS HAVE REACHED EOT.
;IF BOT WAS REACHED AND ERROR IS PRINTED AND THE
;PROGRAM WILL HALT. TESTING MAY BE RESUMED BY PRESSING
;THE CONTINUE SWITCH TWICE.
;CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13) DETERMINE WHETHER
;OR NOT TO CHECK FOR STATUS ERRORS (11) OR DATA ERRORS (13),
;CONSOLE SWITCH FIVE (5) IS USED TO CAUSE A CONTINUOUS
;READ AND SPACE (FORWARD OR REVERSE) OF THE CURRENT
;RECORD ON TAPE (YOZZLE).

```

6625 007156 012704 022457      MOV      #MSG6,R4
6626 007162 004737 020530      JSR      PC,TTOUT      ;PRINT READ ERROR TAG
6627 007166 000402          BR       RDS
6628 007170 004737 007376      JSR      PC,RRTY      ;DO RETRY
6629 007174 032777 020000 171450 RDS:    BIT      #20000,JSWR   ;SEE IF SHOULD DO DATA CHECK
6630 007202 001007          BNE     RD6           ;IF NOT: BR
6631 007204 005737 000756      TST     TMFLG        ;IS IT TM?
6632 007210 001004          BNE     RD6           ;IF SO: BR
6633 007212 004737 013712      JSR      PC,DCHK      ;GO CHECK DATA
6634 007216 005037 000742      CLR     SERFL        ;CLEAR STATUS ERROR FLAG
6635 007222 004737 012516      JSR      PC,DS3       ;CLEAR BUFFER
6636 007226 032777 000040 171416 RDS:    BIT      #40,JSWR    ;SEE IF SHOULD YOZZLE
6637 007234 001402          BEQ     RD7           ;IF NOT: BR
6638 007236 004737 007600      JSR      PC,YOZ       ;ELSE GO YOZZLE
6639 007242 013737 000632 000730 RD7:    MOV      RSTAL,STAL   ;SET DELAY
6640 007250 004737 010634      JSR      PC,STALL    ;STALL
6641 007254 005737 000756      TST     TMFLG        ;JUST DONE TM?
6642 007260 001033          BNE     RDEX         ;IF SO: BR
6643 007262 005737 000760      TST     EOTREC       ;WAS EOT REACHED
6644 007266 100430          BMI     RDEX         ;IF SO: BR
6645 007270 005300          DEC     RD
6646 007272 001225          BNE     RD0
6647 007274 005200          INC     RD
6648 007276 005737 000646      RD10:   TST     TMEX
6649 007302 001422          BEQ     RDEX         ;IF NOT: BR
6650 007304 005300          DEC     RD           ;ELSE READ TM
6651 007306 012777 177776 171270 MOV      #-2,AMWC     ;SET BYTE COUNT
6652 007314 005737 001010      TST     STCDFL       ;SEE IF 7 TRK CORE DUMP
6653 007320 001402          BEQ     1$           ;IF NOT: BR
6654 007322 005277 171256      INC     AMWC         ;SET TO ONE CHAR
6655 007326 005237 000756      1$:     INC     TMFLG    ;SET TM FLAG
6656 007332 012737 024561 000716 MOV      #MSG70,EMADDR
6657 007340 042777 040000 171244 BIC     #40000,AMTRD ;SET TO READ LPC ON READ TM
6658 007346 000606          BR      RD1A         ;GO READ
6659 007350 005037 000756      RDEX:   CLR     TMFLG
6660 007354 005737 000760      TST     EOTREC       ;WAS EOT REACHED
6661 007360 100005          BPL     RDEXX        ;IF NOT: BR
6662 007362 005726          TST     (SP)+        ;RESET STACK
6663 007364 005037 000760      CLR     EOTREC       ;CLEAR EOT IND.
6664 007370 000137 004142      JMP     REOT         ;GO REWIND
6665 007374 000207      RDEXX: RTS     PC     ;EXIT
6666
6667
6668
6669
6670
6671
6672
6673
6674 007376 005237 000774      RRTY:   INC     RRTYFL  ;SET READ RETRY FLAG
6675 007402 032777 002000 171242 BIT      #2000,JSWR   ;SEE IF PRINT?
6676 007410 001004          BNE     RRTY0        ;IF NOT: BR
6677 007412 012704 023753      MOV      #MSG53,R4
6678 007416 004737 020530      JSR      PC,TTOUT    ;PRINT ORIGINAL ERROR MESSG
6679 007422 005037 000752      RRTY0: CLR     RRCNT    ;CLEAR RETRY COUNT
6680 007426 004737 007600      RRTY1: JSR      PC,YOZ     ;GO REREAD

```

```

;*****
;READ ERROR RETRY
;*****

```


6710
6711
6712
6713
6714
6715
6716
6717
6718
6719
6720
6721
6722
6723
6724
6725
6726
6727
6728
6729
6730
6731
6732
6733
6734
6735
6736
6737
6738
6739
6740
6741
6742
6743
6744
6745
6746
6747
6748
6749
6750
6751
6752
6753
6754
6755
6756
6757
6758
6759
6760
6761
6762
6763
6764
6765

007600 012777 000001 171050
007606 013737 000640 000730
007614 004737 010634
007620 012777 177777 170756
007626 112777 000012 170746
007634 012737 007654 000724
007642 012737 177775 000730
007650 000137 017006
007654 013737 000640 000730
007662 004737 010634
007666 113777 000626 170706
007674 012777 032210 170704
007702 013777 000622 170674
007710 005737 000756
007714 001410
007716 012777 177776 170660
007724 005737 001010
007730 001402
007732 005277 170646
007736 012737 007750 000724
007744 000137 017006
007750 032777 004000 170674
007756 001002
007760 004737 016036
007764 005737 000774
007770 001401
007772 000207
007774 032777 020000 170650
010002 001002
010004 004737 013712
010010 004737 012516
010014 105777 170636
010020 100034
010022 122777 000203 170630
010030 001030
010032 012704 023600
010036 004737 020530
010042 013703 000640
010046 004737 020716
010052 010037 000712
010056 012705 000640
010062 012701 000006

```
*****  
;YOZZLE SUBROUTINE:  
*****  
;THIS SUBROUTINE, ENTERED VIA SWITCH FIVE (5), IS USED TO PERFORM  
;A CONTINUOUS READ AND SPACE OVER OF THE CURRENT RECORD ON TAPE.  
;FULL STATUS AND DATA CHECKING MAY BE PERFORMED  
;OR NOT VIA CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13).  
;A SOFTWARE DELAY IS PERFORMED BETWEEN EACH READ  
;AND SPACE OPERATION AND MAY BE VARIED BY TYPING  
;CNTRL C ON THE TTY AND ENTERING A VALUE IN RESPONSE  
;TO THE PRINTED REQUEST.  
*****  
YOZ:  MOV #1,ATKS ;SET TTY ENABLE  
      MOV YSTAL,STAL  
      JSR PC,STALL ;DO YOZZLE STALL  
YOZO: MOV #-1,AMWC ;SET TO 1 RECORD SPACING  
YOZA: MOVB #12,AMTC ;SET TO SPACE REVERSE  
YOZB: MOV #YOZC,RTRN ;SET RETURN ADDRESS  
      MOV #177775,STAL ;SET TIME MULTIPLIER  
      JMP TAPG ;GO YOZZLE  
YOZC: MOV YSTAL,STAL ;DO YOZZLE STALL  
      JSR PC,STALL ;SET READ COMMAND F OR R  
      MOVB RDCMD,AMTC ;SET READ ADDRESS  
      MOV #RDATA,AMDA ;SET CHARACTER COUNT  
      MOV CARCNT,AMWC ;IS IT A TM?  
      TST TMFLG ;IF NOT: BR  
      BEQ YOZC1 ;SET FOR TM  
      MOV #-2,AMWC ;SEE IF 7 TRK CORE DUMP  
      TST STCDFL ;IF NOT: BR  
      BEQ YOZC1 ;SET TO ONE CHARACTER  
      INC AMWC ;SET RETURN ADDRESS  
YOZC1: MOV #YOZD,RTRN ;GO YOZZLE  
      JMP TAPG ;SEE IF ERROR CHECK  
YOZD: BIT #4000,ASWR ;IF NOT: BR  
      BNE YOZE ;ELSE GO CHECK ERRORS  
      JSR PC,ERCHK ;IS IT A READ RETRY?  
      TST RRTYFL ;IF NOT: BR  
      BEQ YOZE1  
      RTS PC  
YOZE1: BIT #20000,ASWR ;SEE IF SHOULD CHECK DATA  
      BNE YOZF ;IF NOT: BR  
      JSR PC,DCHK ;ELSE GO CHECK DATA  
YOZF: JSR PC,DS3 ;GO CLEAR DATA AREA  
      TSTB ATKS ;SEE IF HAVE NEW STALL VALUE  
      BPL YOZG ;IF NOT: BR  
      CMPB #203,ATKB ;SEE IF CONT C  
      BNE YOZG ;IF NOT: BR  
      MOV #MSG44,R4 ;PRINT YSTALL REQUEST  
      JSR PC,TTOUT  
      MOV YSTAL,R3 ;PRINT PRESENT STALL  
      JSR PC,OC1P ;SAVE RO(REC CNT)  
      MOV RO,TEMP3 ;SET ADDRESS OF YSTL  
      MOV #YSTAL,R5 ;SET NUMBER OF CHAR TO INPUT  
      MOV #6,R1
```

6766	010066	012702	177777			MOV	#-1, R2	;SET MAXIMUM LIMIT
6767	010072	012703	001000			MOV	#1000, R3	;SET MINIMUM LIMIT
6768	010076	004737	020272			JSR	PC, TTR	;GO GET VALUE
6769	010102	013700	000712			MOV	TEMP3, R0	;RESTORE R0(REC CNTR)
6770	010106	000137	007600			JMP	YOZ	;RESTART YOZZLE
6771	010112	032777	000040	170532	YOZG:	BIT	#40, @SWR	;SEE IF SHOULD CONTINUE YOZZLE
6772	010120	001227				BNE	YOZ	;IF SO: BR
6773	010122	012777	000100	170526		MOV	#100, @TKS	;SET TTY INTERRUPT ENABLE
6774	010130	000207				RTS	PC	;EXIT
6775								

```

6777
6778
6779
6780
6781
6782
6783
6784
6785
6786
6787
6788
6789
6790
6791
6792 010132 005037 000726 BKSP: CLR HDRFL ;CLEAR HEADER FLAG
6793 010136 013700 000620 BO: MOV RCNT,RO
6794 010142 005100 COM RO ;BUILD SPACE AMOUNT
6795 010144 005200 INC RO
6796 010146 005737 000760 TST EOTREC ;SEE IF EOT WAS REACHED
6797 010152 001407 BEQ BKO ;IF NOT: BR
6798 010154 013700 000760 MOV EOTREC,RO ;GET SHORTENED BLOCK COUNT
6799 010160 042700 100000 BIC #100000,RO
6800 010164 005400 NEG RO
6801 010166 005037 000760 CLR EOTREC ;CLEAR EOT FLAG
6802 010172 010037 000714 BKO: MOV RO,TEMP4 ;SAVE BACKSPACE COUNT
6803 010176 005737 000646 TST TMEX ;IS THERE A TM?
6804 010202 001520 BEQ BOA ;IF NOT: BR
6805 010204 012737 024464 000716 MOV #MSG63,EMADDR ;POINT TO TM SP ERROR MSG
6806 010212 012777 177777 170364 MOV #-1,@MWC ;SET FOR 1 RECORD
6807 010220 013700 000620 MOV RCNT,RO ;RO=RECORD COUNT
6808 010224 063700 000714 ADD TEMP4,RO ;RO=RCNT-BACKSPACE CNT FOR HEADER
6809 010230 004737 010466 JSR PC,SPBK ;BACKSPACE OVER TM
6810 010234 032737 000002 000746 BIT #2,BTFLG ;WAS THERE AN ERROR
6811 010242 001401 BEQ BK1 ;IF NOT: BR
6812 010244 000207 RTS PC ;ELSE RETURN WITH ERROR FLAG SET
6813 010246 017737 170326 000740 BK1: MOV @MTC,ERSAV ;GET STATUS
6814 010254 032737 040000 000740 BIT #40000,ERSAV ;IS TM SET
6815 010262 001053 BNE BK3 ;IF SET: BR
6816 010264 005737 000726 BK1C: TST HDRFL ;ALREADY PRINTED HEADER?
6817 010270 001014 BNE BK2 ;IF SO: BR
6818 010272 032777 002000 170352 BIT #2000,@SWR ;SHOULD PRINT?
6819 010300 001040 BNE BK1B ;IF NOT: BR
6820 010302 004737 017506 JSR PC,PAPRT ;PRINT HEADER
6821 010306 013704 000716 MOV EMADDR,R4 ;POINT TO TM SP ERROR
6822 010312 004737 020530 JSR PC,TTOUT ;PRINT ERROR
6823 010316 012704 022750 MOV #MSG23,R4
6824 010322 004737 020530 BK2: JSR PC,TTOUT ;PRINT COMMAND HEADER
6825 010326 017703 170250 MOV @MTC,R3
6826 010332 005037 000712 CLR TEMP3
6827 010336 000303 BK1A: SWAB R3 ;POSITION MOST SIGNIFICANT
6828 010340 004737 021144 JSR PC,DOUT ;PRINT
6829 010344 000303 SWAB R3 ;POSITION LEAST SIGNIFICANT
6830 010346 004737 JSR PC,DOUT ;PRINT
6831 010352 005737 000712 TST TEMP3 ;SEE IF PRINTED STATUS
6832 010356 001011 BNE BK1B ;IF SO: BR

```


6879
6880
6881
6882
6883
6884
6885
6886
6887
6888
6889
6890
6891
6892
6893
6894
6895
6896
6897
6898
6899
6900
6901
6902
6903
6904
6905
6906
6907
6908
6909
6910
6911
6912
6913
6914
6915
6916
6917
6918
6919
6920
6921
6922

010634 005337 000730
010640 001375
010642 000207

STALL:

DEC STAL ;
BNE STALL ;DELAY
RTS PC ;EXIT

;STALL ROUTINE:
;THIS ROUTINE IS USED TO PROVIDE SOFTWARE DELAYS
;DURING READ, WRITE, TURN AROUND, AND YOZZLE.
;THE DELAY TIMES MAY BE SET BY THE OPERATOR AT
;INITIAL START FROM 200(8) OR MAY BE MODIFIED
;AT ANY TIME BY ENTERING CNTRL C ON THE TTY AND
;INSERTING NEW VALUES IN RESPONSE TO THE REQUEST
;PRINTED.
;THE READ STALL AND THE WRITE STALL ARE DELAYS
;EXECUTED BETWEEN EACH RECORD OF THE DATA BLOCK.
;THE TURN AROUND STALL IS EXECUTED EACH TIME
;THE DIRECTION OF TAPE MOVEMENT IS CHANGED AND
;ALSO EACH TIME THE TAPE OPERATION CHANGES FROM
;WRITE TO READ OR READ TO WRITE.
;THE YOZZLE STALL IS EXECUTED ONLY DURING THE
;YOZZLE ROUTINE.

010644 012701 177760
010650 012702 174000
010654 004737 020240
010660 013737 000676 000622
010666 012737 177777 012750
010674 000207

CCNTR:

MOV #-20,R1 ;SET HIGH LIMIT
MOV #-4000,R2 ;SET LOW LIMIT
JSR PC,RANG ;GO GENERATE NUMBER
MOV RANSV, CARCNT ;SET CHAR COUNT
MOV #-1,PATS ;PRESET DATA PATTERN
RTS PC ;EXIT

;RANDOM CHARACTER COUNT GENERATOR:
;THIS ROUTINE ENTERED VIA CONSOLE SWITCH
;SEVEN (7) IS USED TO GENERATE A RANDOM
;CHARACTER COUNT FOR EACH DATA BLOCK.
;ALL RECORDS WITHIN A GIVEN BLOCK WILL BE
;THE SAME, BUT EACH BLOCK WILL VARY.
;THE LIMITS ARE TWENTY (20) TO TWO THOUSAND
;(2000) OCTAL CHARACTERS PER RECORD.

```

6924
6925
6926
6927
6929
6929
6930
6931
6932
6933
6934
6935 010676 012702 000001          RCNTR:
6936 010702 012701 000500          MOV
6937 010706 004737 020240          JSR
6938 010712 013737 000676 000620  MOV
6939 010720 000207          RTS
6940
6941
6942
6943
6944
6945
6946
6947
6948
6949
6950
6951
6952
6953
6954
6955
6956
6957
6958
6959
6960
6961
6962
6963
6964
6965
6966
6967
6968 010722 005737 000700          TINP:
6969 010726 001001          BNE
6970 010730 000207          RTS
6971 010732 005037 000734          TINPA:
6972 010736 005037 004716          CLR
6973 010742 012700 000010          CLR
6974 010746 012701 001012          MOV
6975 010752 005021          MOV
6976 010754 005300          TINPB:
6977 010756 001375          CLR
6978 010760 005737 021624          DEC
6979 010764 001405          BNE

```

```

;*****
;RANDOM RECORD COUNT GENERATOR:
;
;THIS ROUTINE ENTERED VIA CONSOLE SWITCH SIX (6)
;IS USED TO GENERATE A RANDOM NUMBER OF RECORDS
;FOR EACH BLOCK OF DATA.
;THE LIMITS ARE ONE (1) TO FIVE HUNDRED (500) OCTAL
;RECORDS PER BLOCK.
;*****
;SET LOW LIMIT
;SET HIGH LIMIT
;GO GENERATE NUMBER
;SET RECORD COUNT
;EXIT

;*****
;TEST CONDITION ENTRY ROUTINE:
;
;THIS ROUTINE IS USED TO ALLOW THE OPERATOR
;TO ENTER, AT THE TTY, THE NECESSARY PARAMETERS
;TO RUN THE PROGRAM AS HE WISHES. THE
;ROUTINE IS ONLY ENTERED UPON INITIAL STARTING
;FROM LOCATION 200(8).
;THE MAIN PURPOSE OF THIS ROUTINE IS TO ESTABLISH
;A TABLE OF DEVICES TO BE TESTED. THIS TABLE
;CONSISTS OF AN ENTRY FOR EACH OF ONE (1) TO
;EIGHT (8) DEVICES. EACH ENTRY CONTAINS THE
;DEVICE UNIT NUMBER, DENSITY, PARITY, AND
;NUMBER OF TRACKS. THE INFORMATION IS ENTERED
;IN RESPONSE TO PRINTED REQUESTS AT THE TTY.
;UNITS MAY BE ENTERED IN ANY ORDER. EACH
;PARAMETER IS CHECK FOR LEGALITY BEFORE BEING
;SET INTO THE TABLE.
;UPON COMPLETION OF THE DEVICE TABLE, REQUESTS
;ARE PRINTED FOR ENTRY OF THE NUMBER OF CHARACTERS
;PER RECORD AND THE NUMBER OF RECORDS PER BLOCK
;NEXT REQUEST IS FOR A PATTERN NUMBER TO BE USED
;FOR WRITING AND CHECKING OF READ DATA.
;THE LAST REQUESTS ARE FOR ENTRY OF THE DESIRED
;WRITE, READ, AND TURN AROUND STALLS.
;*****
;SEE IF SHOULD INPUT FROM TTY
;IF SO: BR
;EXIT
;CLEAR TABLE POINTER
;CLEAR EOT UNIT COUNTER
;SET SIZE OF TABLE
;SET START OF TABLE
;CLEAR TABLE
;SEE IF DONE
;IF NOT: BR
;SEE IF AUTO SEQUENCE
;IF NOT: BR

```


6980	010766	012704	024731		MOV	#MSG77,R4	
6981	010772	004737	020530		JSR	PC,TTOUT	;PRINT AUTO SEQ PROGRAM NAME
6982	010776	000410			BR	TINPO	
6983	011000	012704	023144		TINPB1: MOV	#MSG31,R4	
6984	011004	004737	020530		JSR	PC,TTOUT	;PRINT PROGRAM NAME
6985	011010	012704	023261		MOV	#MSG31A,R4	
6986	011014	004737	020530		JSR	PC,TTOUT	;PRINT REST OF TITLE
6987	011020	122737	000004	000041	TINPO: CMPB	#4,#41	;SEE IF LOAD MEDIUM
6988	011026	001006			BNE	1\$;IF NOT: BR
6989	011030	012704	026116		MOV	#MSG97,R4	
6990	011034	004737	020530		JSR	PC,TTOUT	;ELSE PRINT NO TEST
6991	011040	000137	004662		JMP	REOT10	;END TEST
6992	011044	012704	025253		1\$: MOV	#MSG84,R4	
6993	011050	004737	020530		JSR	PC,TTOUT	;REQUEST STARTING REGISTER ADDRESS
6994	011054	013703	000600		MOV	MTS,R3	
6995	011060	004737	020716		JSR	PC,OTCP	;PRINT CURRINT REGISTER START
6996	011064	013705	000674		MOV	REGST,R5	;SAVE ADDRESS LOCATION
6997	011070	012701	000006		MOV	#6,R1	;SET SIZE OF ENTRY
6998	011074	012702	177770		MOV	#177770,R2	;SET UPPER LIMIT
6999	011100	012703	170000		MOV	#170000,R3	;SET LOWER LIMIT
7000	011104	004737	020272		JSR	PC,TTR	;GO GET RESPONSE
7001	011110	012705	000602		MOV	#MTC,R5	;SET TABLE BASE
7002	011114	013704	000600		MOV	MTS,R4	;GET INITIAL ADDRESS
7003	011120	062704	000002		2\$: ADD	#2,R4	;BUMP ADDRESS
7004	011124	010425			MOV	R4,(R5)+	;FILL TABLE
7005	011126	020527	000614		CMP	R5,#MTRD+2	;DONE?
7006	011132	001372			BNE	2\$;IF NOT: BR
7007	011134	012704	025276		MOV	#MSG85,R4	
7008	011140	004737	020530		JSR	PC,TTOUT	;REQUEST VECTOR ADDR.
7009	011144	013703	000614		MOV	VECT,R3	
7010	011150	004737	020716		JSR	PC,OTCP	;PRINT CURRENT VECTOR
7011	011154	012705	000614		MOV	#VECT,R5	;SET SAVE LOCATION
7012	011160	012701	000003		MOV	#3,R1	;SET SIZE OF RESPONSE
7013	011164	012702	000476		MOV	#476,R2	;SET UPPER LIMIT
7014	011170	012703	000060		MOV	#60,R3	;SET LOWER LIMIT
7015	011174	004737	020272		JSR	PC,TTR	;GO GET RESPONSE
7016	011200	013700	000614		MOV	VECT,R0	;GET VECTOR ADDRESS
7017	011204	012720	017466		MOV	#MTINT,(R0)+	;LOAD VECTOR WITH HANDLER ADDR.
7018	011210	012710	000340		MOV	#340,(R0)	;LOAD PRIORITY LEVEL
7019	011214	005737	021624		TST	ASEQF	;SEE IF AUTO SEQ
7020	011220	001403			BEQ	TINPOD	;IF NOT: BR
7021	011222	005726			TST	(SP)+	;RESET STACK
7022	011224	000137	021252		JMP	ASEQ	;GO TO AUTO SEQ
7023	011230	012704	023314		TINPOD: MOV	#MSG32,R4	
7024	011234	004737	020530		JSR	PC,TTOUT	;PRINT UNIT NUMBER REQUEST
7025	011240	005037	000710		CLR	TEMP2	;CLEAR BUFFER
7026	011244	012705	000710		MOV	#TEMP2,R5	;SET UNIT DESCRIPTION BUFFER ADDRESS
7027	011250	012701	000001		MOV	#1,R1	;SET NUMBER OF CHARACTERS TO INPUT
7028	011254	012702	000007		MOV	#7,R2	;SET MAXIMUM LIMIT
7029	011260	012703	000000		MOV	#0,R3	;SET MINIMUM LIMIT
7030	011264	004737	020272		JSR	PC,TTR	;GO GET UNIT NUMBER
7031	011270	005737	000706		TST	TEMP1	;SEE IF HAVE NEW PARAMETER
7032	011274	001014			BNE	TINPOB	;IF SO: BR
7033	011276	005737	000734		TST	UNP	;SEE IF FIRST ENTRY
7034	011302	001002			BNE	TINPOA	;IF NOT: BR
7035	011304	000137	011230		JMP	TINPOD	;ELSE RETRY

7036	011310	013700	000734		TINPOA:	MOV	UNP,RO	
7037	011314	012760	177777	001012		MOV	#-1,UN1(RO)	;SET END UNIT TABLE
7038	011322	000137	011634			JMP	TINP2B	;GO GET RECORD COUNT
7039	011326	013700	000734		TINPOB:	MOV	UNP,RO	
7040	011332	042760	003400	001012		BIC	#3400,UN1(RO)	;CLEAR UNIT NUMBER
7041	011340	012703	000010			MOV	#10,R3	;SET ROTATION FACTOR
7042	011344	004737	012342			JSR	PC,TPOS	;GO LOAD UNIT NUMBER TO PROPER POSITION
7043	011350	016037	001012	000616		MOV	UN1(RO),UDES	;SELECT UNIT
7044	011356	013777	000616	167216		MOV	UDES,UMTC	;LOAD UNIT NUMBER
7045	011364	032777	000100	167206	TINPOC:	BIT	#100,UMTS	;SEE IF UNIT AVAILABLE
7046	011372	001011				BNE	TINPOD	;IF RO: BR
7047	011374	005337	000730			DEC	STAL	
7048	011400	001371				BNE	TINPOC	;DELAY
7049	011402	012704	023701			MOV	#MSG49,R4	
7050	011406	004737	020530			JSR	PC,TTOUT	;PRINT UNIT NOT AVAILABLE
7051	011412	000137	011230			JMP	TINPOD	;REDO
7052	011416	032777	000020	167154	TINPOD:	BIT	#20,UMTS	;SEE IF 7 CHANNEL
7053	011424	001404				BEQ	TINPOE	;IF NOT: BR
7054	011426	012704	023715			MOV	#MSG50,R4	;7 CHANNEL MSG
7055	011432	000137	011442			JMP	TINPOF	
7056	011436	012704	023724		TINPOE:	MOV	#MSG51,R4	;9 CHANNEL MSG
7057	011442	004737	020530		TINPOF:	JSR	PC,TTOUT	;GO PRINT 7 OR 9 CHANNEL
7058	011446	012704	023334		TINP1:	MOV	#MSG33,R4	
7059	011452	004737	020530			JSR	PC,TTOUT	;PRINT DENSITY REQUEST
7060	011456	005037	000710			CLR	TEMP2	;CLEAR BUFFER
7061	011462	012701	000001			MOV	#1,R1	;SET NUMBER OF CHARACTERS TO INPUT
7062	011466	012702	000003			MOV	#3,R2	;SET MAXIMUM LIMIT
7063	011472	012703	000000			MOV	#0,R3	;SET MINIMUM LIMIT
7064	011476	004737	020272			JSR	PC,TTR	;GO GET DENSITY
7065	011502	005737	000706			TST	TEMP1	;SEE IF HAVE NEW PARAMETER
7066	011506	001407				BEQ	TINP2	;IF NOT: BR
7067	011510	042737	060000	000616		BIC	#60000,UDES	;ELSE CLEAR OLD PARAMETER
7068	011516	012703	000015			MOV	#15,R3	;SET POSITION FACTOR
7069	011522	004737	012342			JSR	PC,TPOS	;GO LOAD DENSITY INTO PROPER POSITION
7070	011526	012704	023350		TINP2:	MOV	#MSG34,R4	
7071	011532	004737	020530			JSR	PC,TTOUT	;PRINT PARITY REQUEST
7072	011536	005037	000710			CLR	TEMP2	;CLR BUFFER
7073	011542	012701	000001			MOV	#1,R1	;SET NUMBER OF CHARACTERS TO INPUT
7074	011546	012702	000001			MOV	#1,R2	;SET MAXIMUM LIMIT
7075	011552	012703	000000			MOV	#0,R3	;SET MINIMUM LIMIT
7076	011556	004737	020272			JSR	PC,TTR	;GO INPUT PARITY
7077	011562	005737	000706			TST	TEMP1	;SEE IF HAVE NEW PARAMETER
7078	011566	001407				BEQ	TINP2A	;IF NOT: BR
7079	011570	042737	004000	000616		BIC	#4000,UDES	;ELSE CLEAR OLD PARAMETER
7080	011576	012703	000013			MOV	#13,R3	;SET POSITION FACTOR
7081	011602	004737	012342			JSR	PC,TPOS	;GO LOAD PARITY TO PROPER POSITION
7082	011606	005237	004716		TINP2A:	INC	REOTC	;BUMP EOT UNIT COUNTER
7083	011612	022737	000016	000734		CMP	#16,UNP	;SEE IF DONE UNITS
7084	011620	001405				BEQ	TINP2B	;IF SO: BR
7085	011622	062737	000002	000734		ADD	#2,UNP	;POINT TO NEXT UNIT
7086	011630	000137	011230			JMP	TINPOD	;ELSE LOOK FOR NEXT UNIT
7087	011634	005037	000734		TINP2B:	CLR	UNP	;CLEAR UNIT POINTER
7088	011640	013700	004716			MOV	REOTC,RO	
7089	011644	000337	004716			SWAB	REOTC	
7090	011650	110037	004716			MOVB	RO,REOTC	;SET UNIT EOT COUNTER
7091	011654	012704	023363		TINP3:	MOV	#MSG35,R4	

K05

7092	011660	004737	020530		JSR	PC, TTOUT	;PRINT RECORD COUNT REQUEST
7093	011664	013703	000620		MOV	RCNT, R3	
7094	011670	004737	020716		JSR	PC, OCTP	;PRINT RECORD COUNT
7095	011674	012705	000620		MOV	#RCNT, R5	;SET RECORD COUNT ADDRESS
7096	011700	012701	000006		MOV	#6, R1	;SET NUMBER OF CHARACTERS TO INPUT
7097	011704	012702	177777		MOV	#-1, R2	;SET MAXIMUM LIMIT
7098	011710	012703	000001		MOV	#1, R3	;SET MINIMUM LIMIT
7099	011714	004737	020272		JSR	PC, TTR	;GO GET RECORD COUNT
7100	011720	013737	000620	000642	MOV	RCNT, RCSAV	;SAVE RECORD COUNT
7101	011726	012704	023404		MOV	#MSG36, R4	
7102	011732	004737	020530		JSR	PC, TTOUT	;PRINT CHARACTER COUNT REQUEST
7103	011736	005437	000622		NEG	CARCNT	
7104	011742	013703	000622		MOV	CARCNT, R3	
7105	011746	004737	020716		JSR	PC, OCTP	;PRINT CHAR COUNT
7106	011752	012705	000622		MOV	#CARCNT, R5	;SET CHARACTER COUNT ADDRESS
7107	011756	012701	000006		MOV	#6, R1	;SET NUMBER OF CHARACTERS TO INPUT
7108	011762	012702	004000		MOV	#4000, R2	;SET MAXIMUM LIMIT
7109	011766	012703	000004		MOV	#4, R3	;SET MINIMUM LIMIT
7110	011772	004737	020272		JSR	PC, TTR	;GO GET CHARACTER COUNT
7111	011776	005437	000622		NEG	CARCNT	;SET TO TWO'S COMPLEMENT
7112	012002	013737	000622	000644	MOV	CARCNT, CCSAV	;SAVE CHAR COUNT
7113	012010	012704	023430		MOV	#MSG37, R4	;PRINT PATTERN NUMBER REQUEST
7114	012014	004737	020530		JSR	PC, TTOUT	
7115	012020	013703	000624		MOV	PATRN, R3	
7116	012024	004737	020716		JSR	PC, OCTP	;PRINT PATTERN
7117	012030	005037	012746		CLR	DOFL	;CLEAR EXTERNAL DATA FLAG
7118	012034	012705	000624		MOV	#PATRN, R5	;SET PATTERN NUMBER ADDRESS
7119	012040	012701	000002		MOV	#2, R1	;SET NUMBER OF CHARACTERS TO INPUT
7120	012044	012702	000015		MOV	#15, R2	;SET MAXIMUM LIMIT
7121	012050	012703	000000		MOV	#0, R3	;SET MINIMUM LIMIT
7122	012054	004737	020272		JSR	PC, TTR	;GO GET PATTERN NUMBER
7123	012060	012704	024266		MOV	#MSG60, R4	;PRINT TM REQUEST
7124	012064	004737	020530		JSR	PC, TTOUT	
7125	012070	013703	000646		MOV	TMEX, R3	
7126	012074	004737	020716		JSR	PC, OCTP	;PRINT TMEX VALUE
7127	012100	012705	000646		MOV	#TMEX, R5	;SE TMEX ADDRESS
7128	012104	012701	000001		MOV	#1, R1	;SET NUMBER OF CHARACTERS TO INPUT
7129	012110	010102			MOV	R1, R2	;SET MAXIMUM LIMIT
7130	012112	005003			CLR	R3	;SET MINIMUM LIMIT
7131	012114	004737	020272		JSR	PC, TTR	;GO GET RESPONSE
7132	012120	012704	023453		MOV	#MSG38, R4	
7133	012124	004737	020530		JSR	PC, TTOUT	;PRINT SINGLE PASS REQUEST
7134	012130	013703	000630		MOV	SPFLG, R3	
7135	012134	004737	020716		JSR	PC, OCTP	;PRINT CURRENT FLAG SETTING
7136	012140	012705	000630		MOV	#SPFLG, R5	;GET ADDRESS OF FLAG
7137	012144	012701	000001		MOV	#1, R1	;SET SIZE OF RESPONSE
7138	012150	012702	000001		MOV	#1, R2	;SET UPPER LIMIT
7139	012154	012703	000000		MOV	#0, R3	;SET LOWER LIMIT
7140	012160	004737	020272		JSR	PC, TTR	;GO GET RESPONSE
7141	012164	012704	023513	TINP4:	MOV	#MSG40, R4	
7142	012170	004737	020530		JSR	PC, TTOUT	;PRINT READ STALL REQUEST
7143	012174	013703	000632		MOV	RSTAL, R3	
7144	012200	004737	020716		JSR	PC, OCTP	;PRINT READ STALL
7145	012204	012705	000632		MOV	#RSTAL, R5	;SET READ STALL ADDRESS
7146	012210	012701	000006		MOV	#6, R1	;SET NUMBER OF CHARACTERS TO INPUT
7147	012214	012702	177777		MOV	#-1, R2	;SET MAXIMUM LIMIT

```

7148 012220 012703 000001      MOV      #1,R3      ;SET MINIMUM LIMIT
7149 012224 004737 020272      JSR      PC,TTR     ;GO GET READ STALL
7150 012230 012704 023542      MOV      #MSG41,R4
7151 012234 004737 020530      JSR      PC,TTOUT   ;PRINT WRITE STALL REQUEST
7152 012240 013703 000634      MOV      #WSTAL,R3
7153 012244 004737 020716      JSR      PC,OCIP    ;PRINT READ STALL
7154 012250 012705 000634      MOV      #WSTAL,R5 ;SET WRITE STALL ADDRESS
7155 012254 012701 000006      MOV      #6,R1     ;SET NUMBER OF CHARACTERS TO INPUT
7156 012260 012702 177777      MOV      #-1,R2    ;SET MAXIMUM LIMIT
7157 012264 012703 000001      MOV      #1,R3     ;SET MINIMUM LIMIT
7158 012270 004737 020272      JSR      PC,TTR     ;GO GET WRITE STALL
7159 012274 012704 023554      MOV      #MSG42,R4
7160 012300 004737 020530      JSR      PC,TTOUT   ;PRINT TURN AROUND STALL REQUEST
7161 012304 013703 000636      MOV      #TSTAL,R3
7162 012310 004737 020716      JSR      PC,OCIP    ;PRINT TA STALL
7163 012314 012705 000636      MOV      #TSTAL,R5 ;SET TURN AROUND STALL ADDRESS
7164 012320 012701 000006      MOV      #6,R1     ;SET NUMBER OF CHARACTERS TO INPUT
7165 012324 012702 177777      MOV      #-1,R2    ;SET MAXIMUM LIMIT
7166 012330 012703 000001      MOV      #1,R3     ;SET MINIMUM LIMIT
7167 012334 004737 020272      JSR      PC,TTR     ;GO GET TURN AROUND STALL
7168 012340 000207      RTS      PC         ;EXIT
  
```

:UNIT DESCRIPTION POSITIONING SUBROUTINE*****

```

7171
7172 012342 000241      TPOS:  CLC
7173 012344 006137 000710      ROL      TEMP2     ;POSITION CHARACTER
7174 012350 005303      DEC      R3        ;SEE IF DONE
7175 012352 001373      BNE      TPOS      ;IF NOT: BR
7176 012354 013700 000734      MOV      UNP,RO    ;LOAD UNIT POINTER
7177 012360 053760 000710 001012      BIS      TEMP2,UN1(RO) ;LOAD CHARACTER INTO UN1(RO)
7178 012366 000207      RTS      PC         ;EXIT
  
```

M05

7181
7182
7183
7184
7185
7186
7187
7188
7189
7190
7191
7192
7193
7194
7195
7196
7197
7198
7199
7200
7201
7202
7203
7204
7205
7206
7207
7208
7209
7210
7211
7212
7213
7214
7215
7216
7217
7218
7219
7220
7221
7222
7223
7224
7225
7226
7227
7228
7229
7230
7231
7232
7233
7234
7235

012370 005737 013304
 012374 001045
 012376 005737 021624
 012402 001406
 012404 005737 000624
 012410 100003
 012412 004737 013236
 012416 000207
 012420 023737 000624 012750
 012426 001014
 012430 013703 000616
 012434 042703 173777
 012440 023703 012752
 012444 001404
 012446 010337 012752
 012452 004737 013306
 012456 000207
 012460
 012460 012703 026176
 012464 013701 000624
 012470 010137 012750
 012474 062701 000001
 012500 000241
 012502 006101
 012504 000171 002734
 012510 000240
 012512 004737 013306
 012516 012702 002002
 012522 012701 032210
 012526 005021
 012530 005302
 012532 001375
 012534 013737 000616 012752
 012542 042737 173777 012752
 012550 000207

```

*****
;DATA SETUP ROUTINE:
;THIS ROUTINE IS USED TO GENERATE INTO THE ENTIRE
;WRITE BUFFER (4000 OCTAL CHARACTERS) THE DATA PATTERN
;SELECTED BY THE OPERATOR. THESE ARE 20 (8) FIXED
;DATA PATTERNS AVAILABLE AND ONE SELECTION (DATA PATTERN 0)
;WHICH WILL READ ANY PATTERN PRESENTED AT THE
;HIGH SPEED PAPER TAPE READER. THIS TAPE MUST BE PREPARED
;BY USING THE PROGRAM CALLED DTC.
;RANDOM DATA MAY ALSO BE USED VIA CONSOLE
;SWITCH EIGHT (8).
;THIS ROUTINE IS ALSO USED TO CLEAR OUT THE
;READ BUFFER (4000 OCTAL CHARACTERS) BEFORE EACH
;RECORD IS READ.
*****
DSUP: TST R0FL ;SEE IF DID RANDOM DATA
      BNE DS1 ;IF SO: BR
DSO: TST ASEQF ;SEE IF AUTO SEQ
      BEQ DSOA ;IF NOT: BR
      TST PATRN ;SEE IF AUTO RANDOM
      BPL DSOA ;IF NOT: BR
      JSR PC,DATA ;ELSE GO GENERATE RANDOM DATA
      RTS PC ;RETURN
DSOA: CMP PATRN,PATS ;NEW PATTERN?
      BNE DSOC ;IF SO: BR
      MOV UDES,R3 ;GET UNIT DESCRIPTION
      BIC #173777,R3 ;MASK PARITY
      CMP PARS,R3 ;SEE IF SAME AS LAST TIME
      BEQ DSOB ;IF SO: BR
      MOV R3,PARS ;SAVE PARITY
      JSR PC,CRCLRC ;GO GENERATE EXPECTED CRC/LRC
      RTS PC
DSOB:
DSOC: MOV #WDATA,R3 ;R3 = ADDRS OF WRITE BUFFER
      MOV PATRN,R1 ;R1 = PATTERN SELECTOR
      MOV R1,PATS
      ADD #1,R1 ;BUMP POINTER
      CLC
      ROL R1 ;MAKE PATTERN SELECTOR EVEN
      JMP @DATBL(R1) ;GO GENERATE PATTERN
DS1: NOP
      JSR PC,CRCLRC
DS3: MOV #2002,R2 ;R2=BUFFER SIZE +2
      MOV #RDATA,R1 ;R1=READ DATA START
      CLR (R1)+ ;CLEAR BUFFER
      DEC R2 ;SEE IF DONE ALL
      BNE DS4 ;IF NOT: BR
      MOV UDES,PARS ;SET PARITY
      BIC #173777,PARS
      RTS PC ;EXIT
  
```

```

7237
7238 ;EXTERNAL DATA INPUT FROM H/S READER (256 CHARACTER MAXIMUM)
7239
7240 012552 005737 012746 DATO: TST DOFL ;SEE IF SHOULD DO EXTERNAL INPUT
7241 012556 001354 BNE DS1 ;IF NOT: BR
7242 012560 012737 000001 012746 MOV #1,DOFL ;SET EXTERNAL FLAG
7243 012566 005077 166076 CLR @PRB ;CLEAR READER BUFFER
7244 012572 005077 166070 CLR @PRS ;CLEAR READER STATUS
7245 012576 005037 000706 CLR TEMP1 ;CLEAR FOR USE AS CHARACTER FLAG
7246 012602 052777 000001 166056 DATOA: BIS #1,@PRS ;START READER
7247 012610 005037 000714 CLR TEMP4
7248 012614 012704 000004 MOV #4,R4 ;SET UP READER DONE DELAY
7249 012620 032777 000200 166040 DATOB: BIT #200,@PRS ;SEE IF DONE
7250 012626 001006 BNE IS ;IF SO :BR
7251 012630 005337 000714 DEC TEMP4
7252 012634 001371 BNE DATOB ;DELAY FOR READER DONE
7253 012636 005304 DEC R4
7254 012640 001367 BNE DATOB ;CONTINUE DELAY
7255 012642 000722 BR DS1 ;IF READER NEVER DONE: BR
7256 012644 005001 IS: CLR R1 ;CLEAR SAVE LOCATION
7257 012646 117701 166016 MOVB @PRB,R1 ;SAVE CHARACTER
7258 012652 005737 000706 TST TEMP1 ;SEE IF HAVE FOUND START CHARACTER
7259 012656 001012 BNE DATOC ;IF SO : BR
7260 012660 105701 TSTB R1 ;SEE IF CHARACTER IS 0
7261 012662 001747 BEQ DATOA ;IF SO : BR
7262 012664 012737 000001 000706 MOV #1,TEMP1 ;ELSE SET CHARACTER FOUND FLAG
7263 012672 010137 000710 MOV R1,TEMP2 ;SAVE DATA SIZE
7264 012676 010102 MOV R1,R2 ;SAVE DATA SIZE
7265 012700 000137 012602 JMP DATOA ;GO GET FIRST DATA CHAR
7266 012704 110123 DATOC: MOVB R1,(R3)+ ;LOAD BUFFER
7267 012706 005302 DEC R2 ;SEE IF READ ALL
7268 012710 001334 BNE DATOA ;IF NOT : BR
7269 012712 012701 026176 DATOD: MOV #WDATA,R1 ;R1 = START OF WRITE BUFFER
7270 012716 013702 000710 MOV TEMP2,R2 ;R2 = SIZE OF DATA FIELD
7271 012722 112123 DATOE: MOVB (R1)+,(R3)+ ;REPEAT LOAD OF DATA FIELD
7272 012724 022703 032210 CMP #RDATA,R3 ;SEE IF DONE
7273 012730 003002 BGT DATOF ;IF NOT: BR
7274 012732 000137 012510 JMP DS1 ;EXIT
7275 012736 005302 DATOF: DEC R2 ;SEE IF AT END OF DATA FIELD
7276 012740 001370 BNE DATOE ;IF NOT : BR
7277 012742 000137 012712 JMP DATOD ;ELSE RESTART FILL
7278 012746 000000 DOFL: 0 ;EXTERNAL DATA FLAG=1 IF ALREADY DONE
7279 012750 177777 PATS: -1
7280 012752 177777 PARS: -1
7281
    
```

```

7283
7284
7285
7286 012754 012701 177777
7287 012760 012702 002002
7288 012764 010123
7289 012766 005302
7290 012770 001375
7291 012772 000137 012510
7292
7293
7294
7295 012776 005001
7296 013000 000137 012760
7297
7298
7299
7300 013004 012701 000001
7301 013010 000241
7302 013012 012702 004004
7303 013016 110123
7304 013020 106101
7305 013022 005302
7306 013024 001374
7307 013026 000137 012510
7308
7309
7310
7311 013032 012701 000376
7312 013036 000261
7313 013040 000137 013012
7314
7315
7316
7317
7318 013044 012701 052525
7319 013050 000137 012760
7320
7321
7322
7323 013054 012701 125252
7324 013060 000137 012760
7325
7326
7327
7328 013064 012701 125125
7329 013070 000137 012760
7330
7331
7332
7333 013074 012701 052652
7334 013100 000137 012760
7335
  
```

```

; ALL ONES*****
DAT1:  MOV    #-1, R1      ; R1=DATA
DAT1A: MOV    #2002, R2   ; R2=WORD COUNT +2
DAT1B: MOV    R1, (R3)+   ; LOAD BUFFER
        DEC    R2         ; SEE IF DONE
        BNE   DAT1B      ; IF NOT: BR
        JMP   DS1        ; RETURN

; ALL ZEROS*****
DAT2:  CLR    R1          ; R1=DATA
        JMP   DAT1A      ; LOAD BUFFER

; WALKING ONE*****
DAT3:  MOV    #1, R1     ; R1=DATA
        CLC
DAT3A: MOV    #4004, R2   ; R2=CHARACTER COUNT+4
DAT3B: MOVB   R1, (R3)+  ; LOAD BUFFER
        ROLB  R1         ; SET NEXT CHARACTER
        DEC   R2         ; SEE IF DONE
        BNE   DAT3B      ; IF NOT: BR
        JMP   DS1        ; RETURN

; WALKING ZERO*****
DAT4:  MOV    #376, R1   ; R1=START OF DATA
        SEC
        JMP   DAT3A      ; LOAD BUFFER

; ALTERNATING ONE/ZERO*****
DAT5:  MOV    #52525, R1 ; R1=DATA
        JMP   DAT1A      ; LOAD BUFFER

; ALTERNATING ZERO/ONE*****
DAT6:  MOV    #125252, R1 ; R1=DATA
        JMP   DAT1A      ; LOAD BUFFER

; ONE/ZERO IN ALTERNATING CHARACTERS*****
DAT7:  MOV    #125125, R1 ; R1=DATA
        JMP   DAT1A      ; LOAD BUFFER

; ZERO/ONE IN ALTERNATING CHARACTERS*****
DAT10: MOV    #52652, R1  ; R1=DATA
        JMP   DAT1A      ; LOAD BUFFER
  
```

```

7337
7338
7339
7340 013104 005001
7341 013106 012702 004004
7342 013112 110123
7343 013114 105201
7344 013116 005302
7345 013120 001374
7346 013122 000137 012510
7347
7348
7349
7350 013126 012701 000377
7351 013132 012702 004004
7352 013136 110123
7353 013140 105301
7354 013142 005302
7355 013144 001374
7356 013146 000137 012510
7357
7358
7359
7360 013152 012701 000377
7361 013156 000137 012760
7362
7363
7364
7365 013162 012701 177400
7366 013166 000137 012760
7367
7368
7369
7370 013172 012702 002002
7371 013176 012701 177376
7372 013202 012704 000002
7373 013206 010123
7374 013210 005302
7375 013212 001002
7376 013214 000137 012510
7377 013220 005304
7378 013222 001371
7379 013224 000261
7380 013226 006101
7381 013230 103764
7382 013232 000137 013176
7383
    ;ALL BITS 0-377*****
DAT11: CLR R1 ;R1=STARTING DATA
        MOV #4004,R2 ;R2=CHARACTER COUNT+4
DAT11A: MCVB R1,(R3)+ ;LOAD BUFFER
        INCB R1 ;BUMP DATA
        DEC R2 ;SEE IF DONE
        BNE DAT11A ;IF NOT: BR
        JMP DS1 ;RETURN

    ;ALL BITS 377-0*****
DAT12: MOV #377,R1 ;R1=STARTING DATA
        MOV #4004,R2 ;R2=CHARACTER COUNT+4
DAT12A: MOVB R1,(R3)+ ;LOAD BUFFER
        DECB R1 ;BUMP DATA
        DEC R2 ;SEE IF DONE
        BNE DAT12A ;IF NOT: BR
        JMP DS1 ;RETURN

    ;ALTERNATING CHARACTERS 0 AND 377*****
DAT13: MOV #377,R1 ;R1 = DATA
        JMP DAT1A ;LOAD BUFFER

    ;ALTERNATING CHARACTERS 377 AND 0*****
DAT14: MOV #177400,R1 ;R1 = DATA
        JMP DAT1A ;LOAD BUFFER

    ;WALKING ZERO REPEATED FOUR TIMES*****
DAT15: MOV #2002,R2 ;SET NUMBER OF WORDS
DAT15R: MOV #177376,R1 ;SET START OF DATA
DAT15A: MOV #2,R4 ;SET NUMBER OF REPEATS
DAT15B: MOV R1,(R3)+ ;LOAD DATA
        DEC R2 ;SEE IF DONE
        BNE DAT15C ;IF NOT: BR
        JMP DS1 ;RETURN
DAT15C: DEC R4 ;SEE IF DONE REPEATS
        BNE DAT15B ;IF NOT: BR
        SEC
        ROL R1 ;SET NEXT PATTERN
        BCS DAT15A ;SEE IF SHOULD RESTART
        JMP DAT15R ;IF SO: BR
    
```



```

7385
7386           ;RANDOM DATA GENERATOR SUBROUTINE*****
7387
7388 013236 013704 000622   DATR:  MOV   CARCNT, R4       ;SET SIZE OF RECORD
7389 013242 012703 026176   MOV   #WDATA, R3     ;SET ADDRESS OF START OF BUFFER
7390 013246 012701 177777   MOV   #-1, R1        ;SET HIGH LIMIT
7391 013252 005002          CLR   R2             ;SET LOW LIMIT
7392 013254 004737 020240   DATRO: JSR  PC, RANG  ;GO GENERATE NUMBER
7393 013260 013723 000676   MOV   RANSV, (R3)+  ;LOAD BUFFER
7394 013264 005204          INC   R4             ;SEE IF DONE ALL
7395 013266 001372          BNE   DATRO         ;IF NOT: BR
7396 013270 004737 012510   JSR  PC, DS1        ;GO CHECK FOR 7 CH
7397 013274 012737 000001 013304  MOV   #1, RDFL      ;SET RANDOM DATA FLAG
7398 013302 000207          RTS   PC           ;EXIT
7399 013304 000000          RDFL: 0           ;RANDOM DATA SELECT FLAG

```

```

7401
7402
7403
7404
7405
7406
7407
7408
7409
7410 013306 000240
7411 013310 013700 000622
7412 013314 005400
7413 013316 012701 026176
7414 013322 005037 013674
7415 013326 111104
7416 013330 004737 013522
7417 013334 004737 013650
7418 013340 000241
7419 013342 006004
7420 013344 103014
7421 013346 052704 000400
7422 013352 000241
7423 013354 010405
7424 013356 042705 177703
7425 013362 005105
7426 013364 042705 177703
7427 013370 042704 000074
7428 013374 050504
7429 013376 010437 013674
7430 013402 005300
7431 013404 001402
7432 013406 000137 013326
7433 013412 013704 013674
7434 013416 005137 013674
7435 013422 042737 177050 013674
7436 013430 042704 177727
7437 013434 050437 013674
7438 013440 013737 013674 013676
7439 013446 013700 000622
7440 013452 005400
7441 013454 012701 026176
7442 013460 005037 013674
7443 013464 111104
7444 013466 004737 013522
7445 013472 004737 013650
7446 013476 005300
7447 013500 001371
7448 013502 013704 013676
7449 013506 004737 013650
7450 013512 013737 013674 013700
7451 013520 000207
7452 013522 005704
7453 013524 001010
7454 013526 032737 004000 000616
7455 013534 001404
7456 013536 012704 000420

```

```

;*****
;CRC/LRC CHARACTER BUILD;
;THIS ROUTINE WILL CONSTRUCT AND SAVE THE EXPECTED
;CRC AND LRC CHARACTERS ACCORDING TO DATA AND
;RECORD SIZE IF OPERATING IN NRZ MODE
;*****
CRCLRC: NOP
CRLR:  MOV  CARCNT,RO      ;SET RECORD SIZE
      NEG  RO
      MOV  #WDATA,R1     ;SET START OF BUFFER
      CLR  XORS
      CLO: MOV  (R1),R4   ;GET CHARACTER
      JSR  PC,CLP        ;GO GET PARITY OF CHARACTER
      JSR  PC,XOR        ;XOR CHARACTER
      CLC
      ROR  R4             ;ROTATE 1 RIGHT
      BCC  CL2           ;IF NO CARRY: BR
      BIS  #400,R4       ;SET BIT NINE
      CLC
      CL1: MOV  R4,R5     ;SAVE CHARACTER
      BIC  #177703,R5
      COM  R5
      BIC  #177703,R5
      BIS  #74,R4
      CL2: MOV  R5,R4     ;COMPLEMENT BITS 2,3,4,5
      XORS R4,XORS
      DEC  RO
      BEQ  CLLAST       ;IF LAST CHARACTER: BR
      JMP  CLO          ;GET NEXT
      CLLAST: MOV  XORS,R4
      COM  XORS
      BIC  #177050,XORS  ;COMPLEMENT ALL BUT BITS 3&5
      BIC  #177727,R4
      BIS  R4,XORS
      MOV  XORS,EXCRC   ;SAVE EXPECTED CRC
      MOV  CARCNT,RO
      NEG  RO
      MOV  #WDATA,R1    ;DO EXPT LRC
      CLR  XORS
      CL3: MOV  (R1),R4  ;GET PARITY
      JSR  PC,CLP        ;XOR CHARACTER
      JSR  PC,XOR
      DEC  RO
      BNE  CL3          ;DO ALL FOR LRC
      MOV  EXCRC,R4
      JSR  PC,XOR       ;XOR CRC TO DATA
      MOV  XORS,EXLRC   ;SAVE EXPT LRC
      RTS  PC           ;RETURN
      CLP: TST  R4       ;SEE IF 0 CHAR
      BNE  CLPE         ;IF NOT: BR
      BIT  #4000,UDES   ;SEE IF EVEN PARITY
      BEQ  CLPE         ;IF NOT: BR
      MOV  #420,R4     ;SET 0 CHAR EVEN PARITY

```


H06

7552	014126	117737	000106	014236		MOV8	STAS,STCS	;SAVE CHARACTER
7553	014134	142711	000300			BICB	#300,(R1)	;MASK FOR 7 TRACK DRIVE
7554	014140	122122				CMPB	(R1)+,(R2)+	;SEE IF DATA IS GOOD
7555	014142	001003				BNE	DF71	;IF NOT: BR
7556	014144	105037	000722			CLRB	BBC	;CLEAR BAD RECORD COUNTER
7557	014150	000407				BR	DF72	
7558	014152	004737	014656		DF71:	JSR	PC,DRPKF	;GO DO DROPS AND PICKS
7559	014156	004737	014242			JSR	PC,DERR	;GO PRINT ERROR
7560	014162	012737	000001	000744		MOV	#1,DERFL	;SET DATA ERROR FLAG
7561	014170	000240			DF72:	NOP		
7562	014172	153777	014236	000040		BISB	STCS,STAS	;RESET DATA
7563	014200	005205				INC	RS	;SEE IF DONE ALL
7564	014202	001346				BNE	DF7	;IF NOT: DO ALL
7565	014204	005737	000744		DF3:	TST	DERFL	;SEE IF HAD DATA ERROR
7566	014210	001411				BEQ	DFX	;IF NOT: BR
7567	014212	005737	000742			TST	SERFL	
7568	014216	001006				BNE	DFX	;IF NOT DATA ERROR ONLY: BR
7569	014220	013704	000734			MOV	UNP,R4	
7570	014224	005264	001134			INC	DATER1(R4)	;BUMP DATA ERROR COUNTER
7571	014230	004737	022120			JSR	PC,CKSWR	;CHECK FOR FG
7572	014234	000207			DFX:	RTS	PC	;EXIT
7573	014236	000000			STCS:	0		;7 TRACK DATA SAVE
7574	014240	000000			STAS:	0		;7 TRACK ADDRESS SAVE

7576
7577
7578
7579
7580
7581
7582
7583
7584
7585
7586
7587
7588
7589
7590
7591
7592
7593
7594
7595
7596
7597
7598
7599
7600
7601
7602
7603
7604
7605
7606
7607
7608
7609
7610
7611
7612
7613
7614
7615
7616
7617
7618
7619
7620
7621
7622
7623
7624
7625
7626
7627
7628
7629
7630
7631

014242 032777 002000 164402
014250 001402
014252 000137 014400
014256 005237 000732
014262 005737 000726
014266 001013
014270 005737 000742
014274 001010
014276 004737 017506
014302 012704 022426
014306 004737 020530
014312 004737 016742
014316 012704 022445
014322 004737 020530
014326 010203
014330 162703 032210
014334 005303
014336 004737 020716
014342 012704 022433
014346 004737 020530
014352 114103
014354 004737 021144
014360 012704 022440
014364 004737 020530
014370 114203
014372 004737 021144
014376 122122
014400 105237 000722

DERR: BIT #2000,JSWR
BEQ DERRO
JMP DERR4
DERR0: INC PFLG
TST HDRFL
BNE DERR0A
TST SERFL
BNE DERR0A
JSR PC,PAPRT
MOV #MSG1,R4
JSR PC,TTOUT
JSR PC,FRPRT
DERR0A: MOV #MSG4,R4
JSR PC,TTOUT
MOV R2,R3
SUB #RDATA,R3
DEC R3
DERR0B: JSR PC,OCTP
MOV #MSG2,R4
JSR PC,TTOUT
MOVB -(R1),R3
JSR PC,DOUT
MOV #MSG3,R4
JSR PC,TTOUT
DERR1: MOVB -(R2),R3
DERR2: JSR PC,DOUT
DERR3: CMPB (R1)+,(R2)+
DERR4: INCB BBC

;SEE IF SHOULD PRINT ERRORS
;IF SO: BR
;ELSE SKIP PRINT
;SET PRINT FLAG
;SEE IF HAVE PRINTED HEADER
;IF SO: BR
;ALREADY PRINTED HEADER?
;IF SO: BR
;PRINT CYCLE NUMBER
;LOAD ERROR MSG ADDR
;PRINT ERROR
;PRINT F OR R
;PRINT CHAR NO. HEADER
;POINT TO CHAR
;PRINT CHAR NUMBER
;PRINT EXPECTED DATA
;LOAD EXPECTED DATA
;GO PRINT CHAR
;PRINT RECEIVED DATA
;PRINT BAD CHAR
;RESET POINTERS
;BUMP BAD RECORD CNTR

;DATA ERROR SUBROUTINE:
;THIS SUBROUTINE IS USED TO PRINT OUT ANY
;ERRORS FOUND DURING THE DATA CHECK.
;EACH CHARACTER FOUND BAD WILL BE PRINTED
;IN BIT FORMAT ALONG WITH ITS EXPECTED CHARACTER.
;AN ERROR HEADER CONSISTING OF THE UNIT NUMBER,
;BLOCK NUMBER, RECORD NUMBER, SIZE OF RECORD, AND
;ERROR TYPE (READ FORWARD, WRITE, ETC)
;IS PRINTED ONLY ONCE FOR EACH RECORD FOUND BAD.
;A COUNT IS MADE OF THE NUMBER OF SUCCESSIVE BAD
;CHARACTERS, AND IF TEN (10) SUCCESSIVE BAD CHARACTERS
;ARE FOUND IN A SINGLE RECORD, A MESSAGE INDICATING
;A BAD RECORD CONDITION IS PRINTED AND THE NEXT
;TWENTY (20) CHARACTERS ARE SKIPPED BEFORE CHECKING
;IS RESUMED. IF THE BAD RECORD CONDITION IS FOUND
;THREE TIMES IN A RECORD, ALL REMAINING DATA IS
;SKIPPED EXCEPT THE FINAL TEN (10) CHARACTERS.
;THIS SKIPPING IS OF COURSE ONLY POSSIBLE IN
;RECORDS WHICH CONTAIN A SUFFICIENT NUMBER OF CHARACTERS.
;PRINTING OF ERRORS MAY BE DISALLOWED AT ANY TIME
;BY SETTING CONSOLE SWITCH TEN (10) TO A ONE.
;THE OPERATOR MAY CAUSE THE PROGRAM TO HALT ON ANY ERROR
;BY SETTING CONSOLE SWITCH FIFTEEN (15) TO A ONE.

7632	014404	122737	000010	000722		CMPB	#10,BBC	;SEE IF BLD BTH
7633	014412	001076				BNE	DEREX	;IF NOT: BR
7634	014414	032777	002000	164230		BIT	#2000,ASWR	;SEE IF PRINT INHIBIT
7635	014422	001004				BNE	IS	;IF SO: BR
7636	014424	012704	022526			MOV	#MSG15,R4	
7637	014430	004737	020530			JSR	PC,TTOUT	;PRINT BLD BTH
7638	014434	105037	000722		1\$:	CLRB	BBC	;RESET BAD RECORD CNTR
7639	014440	000337	000722			SWAB	BBC	;POSITION BLD BTH AMOUNT
7640	014444	105237	000722			INCB	BBC	;BUMP AMOUNT
7641	014450	122737	000003	000722		CMPB	#3,BBC	;SEE IF HAD 3 BLD BTHS
7642	014456	101037				BHI	DERR4B	;IF NOT: BR
7643	014460	000337	000722			SWAB	BBC	;REPOSITION BBC
7644	014464	022705	177767			CMP	#177767,R5	;SEE IF ON LAST EIGHT CHARS
7645	014470	101445				BLOS	DERR6	;IF SO: BR
7646	014472	012705	177767			MOV	#177767,R5	;SET CHAR CNTR TO 8
7647	014476	013737	000622	000706		MOV	CARCNT,TEMP1	;LOAD CHAR COUNT
7648	014504	005137	000706			COM	TEMP1	
7649	014510	005237	000706			INC	TEMP1	
7650	014514	162737	000010	000706		SUB	#10,TEMP1	;POINT TO BUFFER -8
7651	014522	013701	000706			MOV	TEMP1,R1	;POINT TO NEXT CHAR
7652	014526	062701	026176			ADD	#WDATA,R1	;POINT TO NEXT WRITE CHAR
7653	014532	013702	000706			MOV	TEMP1,R2	;POINT TO END OF READ DATA -8 FORWARD
7654	014536	062702	032210			ADD	#RDATA,R2	;POINT TO NEXT CHAR
7655	014542	000422				BR	DEREX	;EXIT
7656	014544	012702	000010		DERR4A:	MOV	#10,R2	;POINT TO THE END OF READ DATA -8 REVERSE
7657	014550	062702	032210			ADD	#RDATA,R2	;POINT TO THE NEXT CHAR
7658	014554	000415				BR	DEREX	;EXIT
7659	014556	000337	000722		DERR4B:	SWAB	BBC	;REPOSITION BBC
7660	014562	000241				CLC		
7661	014564	062705	000024			ADD	#24,R5	;SKIP 20 CHARS
7662	014570	103405				BCS	DERR6	;IF EXCEED RECORD SIZE: BR
7663	014572	062701	000024			ADD	#24,R1	;SKIP 20 CHARS
7664	014576	062702	000024		DERR5:	ADD	#24,R2	;SKIP FORWARD 20 CHARS
7665	014602	000402				BR	DEREX	
7666	014604	012705	177777		DERR6:	MOV	#-1,R5	;SET TO EOR
7667	014610	032777	100000	164034	DEREX:	BIT	#100000,ASWR	;SEE IF SHOULD HALT ON ERROR
7668	014616	001412				BEQ	DEREX1	;IF NOT: BR
7669	014620	000000				HALT		
7670	014622	005737	000732			TST	PFLG	;SEE IF PRINTED
7671	014626	001006				BNE	DEREX1	;IF SO: BR
7672	014630	032777	002000	164014		BIT	#2000,ASWR	;SEE IF SHOULD PRINT
7673	014636	001002				BNE	DEREX1	;IF NOT: BR
7674	014640	000137	014256			JMP	DERRO	;ELSE PRINT
7675	014644	004737	022120		DEREX1:	JSR	PC,CKSWR	;TEST FOR IG
7676	014650	005037	000732			CLR	PFLG	;CLEAR FLAG
7677	014654	000207				RTS	PC	;RETURN
7678								

7737	015070	016403	001054		MOV	DRP1(R4),R3	;SET DROP POINTER
7738	015074	016404	001034		MOV	PIK1(R4),R4	;SET PICK POINTER
7739	015100	012737	000010	000736	MOV	#10,BCNT	;SET NUMBER OF BITS
7740	015106	005023			DPC2B: CLR	(R3)+	;CLEAR DROPS
7741	015110	005024			CLR	(R4)+	;CLEAR PICK
7742	015112	005337	000736		DEC	BCNT	;SEE IF DONE
7743	015116	001373			BNE	DPC2B	;IF NOT: BR
7744	015120	000207			RTS	PC	;EXIT
7745	015122	000241			DPC2: CLC		
7746	015124	106003			RORB	R3	;GET NEXT BIT
7747	015126	005337	000736		DEC	BCNT	;SEE IF DONE
7748	015132	001410			BEQ	DPC3	
7749	015134	062737	000002	000770	ADD	#2,BPKP	
7750	015142	062737	000002	000766	ADD	#2,BDPP	
7751	015150	000137	014762		JMP	DPC0	;CONTINUE
7752	015154	000207			RTS	PC	;RETURN
7753	015156	013704	000734		PICK: MOV	UNP,R4	;SET UNIT POINTER
7754	015162	016437	001034	000770	MOV	PIK1(R4),BPKP	;SET PICK POINTER
7755	015170	016437	001054	000766	MOV	DRP1(R4),BDPP	;SET DROP POINTER
7756	015176	113704	000706		MOV8	TEMP1,R4	;R4 = GOOD CHAR
7757	015202	113703	000710		MOV8	TEMP2,R3	;R3 = BAD CHAR
7758	015206	112737	000001	000712	MOV8	#1,TEMP3	;SET PICK FLAG
7759	015214	004737	014746		JSR	PC,DPC	;GO CHECK PICKS
7760	015220	000207			RTS	PC	;EXIT
7761	015222	012704	023113		DPPRT: MOV	#MSG26,R4	
7762	015226	004737	020530		JSR	PC,TTOUT	;PRINT DROP HEADER
7763	015232	013704	000734		MOV	UNP,R4	
7764	015236	016437	001054	000766	MOV	DRP1(R4),BDPP	;SET DROP POINTER
7765	015244	016437	001034	000770	MOV	PIK1(R4),BPKP	;SET PICK POINTER
7766	015252	062737	000016	000766	ADD	#16,BDPP	
7767	015260	062737	000016	000770	ADD	#16,BPKP	
7768	015266	012737	000010	000736	MOV	#10,BCNT	;SET NUMBER TO PRINT
7769	015274	017703	163466		DPPRTO: MOV	#BDPP,R3	
7770	015300	004737	020716		JSR	PC,OC1P	;PRINT DROPS
7771	015304	005337	000736		DEC	BCNT	;SEE IF DONE
7772	015310	001404			BEQ	DPPRT1	;IF NOT: BR
7773	015312	162737	000002	000766	SUB	#2,BDPP	;BUMP POINTER
7774	015320	000765			BR	DPPRTO	;CONTINUE FOR ALL 8 BITS
7775	015322	012737	000010	000736	DPPRT1: MOV	#10,BCNT	;SET NUMBER TO PRINT
7776	015330	012704	023124		MOV	#MSG27,R4	
7777	015334	004737	020530		JSR	PC,TTOUT	;PRINT PICK HEADER
7778	015340	017703	163424		DPPRT2: MOV	#BPKP,R3	
7779	015344	004737	020716		JSR	PC,OC1P	;PRINT PICKS
7780	015350	005337	000736		DEC	BCNT	;SEE IF DONE
7781	015354	001404			BEQ	DPPRTX	;IF SO: BR
7782	015356	162737	000002	000770	SUB	#2,BPKP	;BUMP POINTER
7783	015364	000765			BR	DPPRT2	;CONTINUE FOR ALL 8 BITS
7784	015366	000207			DPPRTX: RTS	PC	;RETURN

```

7787
7788
7789
7790
7791
7792
7793
7794
7795
7796
7797
7798
7799 015370 012700 000001
7800 015374 004737 017506
7801 015400 004737 015222
7802 015404 012704 024473
7803 015410 004737 020530
7804 015414 013704 000734
7805 015420 016403 001074
7806 015424 004737 020716
7807 015430 012704 024720
7808 015434 004737 020530
7809 015440 013704 000734
7810 015444 016403 001154
7811 015450 004737 020716
7812 015454 012704 024504
7813 015460 004737 020530
7814 015464 013704 000734
7815 015470 016403 001114
7816 015474 004737 020716
7817 015500 012704 024676
7818 015504 004737 020530
7819 015510 013704 000734
7820 015514 016403 001174
7821 015520 004737 020716
7822 015524 012704 024707
7823 015530 004737 020530
7824 015534 013704 000734
7825 015540 016403 001214
7826 015544 004737 020716
7827 015550 012704 024515
7828 015554 004737 020530
7829 015560 013704 000734
7830 015564 016403 001134
7831 015570 004737 020716
7832 015574 004737 015606
7833 015600 004737 022120
7834 015604 000207
7835

```

```

;*****
;STATISTICS PRINT
;
;THIS SUBROUTINE PRINTS THE ACCUMULATED
;ERROR STATISTICS FOR EACH DRIVE.
;THE ROUTINE CAN BE CALLED TO PRINT
;AT THE END OF EACH BLOCK BY SELECTING
;SW14=1. THE SUMMARY IS AUTOMATICALLY
;PRINTED FOR A DRIVE WHENEVER A TAPE
;IS REWOUND FROM EOT OR DROPPED.
;*****
PRSTAT: MOV #1,R0 ;SET RECORD COUNTER TO 1
JSR PC,PAPRT ;PRINT CYCLE NUMBER
PRSTA2: JSR PC,DPPRT ;PRINT DROPS AND PICKS
MOV #MSG64,R4
JSR PC,TTOUT ;PRINT WRITE ERROR TAG
MOV UNP,R4
MOV WTER1(R4),R3
JSR PC,OCTP ;PRINT WRITE ERRORS
MOV #MSG76,R4
JSR PC,TTOUT ;PRINT RETRY TOTAL
MOV UNP,R4
MOV RTY1(R4),R3
JSR PC,OCTP ;PRINT RETRIES
MOV #MSG65,R4
JSR PC,TTOUT ;PRINT READ ERROR TAG
MOV UNP,R4
MOV RDER1(R4),R3
JSR PC,OCTP ;PRINT READ ERRORS
MOV #MSG74,R4
JSR PC,TTOUT ;PRINT SOFT ERROR MESSAGE
MOV UNP,R4
MOV GDRTY1(R4),R3
JSR PC,OCTP ;PRINT SOFT ERROR NUMBER
MOV #MSG75,R4
JSR PC,TTOUT ;PRINT HARD RD ERROR MESSG
MOV UNP,R4
MOV BDRTY1(R4),R3
JSR PC,OCTP ;PRINT HARD RD ERROR COUNT
MOV #MSG66,R4
JSR PC,TTOUT ;PRINT DATA ERROR TAG
MOV UNP,R4
MOV DATER1(R4),R3
JSR PC,OCTP ;PRINT DATA ERROR NUMBER
JSR PC,BTPRT ;PRINT BAD TAPE STATS
JSR PC,CKSWR ;CHECK FOR TG
RTS PC ;RETURN

```

```

7838
7839
7840
7841 015606 005037 000754
7842 015612 012704 024526
7843 015616 004737 020530
7844 015622 013704 000734
7845 015626 016437 002714 000762
7846 015634 017703 163122
7847 015640 000241
7848 015642 006003
7849 015644 004737 020716
7850 015650 012704 024530
7851 015654 004737 020530
7852 015660 005777 163076
7853 015664 001001
7854 015666 000207
7855 015670 013701 000762
7856 015674 005721
7857 015676 005000
7858 015700 010003
7859 015702 000241
7860 015704 006003
7861 015706 004737 020716
7862 015712 012704 022513
7863 015716 105724
7864 015720 004737 020530
7865 015724 011103
7866 015726 004737 020716
7867 015732 012704 022521
7868 015736 004737 020530
7869 015742 062701 000040
7870 015746 012103
7871 015750 004737 020716
7872 015754 162701 000040
7873 015760 005720
7874 015762 020077 162774
7875 015766 001405
7876 015770 012704 024526
7877 015774 004737 020530
7878 016000 000737
7879 016002 005737 000772
7880 016006 001002
7881 016010 004737 016016
7882 016014 000207
7883
7884
7885
7886 016016 012703 000041
7887 016022 013704 000762
7888 016026 005024
7889 016030 005303
7890 016032 001375
7891 016034 000207

;BAD TAPE STATISTICS PRINT*****
BTPRT: CLR RTYFL
MOV #MSG67,R4
JSR PC,TTOUT ;DO CR/LF
MOV UNP,R4
MOV BTADDR(R4),BTPT ;SET TABLE POINTER
MOV @BTPT,R3
CLC
ROR R3 ;CORRECT NUMBER
JSR PC,OCTP ;PRINT NUMBER OF BAD SPOTS
MOV #MSG68,R4
JSR PC,TTOUT ;PRINT BAD TAPE TAG
TST @BTPT ;SEE IF ANY BAD SPOTS
BNE BTOVO ;IF SO: BR
RTS PC
BTOVO: MOV BTPT,R1 ;SET TABLE POINTER
TST (R1)+
CLR R0
BTOV1: MOV R0,R3
CLC
ROR R3 ;R3=R3/2 FOR CORRECT NUMBER
JSR PC,OCTP ;PRINT ENTRY NUMBER
MOV #MSG13,R4
TSTB (R4)+ ;SKIP CR/LF
JSR PC,TTOUT ;PRINT BLOCK NUMBER TAG
MOV (R1),R3
JSR PC,OCTP ;PRINT BLOCK NUMBER
MOV #MSG14,R4
JSR PC,TTOUT ;PRINT RECORD NUMBER TAG
ADD #40,R1
MOV (R1)+,R3
JSR PC,OCTP ;PRINT RECORD NUMBER
SUB #40,R1 ;RESET POINTER FOR BLOCK NUMBER
TST (R0)+
CMP R0,@BTPT ;SEE IF DONE
BEQ BTOV2
MOV #MSG67,R4
JSR PC,TTOUT ;DO CR/LF
BR BTOV1 ;CONTINUE
BTOV2: TST BTSTF ;SEE IF STAT ONLY PRINT
BNE BTOVX ;IF SO: BR
JSR PC,BTCLR ;CLEAR TABLE
RTS PC ;RETURN

;CLEAR BAD TAPE TABLE
BTCLR: MOV #41,R3 ;SET SIZE OF TABLE
MOV BTPT,R4 ;SET POINTER
BTCLR1: CLR (R4)+ ;CLEAR ENTRY
DEC R3 ;DONE?
BNE BTCLR1 ;IF NOT: BR
RTS PC ;RETURN
  
```

```

7893
7894
7895
7896
7897
7898
7899
7900
7901
7902
7903
7904
7905
7906
7907
7908
7909
7910
7911
7912
7913
7914
7915
7916
7917 016036 013703 000622
7918 016042 004737 022120
7919 016046 005037 000742
7920 016052 005403
7921 016054 005737 000756
7922 016060 001413
7923 016062 012703 000002
7924 016066 005737 001010
7925 016072 001401
7926 016074 005303
7927 016076 032777 000004 162476 1$:
7928 016104 001401
7929 016106 005003
7930 016110 032777 000004 162464 EROA:
7931 016116 001404
7932 016120 062703 026176
7933 016124 000137 016134
7934 016130 062703 032210
7935 016134 010337 016740
7936 016140 020377 162442
7937 016144 001105
7938 016146 017703 162432
7939 016152 001102
7940 016154 005037 013706
7941 016160 032777 000004 162414
7942 016166 001045
7943 016170 032777 000020 162402
7944 016176 001041
7945 016200 013737 013700 013704
7946 016206 005737 000756
7947 016212 001404
7948 016214 000432

```

```

*****
;READ/WRITE STATUS CHECK SUBROUTINE:
;
;THIS SUBROUTINE IS USED TO PERFORM A CHECK
;OF THE TAPE STATUS REGISTER FOR ERRORS AND
;TO ASSURE A CORRECT CURRENT MEMORY ADDRESS
;AND CHARACTER COUNT AT THE END OF EACH TAPE
;OPERATION (READ OR WRITE).
;IF A STATUS ERROR IS INDICATED BY BIT FIFTEEN (15)
;OF THE COMMAND REGISTER BEING SET, THEN AN ERROR
;HEADER CONSISTING OF UNIT NUMBER, BLOCK NUMBER,
;RECORD NUMBER, RECORD SIZE, AND TYPE OF ERROR
;WILL BE PRINTED FOLLOWED BY THE CONTENTS OF
;THE COMMAND REGISTER AND STATUS REGISTER PLUS
;THE CURRENT MEMORY ADDRESS AND CHARACTER COUNT.
;IF NO STATUS ERROR IS INDICATED, THE CHARACTER COUNT
;AND CURRENT MEMORY ADDRESS ARE BOTH CHECKED AND
;THE ENTIRE PRINT OUT IS DONE IF EITHER IS IN ERROR.
;ERROR PRINT OUTS MAY BE DISALLOWED BY SETTING CONSOLE
;SWITCH TEN (10) TO A ONE (1).
;THE PROGRAM MAY BE HALTED ON ANY ERROR BY SETTING
;CONSOLE SWITCH FIFTEEN TO A ONE (1).
*****

```

```

ERCHK. MOV CARCNT,R3 ;GET CHARACTER COUNT
JSR PC,CKSWR ;CHECK FOR 1G
CLR SERFL ;CLEAR STATUS ERROR FLAG
NEG R3
TST TMFLG ;A TM OPERATION?
BEQ EROA ;IF NOT: BR
MOV #2,R3
TST STCDFL ;SEE IF 7 TRK CORE DUMP
BEQ 1$ ;IF NOT: BR
DEC R3 ;SET TO ONE CHARACTER
BIT #4,QMTC ;SEE IF A WRITE TM?
BEQ EROA ;IF NOT: BR
CLR R3 ;ELSE CLEAR R3
BIT #4,QMTC ;SEE IF WRITE OP
BEQ ERO
ADD #WDATA,R3
JMP ER1
ERO: ADD #RDATA,R3 ;ADD START OF BUFFER
ER1: MOV R3,CADR ;SAVE EXPT ADDRESS
CMP R3,QMDA ;SEE IF ADDRESS OK
BNE ER2 ;IF NOT: BR
MOV QMWC,R3 ;GET CHARACTER COUNT
BNE ER2 ;IF NOT ZERO: BR
CLR LRCPT ;CLEAR LPC PRINT FLAG
JIT #4,QMTC ;A WRITE OP?
BNE ER1B ;IF SO: BR
BIT #20,QMTC ;SEE IF SEVEN TRACK DRIVE
BNE ER1B ;IF SO: BR
MOV EXLRC,LRCV ;SET FOR EXPECTED LPC
TST TMFLG ;IS IT A TM?
BEQ 1$ ;IF NOT: BR
BR ER1B

```

7949	016216	012737	000023	013704		MOV	#23, LRCSV	;USE TM LPC
7950	016224	013704	000616		1\$:	MOV	UDES, R4	;GET UNIT DESCRIPTION
7951	016230	042704	117777			BIC	#117777, R4	;MASK DENSITY
7952	016234	022704	060000			CMP	#60000, R4	;SEE IF9 TRK DENSITY AT 800 BPI
7953	016240	001020				BNE	ER1B	;IF NOT: BR
7954	016242	017737	162342	013702		MOV	QMTD, ACTLRC	;GET ACTUAL LPC
7955	016250	032777	020000	162374		BIT	#20000, QSWR	;SEE IF NO DATA CHECK
7956	016256	001011				BNE	ER1B	;IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
7957	016260	005237	013706			INC	LRCPT	;SET LPC PRINT FLAG
7958	016264	042737	177000	013702		BIC	#177000, ACTLRC	;JUST 9 BITS
7959	016272	023737	013702	013704		CMP	ACTLRC, LRCSV	;DOES ACTUAL AGREE WITH EXPECTED?
7960	016300	001027				BNE	ER2	;IF NOT: BR
7961	016302	032777	100000	162272	ER1B:	BIT	#100000, QMTC	;SEE IF HAVE ERROR
7962	016310	001002				BNE	1\$;IF SO: BR
7963	016312	000137	016720			JMP	EREX	
7964	016316	017737	162256	000740	1\$:	MOV	QMTS, ERSV	;GET STATUS
7965	016324	005737	000756			TST	TMFLG	;A TM OPERATION?
7966	016330	001404				BEQ	ER1A	;IF NOT: BR
7967	016332	042737	042125	000740		BIC	#42125, ERSV	;IGNORE TM INDICATOR AND WRL
7968	016340	001567				BEQ	EREX	;IF NO OTHER ERRORS: BR
7969	016342	005737	000760		ER1A:	TST	EOTREC	;IS IT EOT
7970	016346	100004				BPL	ER2	;IF NOT: BR
7971	016350	042737	032125	000740		BIC	#32125, ERSV	;IGNORE EOT INDICATOR
7972	016356	001560				BEQ	EREX	;IF NO OTHER ERRORS: BR
7973	016360	005237	000742		ER2:	INC	SERFL	;SET STATUS ERROR FLAG
7974	016364	032777	002000	162260		BIT	#20000, QSWR	;SEE IF SHOULD PRINT ERRORS
7975	016372	001411				BEQ	ER3	;IF SO: BR
7976	016374	005737	000774			TST	RTYFL	;SEE IF READ RETRY
7977	016400	001404				BEQ	ER2A	;IF NOT: BR
7978	016402	022737	000003	000752		CMP	#3, RTCNT	;SEE IF LAST RETRY
7979	016410	001402				BEQ	ER3	;IF SO: BR
7980	016412	000137	016664		ER2A:	JMP	EREXO	;ELSE EXIT
7981	016416	005237	000732		ER3:	INC	PFLG	;SET PRINT FLAG
7982	016422	004737	017506			JSR	PC, PAPRT	;PRINT HEADER
7983	016426	013704	000716		ER3A:	MOV	EMADDR, R4	
7984	016432	004737	020530		ER3B:	JSR	PC, TOUT	;PRINT ERROR HEADER
7985	016436	004737	016742			JSR	PC, FRPRT	;PRINT F OR R
7986	016442	005037	000712			CLR	TEMP3	
7987	016446	012704	022750			MOV	#MSG23, R4	
7988	016452	004737	020530			JSR	PC, TOUT	;PRINT COMMAND HEADER
7989	016456	017703	162120			MOV	QMT, R3	
7990	016462	000303			ER7:	SWAB	R3	;POSITION MOST SIGNIFICANT
7991	016464	004737	021144			JSR	PC, DOUT	;PRINT REGISTER
7992	016470	000303				SWAB	R3	;POSITION LEAST SIGNIFICANT
7993	016472	004737	021144			JSR	PC, DOUT	;PRINT REGISTER
7994	016476	005737	000712			TST	TEMP3	;SEE IF PRINTED STATUS REGISTER
7995	016502	001012				BNE	ER10	;IF SO: BR
7996	016504	005237	000712			INC	TEMP3	;SET FLAG
7997	016510	012704	023135			MOV	#MSG30, R4	
7998	016514	004737	020530			JSR	PC, TOUT	;PRINT STATUS HEADER
7999	016520	017703	162054			MOV	QMTS, R3	;LOAD STATUS REGISTER
8000	016524	000137	016462			JMP	ER7	;GO PRINT STATUS
8001	016530	012704	023641		ER10:	MOV	#MSG46, R4	
8002	016534	004737	020530			JSR	PC, TOUT	;PRINT CHARACTER COUNT HEADER
8003	016540	017703	162040			MOV	QMWC, R3	
8004	016544	005403				NEG	R3	;SET TO TRUE VALUE

```

8005 016546 004737 020716 JSR PC, OCTP ;PRINT CHARACTER COUNT
8006 016552 012704 023646 MOV #MSG47, R4
8007 016556 004737 020530 JSR PC, TOUT ;PRINT ADDRESS HEADER
8008 016562 017703 162020 MOV #MDA, R3
8009 016566 004737 020716 JSR PC, OCTP ;PRINT ADDRESS
8010 016572 012737 000255 000702 MOV #255, T0B
8011 016600 004737 020670 JSR PC, T0G ;PRINT /
8012 016604 013703 016740 MOV CADER, R3
8013 016610 004737 020716 JSR PC, OCTP ;PRINT EXPT ADDRESS
8014 016614 005737 013706 TST LRCPT ;WAS LPC CHECKED?
8015 016620 001421 BEQ EREXO ;IF NOT: BR
8016 016622 012704 025245 MOV #MSG83, R4
8017 016626 004737 020530 JSR PC, TOUT ;PRINT LPC TAG
8018 016632 013703 013702 MOV ACTLRC, R3
8019 016636 004737 020716 JSR PC, OCTP ;PRINT ACTUAL LPC
8020 016642 012737 000255 000702 MOV #255, T0B
8021 016650 004737 020670 JSR PC, T0G ;PRINT -
8022 016654 013703 013704 MOV LRCV, R3
8023 016660 004737 020716 JSR PC, OCTP ;PRINT EXPECTED LPC
8024 016664 032777 100000 161760 EREXO: BIT #100000, ASWR ;SEE IF STOP ON ERROR
8025 016672 001412 BEQ EREX ;IF NOT: BR
8026 016674 000000 HALT
8027 016676 005737 000732 TST PFLG ;SEE IF PRINT
8028 016702 001006 BNE EREX ;IF SO: BR
8029 016704 032777 002000 161740 BIT #2000, ASWR ;SEE IF SHOULD PRINT
8030 016712 001002 BNE EREX ;IF NOT: BR
8031 016714 000137 016416 JMP ER3 ;PRINT ERROR
8032 016720 004737 022120 EREX: JSR PC, CKSWR ;GO TEST FOR TG
8033 016724 005037 000732 CLR PFLG ;CLEAR FLAG
8034 016730 017737 161644 000740 MOV #MTS, ERSAV ;SAVE STATUS REGISTER
8035 016736 000207 RTS ;RETURN
8036 016740 000000 CADER: 0 ;EXPT ADDRESS SAVE LOCATION
8037
8038 ;*****
8039 ;F FOR FORWARD/R FOR REVERSE PRINT SUBROUTINE:
8040 ;
8041 ;THIS SUBROUTINE IS USED TO PRINT OUT THE
8042 ;TAPE DIRECTION USED WHEN ANY ERROR IS
8043 ;DETECTED IN STATUS OF READ OR WRITE, DATA, OR
8044 ;SPACING OPERATIONS.
8045 ;*****
8046
8047 016742 032777 000004 161632 FRPRT: BIT #4, AMTC ;SEE IF WRITE COMMAND
8048 016750 001015 BNE FREX ;IF SO: BR
8049 016752 032737 010000 000626 BIT #10000, RDCMD ;SEE IF READ REVERSE
8050 016760 001405 BEQ FRO ;IF NOT: BR
8051 016762 012704 022564 MOV #MSG17, R4
8052 016766 004737 020530 JSR PC, TOUT ;PRINT R
8053 016772 000404 BR FREX
8054 016774 012704 022556 FRO: MOV #MSG16, R4
8055 017000 004737 020530 JSR PC, TOUT ;PRINT F
8056 017004 000207 FREX: RTS ;EXIT

```

8058
8059
8060
8061
8062
8063
8064
8065
8066
8067
8068
8069
8070
8071
8072
8073
8074
8075
8076
8077
8078
8079
8080
8081
8082
8083
8084
8085
8086
8087
8088
8089
8090
8091
8092
8093
8094
8095
8096
8097
8098
8099
8100
8101
8102
8103
8104
8105
8106
8107
8108
8109
8110
8111
8112
8113

017006 005037 000706
017012 013704 000602
017016 005204
017020 113714 000617
017024 032777 000200 161550
017032 001035
017034 005237 000706
017040 001371
017042 004737 017506
017046 032777 000004 161526
017054 001405
017056 012704 022452
017062 004737 020530
017066 000406
017070 012704 022457
017074 004737 020530
017100 004737 016742
017104 012704 023007
017110 004737 020530
017114 000000
017116 005037 000734
017122 000137 003264
017126 000240
017130 000240
017132 005037 000706
017136 032777 000100 161434
017144 001013

```
*****  
:TAPE COMMAND EXECUTE SUBROUTINE:  
:  
:THIS SUBROUTINE IS USED TO EXECUTE THE  
:MAG TAPE COMMAND DESCRIBED BY THE READ  
:OR WRITE ROUTINE. THE FINAL COMMAND IS  
:SENT TO THE DEVICE REGISTER ALONG WITH THE  
:INTERRUPT ENABLE AND GO BITS.  
:ONCE THE COMMAND IS ISSUED, AN INTERRUPT  
:TIMER IS STARTED AND IF NO INTERRUPT IS RETURNED  
:BEFORE TIME OUT OCCURS, AN ERROR WILL BE  
:PRINTED AND THE PROGRAM STOPPED. TESTING MAY  
:BE RESUMED BY PRESSING THE CONTINUE BUTTON.  
:TWO INTERRUPT HANDLERS ARE USED, ONE FOR MAG TAPE  
:AND ANOTHER FOR TELETYPE (TTY).  
:UPON RECEIPT OF A MAG TAPE INTERRUPT, HOUSEKEEPING  
:IS PERFORMED AND CONTROL RETURNED TO THE CALLING  
:ROUTINE (READ, WRITE, ETC).  
:RECEIPT OF A TTY INTERRUPT WILL CAUSE THE  
:PROGRAM TO CHECK FOR ENTRY OF A CNTRL C CHARACTER.  
:IF NOT CNTRL C, THEN CONTINUATION OF WAIT FOR MAG  
:TAPE INTERRUPT IS RETURNED. IF, HOWEVER, THE TTY  
:INTERRUPT WAS CAUSED BY ENTRY OF A CNTRL C,  
:THEN AT THIS TIME REQUESTS FOR NEW STALL VALUES  
:ARE PRINTED AND THE RESPONSES ENTERED. RESUMPTION  
:OF TAPE INTERRUPT WAIT IS THEN RESUMED.  
*****  
TAPG: CLR TEMP1  
MOV MTC, R4 ;GET COMD REGISTER ADDRESS  
INC R4 ;BUMP TO HIGH BYTE  
MOV# UDES+1, (R4) ;LOAD UNIT DESCRIPTION  
TAPG0: BIT #200, MTC ;SEE IF HAVE READY  
BNE TAPG3 ;IF SO: BR  
INC TEMP1 ;SEE IF TIMED OUT  
BNE TAPG0 ;WAIT FOR READY  
JSR PC, PAPRT ;PRINT CYCLE NUMBER  
BIT #4, MTC ;SEE IF WRITE OP  
BEQ TAPG1 ;IF NOT: BR  
MOV #MSG5, R4  
JSR PC, TTOUT ;PRINT WRITE ERR  
BR TAPG2  
TAPG1: MOV #MSG6, R4  
JSR PC, TTOUT ;PRINT READ ERR  
JSR PC, FPRT ;PRINT F OR R  
TAPG2: MOV #MSG25, R4  
JSR PC, TTOUT ;PRINT NO READY ERR  
TAPG2A: HALT  
CLR UNP ;RESET UNIT POINTER  
JMP STAUTO ;RESTART  
TAPG3: NOP  
NOP  
CLR TEMP1 ;SET DELAY  
BIT #100, MTS ;SEE IF SELR  
BNE Z$ ;IF SO: BR
```

F07

8114	017146	005237	000706		INC	TEMP1	
8115	017152	001371			BNE	1\$;DELAY
8116	017154	004737	017506		JSR	PC,PAPRT	;PRINT HEADER
8117	017160	012704	026016		MOV	#MSG95,R4	
8118	017164	004737	020530		JSR	PC,TTOUT	;PRINT SELR LOST
8119	017170	000137	020050		JMP	DRPDRV	;GO DROP DRIVE
8120	017174	005077	161450	2\$:	CLR	QPSW	;SET TO PRIORITY 0
8121	017200	000240			NOP		
8122	017202	000240			NOP		
8123	017204	052777	000101	161370	BIS	#101,QMTC	;SET INTERRUPT ENABLE AND GO
8124	017212	012704	020000		MOV	#20000,R4	
8125	017216	005003			CLR	R3	
8126	017220	032777	000004	161354	BIT	#4,QMTC	;SEE IF WRITE OP
8127	017226	001042			BNE	TAPG8	;IF SO GO TO WRITE EOT WATCH
8128	017230	005303		TAPG4:	DEC	R3	
8129	017232	001376			BNE	TAPG4	
8130	017234	005304			DEC	R4	;SEE IF TIMED OUT
8131	017236	001374			BNE	TAPG4	
8132	017238	012777	000340	161402	TAPG5: MOV	#340,QPSW	;RESET PRIORITY
8133	017240	042777	000100	161326	BIC	#100,QMTC	;CLEAR INTERRUPT ENABLE
8134	017242	032777	002000	161370	BIT	#2000,QSWR	;SEE IF SHOULD PRINT ERRORS
8135	017262	001014			BNE	TAPG6	;IF NOT: BR
8136	017264	004737	017506		JSR	PC,PAPRT	;PRINT CYCLE NUMBER
8137	017270	013704	000716		MOV	EMADDR,R4	
8138	017274	004737	020530		JSR	PC,TTOUT	;PRINT ERROR OP
8139	017300	004737	016742		JSR	PC,FRPRT	;PRINT F OR R
8140	017304	012704	022757		MOV	#MSG24,R4	
8141	017310	004737	020530		JSR	PC,TTOUT	;PRINT NO INTERRUPT
8142	017314	032777	100000	161330	TAPG6: BIT	#100000,QSWR	;SEE IF SHOULD HALT ON ERROR
8143	017322	001401			BEQ	TAPG7	;IF NOT: BR
8144	017324	000000			HALT		
8145	017326	000240		TAPG7:	NOP		
8146	017330	000177	161370		JMP	QRTN	;RETURN TO CALLING ROUTINE
8147	017334	032777	000010	161236	TAPG8: BIT	#10,QMTS	;SEE IF SOWN SET
8148	017342	001012			BNE	2\$;IF SO: BR
8149	017344	032777	002000	161226	BIT	#2000,QMTS	;SEE IF EOT REACHED
8150	017352	001404			BEQ	1\$;IF NOT: BR
8151	017354	052737	000001	017402	BIS	#1,WEOTF	;SET EOT FLAG
8152	017362	000402			BR	2\$	
8153	017364	005037	017402	1\$:	CLR	WEOTF	;CLEAR FLAG
8154	017370	005303		2\$:	DEC	R3	
8155	017372	001360			BNE	TAPG8	;DELAY
8156	017374	005304			DEC	R4	
8157	017376	001356			BNE	TAPG8	;DELAY
8158	017400	000717			BR	TAPG5	
8159	017402	000000		WEOTF:	0		
8160							


```

8162
8163 ;TTY INTERRUPT HANDLER*****
8164
8165 017404 012777 000340 161236 TTINT: MOV #340, @PSW ;RESET PSW
8166 017412 005077 161240 CLR @TKS ;CLEAR TTY STATUS
8167 017416 122777 000203 161234 CMPB #203, @TKB ;SEE IF CONT C
8168 017424 001404 BEQ TTINT0 ;IF SO: BR
8169 017426 004737 022120 JSR PC, CKSWR ;GO CHECK FOR ↑G
8170 017432 000240 NOP
8171 017434 000002 RTI ;ELSE RETURN
8172 017436 010037 000712 TTINT0: MOV RO, TEMP3 ;SAVE RO(REC CNTR)
8173 017442 004737 012164 JSR PC, TINP4 ;GO GET STALL VALUES
8174 017446 013700 000712 MOV TEMP3, RO ;RESTORE RO(REC CNTR)
8175 017452 005077 161202 CLR @TKB ;CLEAR TTY BUFFER
8176 017456 012777 000100 161172 MOV #100, @TKS ;RESET INTERRUPT ENABLE
8177 017464 000002 RTI ;RETURN
8178
8179 ;MAG TAPE INTERRUPT HANDLER*****
8180
8181 017466 022626 MTINT: CMP (SP)+, (SP)+ ;RESET STACK POINTER
8182 017470 042777 000100 161104 BIC #100, @MTC ;RESET INTERRUPT ENABLE
8183 017476 000240 NOP
8184 017500 000240 NOP
8185 017502 000177 161216 JMP @RTRN ;RETURN
8186
8187 ;*****
8188 ;ERROR HEADER PRINT SUBROUTINE:
8189 ;
8190 ;THIS ROUTINE IS USED TO PRINT OUT A HEADER
8191 ;WITH EACH ERROR MESSAGE. THE PRINT IS IN TWO
8192 ;LINES AND CONTAINS THE FOLLOWING INFORMATION.
8193 ;LINE 1: UNIT NUMBER, DATA PATTERN NUMBER
8194 ;LINE 2: CURRENT BLOCK NUMBER, RECORD NUMBER IN
8195 ;WHICH THE ERROR OCCURED PLUS THE TOTAL NUMBER
8196 ;OF RECORDS IN THIS BLOCK, THE RECORD SIZE (NUMBER
8197 ;OF CHARACTERS), AND THE ERROR TYPE (READ, WRITE, SPACE, ETC)
8198 ;PLUS THE TAPE DIRECTION (FORWARD OR REVERSE).
8199 ;ALL NUMBERS ARE IN OCTAL.
8200 ;*****
8201
8202 017506 012704 022476 PAPRT: MOV #MSG11, R4
8203 017512 004737 020530 JSR PC, TOUT ;PRINT UNIT HEADER
8204 017516 013703 000616 MOV UDES, R3
8205 017522 000303 SWAB R3
8206 017524 042703 177770 BIC #177770, R3
8207 017530 004737 020716 JSR PC, OCTP ;PRINT UNIT NUMBER
8208 017534 012704 025556 MOV #MSG90, R4
8209 017540 004737 020530 JSR PC, TOUT ;PRINT DENSITY TAG
8210 017544 005003 CLR R3
8211 017546 032737 020000 000616 BIT #20000, UDES ;SEE IF BIT 1 OF DENSITY=1
8212 017554 001401 BEQ 1$ ;IF NOT: BR
8213 017556 005203 INC R3 ;ELSE SET BIT 1
8214 017560 032737 040000 000616 1$: BIT #40000, UDES ;SEE IF BIT 2 OF DENSITY=1
8215 017566 001402 BEQ 2$ ;IF NOT: BR
8216 017570 052703 000002 BIS #2, R3 ;ELSE SET BIT 2
8217 017574 004737 020716 2$: JSR PC, OCTP ;PRINT DENSITY SETTING
    
```

H07

8218	017600	012704	025564			MOV	#MSG91,R4	
8219	017604	004737	020530			JSR	PC,TTOUT	;PRINT PARITY TAG
8220	017610	005003				CLR	R3	
8221	017612	032737	004000	000616		BIT	#4000,UDES	;SEE IF EVEN PARITY
3222	017620	001401				BEQ	3\$;IF NOT: BR
8223	017622	005203				INC	R3	;ELSE SET TO A ONE
8224	017624	004737	020716		3\$:	JSR	PC,OCTP	;PRINT PARITY
8225	017630	012704	025321			MOV	#MSG86,R4	
8226	017634	004737	020530			JSR	PC,TTOUT	;PRINT PATTRN TAG
8227	017640	032777	000400	161004		BIT	#400,ASWR	;SEE IF RANDOM DATA
8228	017646	001406				BEQ	PAPRTB	;IF NOT: BR
8229	017650	012737	000122	000702	PAPRTA:	MOV	#122,TOB	
8230	017656	004737	020670			JSR	PC,TOG	;PRINT R
8231	017662	000412				BR	PAPRTD	
8232	017664	005737	021624		PAPRTB:	TST	ASEQF	;SEE IF AUTO SEQ
8233	017670	001403				BEQ	PAPRTC	;IF NOT: BR
8234	017672	005737	000624			TST	PATRN	;SEE IF AUTO RANDOM
8235	017676	100764				BMI	PAPRTA	;IF SO: BR
8236	017700	013703	000624		PAPRTC:	MOV	PATRN,R3	
8237	017704	004737	020716			JSR	PC,OCTP	;PRINT PATTRN NUMBER
8238	017710	012704	022513		PAPRTD:	MOV	#MSG13,R4	
8239	017714	004737	020530			JSR	PC,TTOUT	;PRINT BLOCK NO. HEADER
8240	017720	013703	000720			MOV	BLCNTR,R3	
8241	017724	004737	020716			JSR	PC,OCTP	;PRINT NUMBER
8242	017730	012704	022521			MOV	#MSG14,R4	
8243	017734	004737	020530			JSR	PC,TTOUT	;PRINT REC NO. HEADER
8244	017740	010003				MOV	RO,R3	
8245	017742	032777	000004	160632		BIT	#4,AMTC	;SEE IF WRITE OPERATION
8246	017750	001000				BNE	PAPRT1	;IF SO: BR
8247	017752	013703	000620		PAPRT1:	MOV	RCNT,R3	
8248	017756	160003				SUB	RO,R3	;GET RECORD NUMBER
8249	017760	005203				INC	R3	
8250	017762	004737	020716		PAPRT2:	JSR	PC,OCTP	;PRINT RECORD NUMBER
8251	017766	012737	000055	000702		MOV	#55,TOB	;LOAD DASH (-)
8252	017774	004737	020670			JSR	PC,TOG	;PRINT DASH (-)
8253	020000	013703	000620			MOV	RCNT,R3	
8254	020004	004737	020716			JSR	PC,OCTP	;PRINT RECORD COUNT
8255	020010	012704	022464			MOV	#MSG7,R4	
8256	020014	004737	020530			JSR	PC,TTOUT	;PRINT RECORD SIZE HEADER
8257	020020	013703	000622			MOV	CARCNT,R3	;GET CHARACTER COUNT
8258	020024	005303				DEC	R3	
8259	020026	005103				COM	R3	;REMOVE TWOS COMPLEMENT
8260	020030	004737	020716			JSR	PC,OCTP	;PRINT RECORD SIZE
8261	020034	012737	000001	000726		MOV	#1,HDRFL	;SET HEADER FLAG
8262	020042	004737	022120			JSR	PC,CKSWR	;TEST FOR 1G
8263	020046	000207				RTS	PC	;RETURN
9264								

```

8266
8267
8268
8269 020050 000240
8270 020052 012777 010000 160522
8271 020060 012704 025572
8272 020064 004737 020530
8273 020070 013703 000616
8274 020074 000303
8275 020076 042703 177770
8276 020102 004737 020716
8277 020106 012704 025616
8278 020112 004737 020530
8279 020116 013700 000734
8280 020122 052760 100200 001012
8281 020130 005337 004716
8282 020134 004737 015370
8283 020140 005237 001006
8284 020144 123737 001006 004717
8285 020152 103406
8286 020154 012704 026045
8287 020160 004737 020530
8288 020164 000137 004640
8289 020170 000240
8290 020172 005000
8291 020174 032760 100200 001012
8292 020202 001414
8293 020204 062700 000002
8294 020210 022760 177777 001012
8295 020216 001366
8296 020220 012704 025710
8297 020224 004737 020530
8298 020230 000137 004622
8299 020234 000137 004072
8300
8301
8302
8303
8304
8305
8306
8307
8308
8309 020240 063737 000676 000672
8310 020246 063737 000672 000676
8311 020254 023701 000676
8312 020260 101367
8313 020262 020237 000676
8314 020266 101364
8315 020270 000207
8316

```

```

;DROP UNIT SUBROUTINE*****
DRPDRV: NOP
MOV #10000, R2MTC ;POWER CLEAR CONTROLLER
MOV #MSG92, R4
JSR PC, TTOUT ;PRINT UNIT DROPPED
MOV UDES, R3 ;GET UNIT DESCRIPTION
SWAB R3
BIC #177770, R3 ;MASK UNIT NUMBER
JSR PC, OCTP ;PRINT DROPPED UNIT NUMBER
MOV #MSG93, R4
JSR PC, TTOUT ;PRINT REST OF MSG
MOV UNP, R0 ;SET UNIT POINTER
BIS #100200, UN1(R0) ;SET DROPPED FLAG
DEC REOTC ;DECREMENT EOT UNIT COUNTER
JSR PC, PRSTAT ;PRINT CURRENT STATS
INC DUCTR ;BUMP DROPPED UNIT COUNTER
CMPB DUCTR, REOTC+1 ;SEE IF DROPPED ALL UNITS
BLO 1$ ;IF NOT: BR
MOV #MSG96, R4
JSR PC, TTOUT ;PRINT ALL DROPPED: STOP
JMP REOT9 ;GO TO END ROUTINE
1$: NOP
CLR R0
2$: BIT #100200, UN1(R0) ;SEE IF ANY DRIVES LEFT IN THIS PASS
BEQ 3$ ;IF SO: BR
ADD #2, R0 ;BUMP POINTER
CMP #-1, UN1(R0) ;SEE IF LAST ENTRY
BNE 2$ ;IF NOT: BR
MOV #MSG94, R4
JSR PC, TTOUT ;PRINT NO MORE UNITS
JMP REOT8 ;GO TO END OF PASS ROUTINE
3$: JMP START7 ;GO TO NEXT UNIT

```

```

;*****
;RANDOM NUMBER GENERATOR SUBROUTINE:
;
;THIS SUBROUTINE IS USED TO GENERATE THE RANDOM
;NUMBERS REQUIRED FOR USE AS RANDOM DATA,
;RECORD COUNT, AND CHARACTER COUNT.
;*****
RANG: ADD RANS, RANBAS
ADD RANBAS, RANS ;GET NEW NUMBER
CMP RANS, R1 ;SEE IF NUMBER TOO BIG
BHI RANG ;IF SO: BR
CMP R2, RANS ;SEE IF NUMBER TOO SMALL
BHI RANG ;IF SO: BR
RTS PC ;EXIT

```

8318
8319
8320
8321
8322
8323
8324
8325
8326
8327
8328
8329
8330
8331
8332
8333
8334
8335
8336
8337
8338
8339
8340
8341
8342
8343
8344
8345
8346
8347
8348
8349
8350
8351
8352
8353
8354
8355
8356
8357
8358
8359
8360
8361
8362
8363
8364
8365
8366
8367
8368
8369

020272 005037 000706
020276 005000
020300 004737 020456
020304 122737 000215 000704
020312 001005
020314 005737 000706
020320 001446
020322 000137 020414
020326 122737 000260 000704
020334 101402
020336 000137 020440
020342 122737 000270 000704
020350 101002
020352 000137 020440
020356 005237 000706
020362 000241
020364 006100
020366 000241
020370 006100
020372 000241
020374 006100
020376 042737 177770 000704
020404 053700 000704
020410 005301
020412 001332
020414 020002
020416 101402
020420 000137 020440
020424 020300
020426 101402
020430 000137 020440
020434 010015
020436 000207

```
*****  
: TTY ENTRY SUBROUTINE:  
: THIS SUBROUTINE IS USED BY THE TEST CONDITION  
: ENTRY ROUTINE TO READ THE RESPONSE ENTERED  
: AT THE TTY AND CHECK THEM FOR LEGALITY AND  
: LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL  
: (0-7) AND MUST FALL WITHIN THE LIMITS SET BY  
: THE CALLING ROUTINE.  
: IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,  
: A QUESTION MARK IS TYPED (?) AND THE RESPONSE  
: MAY BE REENTERED.  
: ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND  
: MAY BE TERMINATED AT LESS THAN SIX BY TYPING A  
: CARRIAGE RETURN  
*****  
TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG  
CLR RO  
TTR0: JSR PC,TIN ;GO READ CHARACTER  
CMPB #215,TIB ;SEE IF CR  
BNE TTR1 ;IF NOT: BR  
TST TEMP1 ;SEE IF FIRST CHARACTER  
BEQ TTR5 ;IF SO: BR  
JMP TTR2 ;ELSE GO LOAD VALUE  
TTR1: CMPB #260,TIB ;SEE IF CHAR IS LESS THAN 0  
BLOS TTR1A ;IF NOT: BR  
JMP TTR2 ;ELSE GO TO ERROR  
TTR1A: CMPB #270,TIB ;SEE IF CHAR IS GREATER THAN 7  
BHI TTR1B ;IF NOT: BR  
JMP TTR2 ;ELSE GO TO ERROR  
TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG  
CLC  
ROL RO  
CLC  
ROL RO ;SHIFT 3 LEFT  
CLC  
ROL RO  
TTR2: BIC #177770,TIB ;STRIP ASCII  
BIS TIB,RO ;LOAD CHARACTER  
DEC R1 ;SEE IF DONE  
BNE TTR0 ;IF NOT: BR  
CMP RO,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT  
BLOS TTR3 ;IF NOT: BR  
JMP TTR2 ;ELSE GO TO ERROR  
TTR3: CMP R3,RO ;SEE IF BELOW MINIMUM LIMIT  
BLOS TTR4 ;IF NOT: BR  
JMP TTR2 ;ELSE GO TO ERROR  
TTR4: MOV RO,(R5) ;LOAD VALUE  
TTR5: RTS PC ;EXIT
```

```

8371
8372
8373
8374 020440 012704 023574
8375 020444 004737 020530
8376 020450 162716 000020
8377 020454 000207
8378
8379
8380
8381 020456 005077 160174
8382 020462 005077 160172
8383 020466 005037 000704
8384 020472 005277 160160
8385 020476 105777 160154
8386 020502 100375
8387 020504 017737 160150 000704
8388 020512 105777 160144
8389 020516 100375
8390 020520 113777 000704 160136
8391 020526 000207
8392
8393
8394
8395 020530 112437 000702
8396 020534 122737 000043 000702
8397 020542 001460
8398 020544 122737 000045 000702
8399 020552 001407
8400 020554 122737 000041 000702
8401 020562 001434
8402 020564 004737 020670
8403 020570 000757
8404 020572 112737 000015 000702
8405 020600 004737 020670
8406 020604 012703 000004
8407 020610 005037 000702
8408 020614 004737 020670
8409 020620 005303
8410 020622 001372
8411 020624 112737 000012 000702
8412 020632 004737 020670
8413 020636 105737 001004
8414 020642 100401
8415 020644 000731
8416 020646 005037 001004
8417 020652 000414
8418 020654 112737 000007 000702
8419 020662 004737 020670
8420 020666 000720
8421 020670 105777 157766
8422 020674 100375
8423 020676 113777 000702 157760
8424 020704 000207
8425
8426

```

```

;TTY ENTRY ERROR SUBROUTINE*****
TINER: MOV #MSG43,R4
JSR PC,TTOUT ;PRINT?
SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE
RTS PC ;REDO VALUE ENTRY

;TTY READ SUBROUTINE*****
TTIN: CLR @TKS
CLR @TKB
CLR TIB
INC @TKS
TTIN1: TSTB @TKS
BPL TTIN1
MOV @TKB,TIB
TTIN2: TSTB @TPS
BPL TTIN2
MOVB TIB,@TPB
RTS PC

;TTY OUTPUT SUBROUTINE*****
TTOUT: MOVB (R4)+,TOB
CMPB #43,TOB
BEQ TEX
CMPB #45,TOB
BEQ TCRLF
CMPB #41,TOB
BEQ TBELL
JSR PC,TOG
BR TTOUT
TCRLF: MOVB #15,TOB
JSR PC,TOG
MOV #4,R3
TCRLFA: CLR TOB
JSR PC,TOG
DEC R3
BNE TCRLFA ;DO FILLERS
MOVB #12,TOB
JSR PC,TOG
TSTB RDSW
BMI IS
BR TTOUT
IS: CLR RDSW
BR TEX
TBELL: MOVB #7,TOB
JSR PC,TOG
BR TTOUT
TOG: TSTB @TPS
BPL TOG
MOVB TOB,@TPB
TEX: RTS PC

```

```

;OCTAL OUTPUT SUBROUTINE*****
8428
8429
8430 020706 012737 000001 021142 OCTPE: MOV #1, OFL
8431 020714 000402 BR OCTPE1
8432 020716 005037 021142 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
8433 020722 010304 OCTPE1: MOV R3, R4
8434 020724 001007 BNE OCTP0 ;IF NOT ZERO: BR
8435 020726 005737 021142 TST OFL
8436 020732 001004 BNE OCTP0
8437 020734 004737 021122 JSR PC, OCTPG1 ;ELSE PRINT ZERO
8438 020740 000137 021064 JMP OCTP3 ;SPACE AND EXIT
8439 020744 032704 100000 OCTP0: BIT #100000, R4 ;SEE IF MSD = 1
8440 020750 001406 BEQ OCTP1 ;IF NOT: BR
8441 020752 012704 000001 MOV #1, R4
8442 020756 004737 021100 JSR PC, OCTPG ;PRINT 1
8443 020762 000137 020774 JMP OCTP2
8444 020766 005004 OCTP1: CLR R4
8445 020770 004737 021100 JSR PC, OCTPG ;PRINT 0
8446 020774 010304 OCTP2: MOV R3, R4
8447 020776 006004 ROR R4
8448 021000 006004 ROR R4
8449 021002 006004 ROR R4 ;POSITION DIGIT
8450 021004 006004 ROR R4
8451 021006 000304 SWAB R4
8452 021010 004737 021100 JSR PC, OCTPG ;PRINT DIGIT 2
8453 021014 010304 MOV R3, R4
8454 021016 006004 ROR R4
8455 021020 000304 SWAB R4
8456 021022 004737 021100 JSR PC, OCTPG ;PRINT DIGIT 3
8457 021026 010304 MOV R3, R4
8458 021030 006104 ROL R4
8459 021032 006104 ROL R4
8460 021034 000304 SWAB R4
8461 021036 004737 021100 JSR PC, OCTPG ;PRINT DIGIT 4
8462 021042 010304 MOV R3, R4
8463 021044 006004 ROR R4
8464 021046 006004 ROR R4
8465 021050 006004 ROR R4
8466 021052 004737 021100 JSR PC, OCTPG
8467 021056 010304 MOV R3, R4
8468 021060 004737 021100 JSR PC, OCTPG ;PRINT DIGIT 5
8469 021064 012737 000240 000702 OCTP3: MOV #240, TOB
8470 021072 004737 020670 JSR PC, TOG ;PRINT SPACE
8471 021076 000207 RTS PC ;EXIT
8472 021100 042704 177770 OCTPG: BIC #177770, R4
8473 021104 001004 BNE OCTPG0
8474 021106 005737 021142 TST OFL
8475 021112 001001 BNE OCTPG0
8476 021114 000207 RTS PC
8477 021116 005237 021142 OCTPG0: INC OFL
8478 021122 052704 000260 OCTPG1: BIS #260, R4
8479 021126 010437 000702 MOV R4, TOB
8480 021132 004737 020670 JSR PC, TOG
8481 021136 010304 MOV R3, R4
8482 021140 000207 RTS PC
8483 021142 000000 OFL: 0 ;FIRST CHAR FLAG
    
```

8484
8485
8486
8487
8488
8489
8490
8491
8492
8493
8494
8495
8496
8497
8498
8499
8500
8501
8502
8503
8504
8505
8506
8507
8508
8509
8510
8511

021144 005037 000702
021150 012704 000010
021154 110337 000702
021160 105777 157476
021164 100375
021166 132737 000200 000702
021174 001404
021176 012777 000061 157460
021204 000403
021206 012777 000060 157450
021214 006137 000702
021220 005304
021222 001356
021224 000207

021226 005777 157360
021232 100775
021234 005777 157352
021240 100375
021242 005777 157344
021246 100775
021250 000207

```

;DATA CHARACTER OUTPUT SUBROUTINE*****
DOUT: CLR TOB
      MOV #10, R4 ;SET NUMBER TO PRINT
      MOVB R3, TOB
DOUT1: TSTB @TPS
      BPL DOUT1
      BITB #200, TOB
      BEQ DOUT2
      MOV #061, @TPB
      BR DOUT3
DOUT2: MOV #060, @TPB
DOUT3: ROL TOB
      DEC R4
      BNE DOUT1
      RTS PC

;ASSURE VALID STATUS DELAY SUBROUTINE*****
STDLY: TST @MTRD
      BMI STDLY ;AWAIT TIMER = 0
1$: TST @MTRD
      BPL 1$ ;AWAIT TIMER =1
2$: TST @MTRD
      BMI 2$ ;AWAIT TIMER = 0
      RTS PC ;EXIT
```

```

;AUTO SEQUENCE TEST ROUTINE*****
8513
8514
8515 021252 012704 025034 ASEQ:  MOV  #MSG78,R4
8516 021256 004737 020530      JSR  PC,TTOUT      ;PRINT CONT. REQUEST
8517 021262 013703 021626      MOV  ASEQCF,R3
8518 021266 004737 020716      JSR  PC,OCTP      ;PRINT CURRENT VALUE
8519 021272 012705 021626      MOV  #ASEQCF,R5   ;SET ENTRY ADDRESS
8520 021276 012701 000001      MOV  #1,R1        ;SET SIZE OF ENTRY
8521 021302 012702 000001      MOV  #1,R2        ;SET UPPER LIMIT
8522 021306 005003                CLR  R3            ;SET LOWER LIMIT
8523 021310 004737 020272      JSR  PC,TTR       ;GET INPUT
8524
8525 021314 004737 021630 ASEQ0: JSR  PC,HRDS     ;SELECT HARDWARE CONFIGURATION
8526 021320 012704 025051      MOV  #MSG73,R4
8527 021324 004737 020530      JSR  PC,TTOUT     ;PRINT DIVIDER
8528 021330 012704 025117      MOV  #MSG80,R4
8529 021334 004737 020530      JSR  PC,TTOUT     ;PRINT UNITS NUMBER MESSG.
8530 021340 012700 001012      MOV  #UN1,RO      ;POINT TOP OF DRIVE TABLE
8531 021344 005710                ASEQ2: TST  (RO)    ;SEE IF END
8532 021346 100424                BMI  AMOD1        ;IF SO: BR
8533 021350 011037 000706      MOV  (RO),TEMP1   ;GET UNIT DESCRIPTION
8534 021354 113703 000707      MOV  TEMP1+1,R3   ;POSITION AND
8535 021360 042703 177770      BIC  #177770,R3   ;MASK UNIT NUMBER
8536 021364 004737 020716      JSR  PC,OCTP      ;PRINT DRIVE TABLE
8537 021370 012704 023724      MOV  #MSG51,R4    ;PRESET FOR 9 TRK MSG
8538 021374 032710 020000      BIT  #20000,(RO) ;SEE IF 7 TRK
8539 021400 001002                BNE  1$           ;IF NOT: BR
8540 021402 012704 023715      MOV  #MSG50,R4    ;SET TO 7 TRK MSG
8541 021406 004737 020530      1$:  JSR  PC,TTOUT  ;PRINT TRK MSG
8542 021412 062700 000002      ADD  #2,RO        ;BUMP POINTER
8543 021416 000752                BR   ASEQ2        ;DO ALL
8544 021420 005037 000720      AMOD1: CLR  BLCNTR
8545
8546 021424 004737 004732 AMOD1B: JSR  PC,RWDA     ;GO REWIND ALL DRIVES
8547 021430 012737 000006 021622  MOV  #6,ABLCNT    ;SET NUMBER OF BLOCKS
8548 021436 012737 174000 000622  MOV  #-4000,CARCNT ;SET RECORD SIZE
8549 021444 012737 000100 000620  MOV  #100,RCNT    ;SET RECORD COUNT
8550 021452 012737 000003 000624  MOV  #3,PATRN     ;SELECT PATTERN 3
8551 021460 005037 000646      CLR  TMEX         ;ASSURE NO TM
8552 021464 004737 003264      JSR  PC,STAUTO    ;GO DO THIS PATTERN
8553 021470 012737 000007 000624  MOV  #7,PATRN     ;SELECT PATTERN 7
8554 021476 004737 003264      JSR  PC,STAUTO    ;GO DO THIS PATTERN
8555 021502 012737 000011 000624  MOV  #11,PATRN    ;SELECT PATTERN 11
8556 021510 004737 003264      JSR  PC,STAUTO    ;GO DO THIS PATTERN
8557 021514 012737 177777 021622  MOV  #-1,ABLCNT   ;FORCE TO END OF TAPE
8558 021522 012737 177777 000624  MOV  #-1,PATRN    ;SELECT AUTO RANDOM DATA
8559 021530 012737 152634 000672  MOV  #152634,RANBAS ;RESET RANDOM DATA BASE
8560 021536 012737 032561 000676  MOV  #32561,RANSAV ;GO DO RANDOM
8561 021544 004737 003264      JSR  PC,STAUTO
8562 021550 012704 025051      MOV  #MSG79,R4
8563 021554 004737 020530      JSR  PC,TTOUT     ;PRINT DIVIDER
8564 021560 012704 025143 ASEQX: MOV  #MSG81,R4
8565 021564 004737 020530      JSR  PC,TTOUT
8566 021570 005737 021626      TST  ASEQCF       ;SEE IF CONTINUOUS AUTO SEQ
8567 021574 001003                BNE  ASEQXX       ;IF SO: BR
8568 021576 000000                HALT
    
```



```

8569 021600 004737 022120
8570 021604 005237 000776
8571 021610 013703 000776
8572 021614 004737 020716
8573 021620 000635
8574 021622 000000
8575 021624 000000
8576 021626 000000
8577
8578
8579
8580 021630 005003
8581 021632 005037 000706
8582 021636 005037 000712
8583 021642 005037 004716
8584 021646 005037 000710
8585 021652 012777 010000 156722
8586 021660 113737 000710 000707
8587 021666 013777 000706 156706
8588 021674 004737 021226
8589 021700 032777 000001 156672
8590 021706 001421
8591 021710 052737 060000 000706
8592 021716 032777 000020 156654
8593 021724 001403
8594 021726 042737 020000 000706
8595 021734 013763 000706 001012
8596 021742 062703 000002
8597 021746 005237 000712
8598
8599 021752 005237 000710
8600 021756 022737 000010 000710
8601 021764 001335
8602 021766 005703
8603 021770 001007
8604 021772 012704 025171
8605 021776 004737 020530
8606 022002 000000
8607 022004 000137 003106
8608 022010 012763 177777 001012
8609 022016 013737 000712 004716
8610 022024 000337 000712
8611 022030 053737 000712 004716
8612 022036 000207
8613
8614
8615 022040 013746 000006
8616 022044 013746 000004
8617 022050 012737 022070 000004
8618 022056 022777 177777 156566
8619 022064 001402
8620 022066 000407
8621 022070 022626
8622 022072 012737 000176 000652
8623 022100 012737 000174 000654
8624 022106 012637 000004

      ASEQXX: JSR      PC,CKSWR      ;TEST FOR ↑G
              INC      SEQCT        ;BUMP PASS COUNT
              MOV      SEQCT,R3
              JSR      PC,OC↑P      ;PRINT PASS COUNT
              BR       ASEQO
      ABLCNT:  0
      ASEQF:   0
      ASEQCF:  0

      ;SUBROUTINE TO SELECT AUTO SEQ HARDWARE*****
      HRDS:   CLR      R3            ;CLEAR TABLE POINTER
              CLR      TEMP1        ;CLEAR UNIT DESCRIPTION HOLDER
              CLR      TEMP3        ;UNIT COUNT
              CLR      REOTC        ;CLEAR EOT COUNTER
              CLR      TEMP2        ;CLEAR UNIT INCREMENT
              MOV      #10000,AMTC  ;POWER CLEAR CONTROLLER
      HRDS1:  MOVB     TEMP2,TEMP1+1 ;POSITION UNIT NUMBER
              MOV      TEMP1,AMTC  ;SELECT DRIVE
              JSR      PC,STDLY     ;GO ASSURE VALID STATUS
              BIT      #1,AMTS      ;SEE IF AVAIL
              BEQ     HRDS2        ;IF NOT: BR
              BIS     #60000,TEMP1 ;SET DENSITY AND PARITY
              BIT     #20,AMTS     ;SEE IF 7 TRK
              BEQ     1$           ;IF NOT: BR
              BIC     #20000,TEMP1 ;ELSE SET TO 7 TRK NORMAL DENSITY
      1$:    MOV      TEMP1,UNI(R3) ;PUT IN TABLE
              ADD     #2,R3
              INC     TEMP3        ;INCREMENT COUNT
      HRDS2:  INC      TEMP2        ;SET FOR NEXT UNIT
              CMP     #10,TEMP2    ;DONE?
              BNE     HRDS1        ;IF NOT: BR
              TST     R3           ;FOUND A UNIT?
              BNE     HRDSX        ;IF SO: BR
              MOV     #MSG82,R4
              JSR     PC,TTOUT     ;TYPE NO UNIT AVAILABLE
              HALT
      HRDSX:  JMP      STAUT        ;START AUTO SEQ AGAIN
              MOV     #-1,UNI(R3) ;MARK END OF TABLE
              MOV     TEMP3,REOTC  ;SET NUMBER OF UNITS
              SWAB   TEMP3
              BIS     TEMP3,REOTC  ;SET EOT CNTR
              RTS     PC           ;RETURN

      SUSWR:  MOV     @#6,-(SP)    ;SAVE VECTORS
              MOV     @#4,-(SP)
              MOV     #1$,@#4      ;SET UP FOR TIMEOUT
              CMP     #-1,@SWR     ;REFERENCE HARDWARE SWITCH REGISTER
              BEQ     2$
              BR     3$
      1$:    CMP     (SP)+,(SP)+    ;ADJUST STACK
      2$:    MOV     #SWREG,SWR    ;POINT TO SOFTWARE SWITCH REG
      3$:    MOV     #DISPREG,DISPLAY ;POINT TO SOFT DISPLAY REG
              MOV     (SP)+,@#4    ;RESTORE VECTORS

```

8625	022112	012637	000006		MOV	(SP)+, @#6	
8626	022116	000207			RTS	PC	:RETURN
8627							
8628	022120	022737	000176	000652	CKSWR:	CMP #SWREG, SWR	:SOFTWARE SWITCH REG PRESENT
8629	022126	001036			BNE	OUT	:NO GET OUT
8630	022130	017737	156524	000704	MOV	@TKB, TIB	:AND STRIP OFF
8631	022136	042737	177600	000704	BIC	#177600, TIB	:THE GARBAGE
8632	022144	022737	000007	000704	CMP	#7, TIB	:IS IT A <↑G>
8633	022152	001024			BNE	OUT	
8634	022154	012704	026152		MOV	#\$CNTG, R4	
8635	022160	004737	020530		JSR	PC, TTOUT	
8636	022164	012704	026156		CNTLU:	MOV #SMSWR, R4	
8637	022170	004737	020530		JSR	PC, TTOUT	
8638	022174	017703	156452		MOV	@SWR, R3	
8639	022200	004737	020706		JSR	PC, OCTPE	
8640	022204	012704	026166		MOV	#\$MNEW, R4	
8641	022210	004737	020530		JSR	PC, TTOUT	
8642	022214	005037	001002		CLR	@TEMPST	
8643	022220	004737	022226		JSR	PC, \$READ	:GO READ A LINE
8644	022224	000207			OUT:	RTS	:RETURN TO MAIN BODY OF PROGRAM
8645							
8646	022226	005037	001002		\$READ:	CLR TEMPST	
8647	022232	012737	000007	001000	MOV	#7, COUNT	
8648	022240	004737	020456		1\$:	JSR PC, TTIN	:GO READ A CHARACTER
8649	022244	042737	177600	000704	BIC	#177600, TIB	:STRIP OFF GARBAGE
8650	022252	122737	000025	000704	CMPB	#25, TIB	:IS IT A ↑U?
8651	022260	001002			BNE	2\$:BRANCH IF NOT
8652	022262	005726			3\$:	TST (SP)+	:POP THE STACK
8653	022264	000737			BR	CNTLU	:START OVER
8654	022266	122737	000015	000704	2\$:	CMPB #15, TIB	:IS IT A <CR>?
8655	022274	001013			BNE	4\$:BRANCH IF NOT
8656	022276	012737	000200	001004	MOV	#200, RDSW	
8657	022304	004737	020572		JSR	PC, TCRLF	:ECHO IT WITH <LF>
8658	022310	022737	000007	001000	CMP	#7, COUNT	:WAS IT FIRST CHARACTER
8659	022316	001037			BNE	7\$:CHANGE SWR IF NOT FIRST ONE
8660	022320	005726			8\$:	TST (SP)+	:POP THE STACK
8661	022322	000740			BR	OUT	:GET OUT
8662	022324	122737	000060	000704	4\$:	CMPB #60, TIB	
8663	022332	003004			BGT	5\$	
8664	022334	122737	000067	000704	CMPB	#67, TIB	
8665	022342	002005			BGE	6\$	
8666	022344	012704	023574		5\$:	MOV #MSG43, R4	
8667	022350	004737	020530		JSR	PC, TTOUT	
8668	022354	000742			BR	3\$:START OVER IF NOT LEGAL CHARACTER
8669	022356	006337	001002		6\$:	ASL TEMPST	
8670	022362	006337	001002		ASL	TEMPST	
8671	022366	006337	001002		ASL	TEMPST	
8672	022372	142737	000060	000704	BICB	#60, TIB	:GET NITTY-GRITTY
8673	022400	153737	000704	001002	BISB	TIB, TEMPST	
8674	022406	005337	001000		DEC	COUNT	:ONLY WANT 6 DIGITS
8675	022412	001754			BEQ	5\$	
8676	022414	000711			BR	1\$	
8677	022416	013777	001002	156226	7\$:	MOV TEMPST, @SWR	:CHANGE SWITCH REGISTER CONTENTS
8678	022424	000735			BR	8\$	
8679							
8680							

```

8682
8683                                     :ERROR MESSAGES*****
8684
8685 022426 042052 020105 043 MSG1: .ASCII /*DE */
8686
8687 022433 045 035507 021440 MSG2: .ASCII /*G; */
8688
8689 022440 041045 020073 043 MSG3: .ASCII /*B; */
8690
8691 022445 045 047103 021440 MSG4: .ASCII /*CN */
8692
8693 022452 053452 020105 043 MSG5: .ASCII /*WE */
8694
8695 022457 052 042522 021440 MSG6: .ASCII /*RE */
8696
8697 022464 051052 020123 043 MSG7: .ASCII /*RS */
8698
8699 022471 052 042523 021440 MSG10: .ASCII /*SE */
8700
8701 022476 022445 052445 044516 MSG11: .ASCII /*%%UNIT NO. */
      022504 020124 047516 020056
      022512 043
8702
8703 022513 045 041052 020116 MSG13: .ASCII /**BN */
      022520 043
8704
8705 022521 052 047122 021440 MSG14: .ASCII /*RN */
8706
8707 022526 020045 020040 020040 MSG15: .ASCII /*% BAD RECORD%%*/
      022534 020040 020040 041040
      022542 042101 051040 041505
      022550 051117 022504 021445
8708
8709 022556 043040 025052 021452 MSG16: .ASCII / F****/
8710
8711 022564 051040 025052 021452 MSG17: .ASCII / R****/
8712
8713 022572 042445 052117 020040 MSG20: .ASCII /*EOT NO. */
      022600 047040 027117 021440
8714 022606 052445 044516 020124 MSG20A: .ASCII /*UNIT WILL REWIND AND BE*/
      022614 044527 046114 051040
      022622 053505 047111 020104
      022630 047101 020104 042502
      022636 045
8715 022637 122 051505 040524 .ASCII /*RESTARTED ON BLOCK ONE*/
      022644 052122 042105 047440
      022652 020116 046102 041517
      022660 020113 047117 022505
8716 022666 044127 047105 040440 .ASCII /*WHEN ALL AVAIL UNITS REACH EOT*/
      022674 046114 040440 040526
      022702 046111 052440 044516
      022710 051524 051040 040505
      022716 044103 042440 052117
      022724 043
8717
8718
    
```

E08

TM A,B-11 TS03 OR TU10,N,W MULTIDRIVE DATA RELIABILITY EXERCISER
DZTMHE.P11 20-DEC-76 12:27 ACT11 HOOKS

MACY11 27(1006) 20-DEC-76 12:35 PAGE 102-1

SEQ 0095

8719	022725	045	020441	044441	MSG22: .ASCII	/%!!!ILLEGAL BOT%%#/ 022732 046114 043505 046101 022740 041040 052117 022445 022746 021445
8720						
8721	022750	041445	046517	020104	MSG23: .ASCII	/%COMD #/ 022756 043
8722						
8723	022757	045	047516	044440	MSG24: .ASCII	/%NO INTERRUPT RETURNED%%#/ 022764 052116 051105 052522 022772 052120 051040 052105 023000 051125 042516 022504 023000 043
8724						
8725	023007	045	020441	047041	MSG25: .ASCII	/%!!!NO CONTROLLER READY !!! STOP:%%/ 023014 020117 047503 052116 023022 047522 046114 051105 023030 051040 040505 054504 023036 020440 020441 051440 023044 047524 035120 045
8726	023051	120	042522	051523	.ASCII	/%PRESS CONTINUE TO RESUME TESTING%%#/ 023056 041440 047117 044524 023064 052516 020105 047524 023072 051040 051505 046525 023100 020105 042524 052123 023106 047111 022507 043
8727						
8728	023113	045	051104	050117	MSG26: .ASCII	/%DROPS: #/ 023120 035123 021440
8729						
8730	023124	050045	041511	051513	MSG27: .ASCII	/%PICKS: #/ 023132 020072 043
8731						
8732	023135	045	052123	052101	MSG30: .ASCII	/%STAT #/ 023142 021440
8733						
8734	023144	022445	046524	040454	MSG31: .ASCII	/%TM,A,B-11:TS03 OR TU10,N,W MULTIDRIVE DATA RELIABILTY EXERCISER (DZTM 023152 041054 030455 035061 023160 051524 031460 047440 023166 020122 052524 030061 023174 047054 053454 046440 023202 046125 044524 051104 023210 053111 020105 040504 023216 040524 051040 046105 023224 040511 044502 052114 023232 020131 054105 051105 023240 044503 042523 020122 023246 042050 052132 044115 023254 042455 022451 043
8735	023261	105	052116	051105	MSG31A: .ASCII	/%ENTER CONDITIONS IN OCTAL%%#/ 023266 041440 047117 044504 023274 044524 047117 020123 023302 047111 047440 052103 023310 046101 021445
8736						
8737	023314	052445	044516	020124	MSG32: .ASCII	/%UNIT NUMBER = #/

	023322	052516	041115	051105	
	023330	036440	021440		
8738					
8739	023334	042045	047105	044523	MSG33: .ASCII /%DENSITY = #/
	023342	054524	036440	021440	
8740					
8741	023350	050045	051101	052111	MSG34: .ASCII /%PARITY = #/
	023356	020131	020075	043	
8742					
8743	023363	045	042522	047503	MSG35: .ASCII /%RECORD COUNT = #/
	023370	042122	041440	052517	
	023376	052116	036440	021440	
8744					
8745	023404	041445	040510	040522	MSG36: .ASCII /%CHARACTER COUNT = #/
	023412	052103	051105	041440	
	023420	052517	052116	036440	
	023426	021440			
8746					
8747	023430	050045	052101	042524	MSG37: .ASCII /%PATTERN NUMBER = #/
	023436	047122	047040	046525	
	023444	042502	020122	020075	
	023452	043			
8748					
8749	023453	045	044523	043516	MSG38: .ASCII /%SINGLE PASS = #/
	023460	042514	050040	051501	
	023466	020123	020075	043	
8750	023473	041	020441	042445	MSG39: .ASCII /!!!%END OF PASS#/
	023500	042116	047440	020106	
	023506	040520	051523	043	
8751	023513	045	042445	052116	MSG40: .ASCII /%%ENTER STALLS%READ = #/
	023520	051105	051440	040524	
	023526	046114	022523	042522	
	023534	042101	036440	021440	
8752					
8753	023542	053445	044522	042524	MSG41: .ASCII /%WRITE = #/
	023550	036440	021440		
8754					
8755	023554	052045	051125	020116	MSG42: .ASCII /%TURN AROUND = #/
	023562	051101	052517	042116	
	023570	036440	021440		
8756					
8757	023574	037445	021445		MSG43: .ASCII /%?%#/
8758					
8759	023600	042445	052116	051105	MSG44: .ASCII /%ENTER YOZZLE STALL = #/
	023606	054440	055117	046132	
	023614	020105	052123	046101	
	023622	020114	020075	043	
8760					
8761	023627	045	051105	020122	MSG45: .ASCII /%ERR AMT #/
	023634	046501	020124	043	
8762					
8763	023641	045	041527	021440	MSG46: .ASCII /%WC #/
8764					
8765	023646	041445	020101	043	MSG47: .ASCII /%CA #/
8766					
8767	023653	045	020441	047041	MSG48: .ASCII /%!!!NO BOT ON REWIND: #/

	023660	020117	047502	020124		
	023666	047117	051040	053505		
	023674	047111	035104	043		
8768						
3769	023701	040	047516	020124	MSG49: .ASCII	/ NOT AVAIL #/
	023706	053101	044501	020114		
	023714	043				
8770	023715	055	052067	045522	MSG50: .ASCII	/-7TRK #/
	023722	021440				
8771	023724	034455	051124	020113	MSG51: .ASCII	/-9TRK #/
	023732	043				
8772	023733	045	047516	035116	MSG52: .ASCII	/%NON:RETRYABLE #/
	023740	042522	051124	040531		
	023746	046102	020105	043		
8773	023753	045	025052	047452	MSG53: .ASCII	/%***ORIGINAL ERROR***#/
	023760	044522	044507	040516		
	023766	020114	051105	047522		
	023774	025122	025052	043		
8774	024001	045	042522	047503	MSG54: .ASCII	/%RECOVERED#/
	024006	042526	042522	021504		
8775	024014	051045	052105	054522	MSG55: .ASCII	/%RETRY: #/
	024022	020072	043			
8776	024025	045	052523	050123	MSG56: .ASCII	/%SUSPECT BAD TAPE#/
	024032	041505	020124	040502		
	024040	020104	040524	042520		
	024046	043				
8777	024047	045	042522	042520	MSG57: .ASCII	/%REPEAT: #/
	024054	052101	020072	043		
8778	024061	045	020441	052441	MSG58: .ASCII	/%!!!UNRECOVERABLE BAD SPOT#/
	024066	051116	041505	053117		
	024074	051105	041101	042514		
	024102	041040	042101	051440		
	024110	047520	021524			
8779						
8780	024114	020445	020441	040502	MSG59: .ASCII	/%!!!BAD TAPE OVERFLOW/
	024122	020104	040524	042520		
	024130	047440	042526	043122		
	024136	047514	127			
8781	024141	045	040524	042520	.ASCII	/%TAPE WILL BE REWOUND AND REMOVED FROM/
	024146	053440	046111	020114		
	024154	042502	051040	053505		
	024162	052517	042116	040440		
	024170	042116	051040	046505		
	024176	053117	042105	043040		
	024204	047522	115			
8782	024207	045	042524	052123	.ASCII	/%TESTING UNTIL ALL ARE RESTARTED AT BLOCK ONE. #/
	024214	047111	020107	047125		
	024222	044524	020114	046101		
	024230	020114	051101	020105		
	024236	042522	052123	051101		
	024244	042524	020104	052101		
	024252	041040	047514	045503		
	024260	047440	042516	021456		
8783	024266	052045	050101	020105	MSG60: .ASCII	/%TAPE MARK = #/
	024274	040515	045522	036440		
	024302	021440				

H08

8784							
8785	024304	020445	020441	040502	MSG61:	.ASCII	/%!!!BACKSPACE ERROR/
	024312	045503	050123	041501			
	024320	020105	051105	047522			
	024326	122					
8786	024327	045	040524	042520		.ASCII	/%TAPE WILL BE REWOUND AND REMOVED FROM /
	024334	053440	046111	020114			
	024342	042502	051040	053505			
	024350	052517	042116	040440			
	024356	042116	051040	046505			
	024364	053117	042105	043040			
	024372	047522	020115				
8787	024376	052045	051505	044524		.ASCII	/%TESTING UNTIL ALL ARE RESTARTED AT BLOCK ONE.*/
	024404	043516	052440	052116			
	024412	046111	040440	046114			
	024420	040440	042522	051040			
	024426	051505	040524	052122			
	024434	042105	040440	020124			
	024442	046102	041517	020113			
	024450	047117	027105	043			
8788	024455	052	042527	052040	MSG62:	.ASCII	/*WE TM*/
	024462	021515					
8789	024464	051452	020105	046524	MSG63:	.ASCII	/*SE TM*/
	024472	043					
8790	024473	045	052127	051105	MSG64:	.ASCII	/%WTERR: */
	024500	035122	021440				
8791	024504	051045	042504	051122	MSG65:	.ASCII	/%RDERR: */
	024512	020072	043				
8792	024515	045	052104	051105	MSG66:	.ASCII	/%DTERR: */
	024522	035122	021440				
8793	024526	021445			MSG67:	.ASCII	/%*/
8794	024530	041040	042101	052040	MSG68:	.ASCII	/ BAD TAPE SPOTS*/
	024536	050101	020105	050123			
	024544	052117	022523	043			
8795	024551	052	042523	051040	MSG69:	.ASCII	/*SE RTY*/
	024556	054524	043				
8796	024561	052	042522	052040	MSG70:	.ASCII	/*RE TM*/
	024566	021515					
8797	024570	051045	040505	020104	MSG71:	.ASCII	/%READ FAILED--RETRY: */
	024576	040506	046111	042105			
	024604	026455	042522	051124			
	024612	035131	021440				
8798	024616	020445	020441	040510	MSG72:	.ASCII	/%!!!HARD READ ERROR*/
	024624	042122	051040	040505			
	024632	020104	051105	047522			
	024640	021522					
8799	024642	051045	051105	040505	MSG73:	.ASCII	/%REREAD SUCCESSFUL--RETRY: */
	024650	020104	052523	041503			
	024656	051505	043123	046125			
	024664	026455	042522	051124			
	024672	035131	021440				
8800	024676	020045	047523	052106	MSG74:	.ASCII	/% SOFT: */
	024704	020072	043				
8801	024707	045	044040	051101	MSG75:	.ASCII	/% HARD: */
	024714	035104	021440				
9802	024720	020045	052122	054522	MSG76:	.ASCII	/% RTRY: */

8803	024726	020072	043			
	024731	045	052045	026115	MSG77: .ASCII	/%TM,A,B-11 AUTO SEQUENCE TEST (DZTMH-E)%/
	024736	026101	026502	030461		
	024744	040440	052125	020117		
	024752	042523	052521	047105		
	024760	042503	052040	051505		
	024766	020124	042050	052132		
	024774	044115	042455	022451		
8804	025002	047105	042524	020122	.ASCII	/ENTER RESPONSES IN OCTAL%#/
	025010	042522	050123	047117		
	025016	042523	020123	047111		
	025024	047440	052103	046101		
	025032	021445				
8805	025034	040445	052125	020117	MSG78: .ASCII	/%AUTO CONT: #/
	025042	047503	052116	020072		
	025050	043				
8806	025051	045	025045	025052	MSG79: .ASCII	/%*****%/
	025056	025052	025052	025052		
	025064	025052	025052	025052		
	025072	025052				
8807	025074	025052	025052	025052	.ASCII	/%*****%/
	025102	025052	025052	025052		
	025110	025052	025052	022452		
	025116	043				
8808	025117	125	044516	051524	MSG80: .ASCII	/UNITS TO BE TESTED%#/
	025124	052040	020117	042502		
	025132	052040	051505	042524		
	025140	022504	043			
8809	025143	105	042116	047440	MSG81: .ASCII	/END OF SEQUENCE NO. #/
	025150	020106	042523	052521		
	025156	047105	042503	020040		
	025164	047516	020056	043		
8810	025171	045	020441	047041	MSG82: .ASCII	/%!!!NO DRIVES AVAILABLE FOR AUTO SEQ--HALT%#/
	025176	020117	051104	053111		
	025204	051505	040440	040526		
	025212	046111	041101	042514		
	025220	043040	051117	040440		
	025226	052125	020117	042523		
	025234	026521	044055	046101		
	025242	022524	043			
8811	025245	045	050114	020103	MSG83: .ASCII	/%LPC #/
	025252	043				
8812	025253	045	042522	044507	MSG84: .ASCII	/%REGISTER START = #/
	025260	052123	051105	051440		
	025266	040524	052122	036440		
	025274	021440				
8813	025276	053045	041505	047524	MSG85: .ASCII	/%VECTOR ADDRESS = #/
	025304	020122	042101	051104		
	025312	051505	020123	020075		
	025320	043				
8814	025321	052	040520	052124	MSG86: .ASCII	/%PATTRN #/
	025326	047122	021440			
8815	025332	050045	042522	040515	MSG87: .ASCII	/%PREMATURE EOT IN AUTO SEQ/
	025340	052524	042522	042440		
	025346	052117	044440	020116		
	025354	052501	047524	051440		

JOB

M.A.B-11 TS03 OR TU10,N,W MULTIDRIVE DATA RELIABILITY EXERCISER
 02TMHE.P11 20-DEC-76 12:27 ACT11 HOOKS

MACY11 27(1006) 20-DEC-76 12:35 PAGE 102-6

SEQ 0100

8816	025362	050505				
	025364	052045	050101	020105	.ASCII	/%TAPE WILL BE REWOUND AND AUTO SEQUENCE/
	025372	044527	046114	041040		
	025400	020105	042522	047527		
	025406	047125	020104	047101		
	025414	020104	052501	047524		
	025422	051440	050505	042525		
	025430	041516	105			
8817	025433	045	044527	046114	.ASCII	/%WILL CONTINUE ON THIS UNIT#/
	025440	041440	047117	047111		
	025446	042525	047440	020116		
	025454	044124	051511	052440		
	025462	044516	021524			
8818	025466	051040	052105	054522	MSG88: .ASCII	/ RETRY#/
	025474	043				
9819						
8820	025475	045	020441	052441	MSG89: .ASCII	/%!!!UNIT IS REWINDING; TEST WILL START WHEN DONE#/
	025502	044516	020124	051511		
	025510	051040	053505	047111		
	025516	044504	043516	020073		
	025524	042524	052123	053440		
	025532	046111	020114	052123		
	025540	051101	020124	044127		
	025546	047105	042040	047117		
	025554	021505				
8821	025556	042052	047105	021440	MSG90: .ASCII	/*DEN #/
8822	025564	050052	051101	021440	MSG91: .ASCII	/*PAR #/
8823	025572	020441	022441	042045	MSG92: .ASCII	/%!!!%DROPPED UNIT: #/
	025600	047522	050120	042105		
	025606	052440	044516	035124		
	025614	021440				
8824	025616	040445	052124	046505	MSG93: .ASCII	/%ATTEMPT TO RESTART UNIT WILL BE/
	025624	052120	052040	020117		
	025632	042522	052123	051101		
	025640	020124	047125	052111		
	025646	053440	046111	020114		
	025654	042502				
8825	025656	046445	042101	020105	.ASCII	/%MADE AT END OF PASS!!!%#/
	025664	052101	042440	042116		
	025672	047440	020106	040520		
	025700	051523	020441	022441		
	025706	021445				
8826	025710	020441	022441	047045	MSG94: .ASCII	/%!!!%NO MORE UNITS TO TEST IN THIS PASS/
	025716	020117	047515	042522		
	025724	052440	044516	051524		
	025732	052040	020117	042524		
	025740	052123	044440	020116		
	025746	044124	051511	050040		
	025754	051501	123			
8827	025757	045	046101	020114	.ASCII	/%ALL ARE DROPPED OR REWUOND.%#/
	025764	051101	020105	051104		
	025772	050117	042520	020104		
	026000	051117	051040	053505		
	026006	047525	042116	022456		
	026014	021445				
8828	026016	020441	022441	047514	MSG95: .ASCII	/%!!!%LOST SELECT REMOTE#/

K08

8829	026024	052123	051440	046105	
	026032	041505	020124	042522	
	026040	047515	042524	043	
	026045	041	020441	040445	MSG96: .ASCII /!!!%ALL ARE DROPPED: END OF PASS STOP!!!#/ MSG97: .ASCII /!:%CANNOT TEST LOAD MEDIUM%#/ \$CNTG: .ASCII /%↑G#/ \$MSWR: .ASCII /%SWR= #/ \$MNEW: .ASCII / NEW= #/ WDATA: 0 ;WRITE BUFFER RDATA: 0 ;READ BUFFER .END
	026052	046114	040440	042522	
	026060	042040	047522	050120	
	026066	042105	020072	047105	
	026074	020104	043117	050040	
	026102	051501	020123	052123	
8830	026110	050117	020441	021441	
	026116	020441	041445	047101	
	026124	047516	020124	042524	
	026132	052123	046040	040517	
	026140	020104	042515	044504	
8831	026146	046525	021445		
8832	026152	057045	021507		
	026156	022445	053523	036522	
8833	026164	021440			
	026166	020040	042516	036527	
	026174	021440			
8834					
8835	026176	000000			
8836					
8837		032210			
8838	032210	000000			
8839					
8840		000001			

ABL CNT	021622	BTOVX	016014	DATA4	002746	DERR1	014370	DRP8	001072
ACTLRC	013702	BTOVO	015670	DATAS	002750	DERR2	014372	DSJP	012370
AMCD1	021420	BTOV1	015700	DATA6	002752	DERR3	014376	DSO	012376
AMOD1B	021424	BTOV2	016002	DATA7	002754	DERR4	014400	DSOA	012420
ASEQ	021252	BTPT	015606	DATBL	002734	DERR4A	014544	DSOB	012456
ASEQCF	021626	BTPT	000762	DATER1	001134	DERR4B	014556	DSOC	012460
ASEQF	021624	BTSTF	000772	DATER2	001136	DERR5	014576	DS1	012510
ASEQX	021560	BTUR	006614	DATER3	001140	DERR6	014604	DS3	012516
ASEQXX	021604	BT00	001654	DATER4	001142	DFA	014004	DS4	012526
ASEQ0	021314	BT01	001760	DATER5	001144	DFB	014012	DUCTR	001006
ASEQ2	021344	BT02	002064	DATER6	001146	DFC	014020	DOFL	012746
BBC	000722	BT03	002170	DATER7	001150	DFD	014034	EMADDR	000716
BCNT	000736	BT04	002274	DATER8	001152	DFX	014234	EOTCT1	001234
BOPP	000766	BT05	002400	DATR	013236	DF0	014044	EOTCT2	001236
BORTY1	001214	BT06	002504	DATRO	013254	DF3	014204	EOTCT3	001240
BORTY2	001216	BT07	002610	DATO	012552	DF7	014120	EOTCT4	001242
BORTY3	001220	BO	010136	DATO A	012502	DF71	014152	EOTCT5	001244
BORTY4	001222	BOA	010444	DATOB	012620	DF72	014170	EOTCT6	001246
BORTY5	001224	BOB	010440	DATOC	012704	DF9	014062	EOTCT7	001250
BORTY6	001226	B1	010536	DATOD	012712	DF91	014074	EOTCT8	001252
BORTY7	001230	B1A	010612	DATOE	012722	DF92	014112	EOTREC	000760
BORTY8	001232	B2	010620	DATOF	012736	DISPLA	000654	ERCHK	016036
B000	001454	CADER	016740	DAT1	012754	DISPRE	000174	EREX	016720
BD10	001474	CARCNT	000622	DAT1A	012760	DOUT	021144	EREXO	016664
BD20	001514	CCNTR	010644	DAT1B	012764	DOUT1	021160	ERSAV	000740
BD30	001534	CCSAV	000644	DAT10	013074	DOUT2	021206	ERTFL	000764
BD40	001554	CHAIN	000210	DAT11	013104	DOUT3	021214	ERO	016130
BD50	001574	CKSWR	022120	DAT11A	013112	DPC	014746	ERDA	016110
BD60	001614	CLLAST	013412	DAT12	013126	DPCG	014754	ER1	016134
BD70	001634	CLP	013522	DAT12A	013136	DPC0	014762	ER1A	016342
BKSP	010132	CLPE	013546	DAT13	013152	DPCOA	015024	ER1B	016302
BK0	010172	CLPO	013556	DAT14	013162	DPC1	015032	ER10	016530
BK1	010246	CLP1	013570	DAT15	013172	DPC1A	015060	ER2	016360
BK1A	010336	CLP2	013620	DAT15A	013202	DPC2	015122	ER2A	016412
BK1B	010402	CLP3	013642	DAT15B	013206	DPC2A	015064	ER3	016416
BK1C	010264	CLO	013326	DAT15C	013220	DPC2B	015106	ER3A	016426
BK2	010322	CL1	013354	DAT15R	013176	DPC3	015154	ER3B	016432
BK3	010412	CL2	013376	DAT2	012776	DPPRT	015222	ER7	016462
BLCNTR	000720	CL3	013464	DAT3	013004	DPPRTX	015366	EXCRC	013676
BPKP	000770	CNTLU	022164	DAT3A	013012	DPPRT0	015274	EXLRC	013700
BP00	001254	COUNT	001000	DAT3B	013016	DPPRT1	015322	FREX	017004
BP10	001274	CRCLRC	013306	DAT4	013032	DPPRT2	015340	FRPRT	016742
BP20	001314	CRLR	013310	DAT5	013044	DROP	014736	FRO	016774
BP30	001334	DATAO	002736	DAT6	013054	DRPDV	020050	GORTY1	001174
BP40	001354	DAT A1	002740	DAT7	013064	DRPK	014724	GORTY2	001176
BP50	001374	DAT A10	002756	DCHK	013712	DRPKF	014656	GORTY3	001200
BP60	001414	DAT A11	002760	DEREX	014610	DRP1	001054	GORTY4	001202
BP70	001434	DAT A12	002762	DEREX1	014644	DRP2	001056	GORTY5	001204
BTADDR	002714	DAT A13	002764	DERFL	000744	DRP3	001060	GORTY6	001206
BTCLR	016016	DAT A14	002766	DERR	014242	DRP4	001062	GORTY7	001210
BTCLR1	016026	DAT A15	002770	DERR0	014256	DRP5	001064	GORTY8	001212
BTFLG	000746	DAT A2	002742	DERR0A	014316	DRP6	001066	HORFL	000726
BTOV	006570	DAT A3	002744	DERR0B	014336	DRP7	001070	HERE	004660

HRDS	021630	MSG55	024014	OCTPE	020706	RDER8	001132	RTYFL	000754
HRDSX	022010	MSG56	024025	OCTPE1	020722	RDEX	007350	RTY1	001154
HRDS1	021660	MSG57	024047	OCTPG	021100	RDEXX	007374	RTY2	001156
HRDS2	021752	MSG58	024061	OCTPG0	021116	RDFL	013304	RTY3	001160
LRCPT	013706	MSG59	024114	OCTPG1	021122	RDSW	001004	RTY4	001162
LRCV	013704	MSG6	022457	OCTPO	020744	RDO	006746	RTY5	001164
MDA	000606	MSG60	024266	OCTP1	020766	RD1	006756	RTY6	001166
MSG1	022426	MSG61	024304	OCTP2	020774	RD1A	006764	RTY7	001170
MSG10	022471	MSG62	024455	OCTP3	021064	RD10	007276	RTY8	001172
MSG11	022476	MSG63	024464	OFL	021142	RD2	007012	RWNO	004720
MSG13	022513	MSG64	024473	OUT	022224	RD3	007034	RWNOA	004732
MSG14	022521	MSG65	024504	PAPRT	017506	RD4	007066	RWNDX	005206
MSG15	022526	MSG66	024515	PAPRTA	017650	RD4A	007170	RWNDC	004756
MSG16	022556	MSG67	024526	PAPRTB	017664	RD5	007174	RWNO1	005016
MSG17	022564	MSG68	024530	PAPRTC	017700	RD6	007222	RWNO2	005040
MSG2	022433	MSG69	024551	PAPRTD	017710	RD7	007242	RWNO3	005050
MSG20	022572	MSG7	022464	PAPRT1	017752	READ	006722	RWNO4	005120
MSG20A	022606	MSG70	024561	PAPRT2	017762	REGST	000674	RWNO5	005150
MSG22	022725	MSG71	024570	PARCNT	013710	REOT	004142	RWNO6	005162
MSG23	022750	MSG72	024616	PARS	012752	REOTC	004716	SEQCT	000776
MSG24	022757	MSG73	024642	PATRN	000624	REOTX	004664	SERFL	000742
MSG25	023007	MSG74	024676	PATS	012750	REOT1	004150	SPBK	010466
MSG26	023113	MSG75	024707	PFLG	000732	REOT1A	004300	SPFLG	000630
MSG27	023124	MSG76	024720	PICK	015156	REOT1B	004334	STAL	000730
MSG3	022440	MSG77	024731	PIK1	001034	REOT1C	004236	STALL	010634
MSG30	023135	MSG78	025034	PIK2	001036	REOT1D	004256	START	002772
MSG31	023144	MSG79	025051	PIK3	001040	REOT10	004662	STARTA	003124
MSG31A	023261	MSG80	025117	PIK4	001042	REOT2	004344	STARTB	003146
MSG32	023314	MSG81	025143	PIK5	001044	REOT3	004402	STARTC	003174
MSG33	023334	MSG82	025171	PIK6	001046	REOT4	004432	STARTD	003236
MSG34	023350	MSG83	025245	PIK7	001050	REOT5	004456	STARTE	003142
MSG35	023363	MSG84	025253	PIK8	001052	REOT6	004522	STARTO	003156
MSG36	023404	MSG85	025276	PRB	000670	REOT6A	004540	START1	003356
MSG37	023430	MSG86	025321	PRS	000666	REOT7	004570	START2	003512
MSG38	023453	MSG87	025332	PRSTAT	015370	REOT8	004622	START3	003526
MSG39	023473	MSG88	025466	PRSTA2	015400	REOT9	004640	START4	003542
MSG4	022445	MSG89	025475	PSW	000650	RPCNT	000750	START5	004016
MSG40	023513	MSG90	025556	RANBAS	000672	PRTY	007376	START6	004034
MSG41	023542	MSG91	025564	RANG	020240	RRTYFL	000774	START7	004072
MSG42	023554	MSG92	025572	RANSV	000676	RRTYX	007566	START8	004104
MSG43	023574	MSG93	025616	RANSET	004110	RRTY0	007422	STAROA	003310
MSG44	023600	MSG94	025710	RCNT	000620	RRTY1	007426	STAROB	003340
MSG45	023627	MSG95	026016	RCNTR	010676	RRTY2	007474	STARIA	003402
MSG46	023641	MSG96	026045	RCSV	000642	RRTY3	007514	STARIB	003414
MSG47	023646	MSG97	026116	RDATA	032210	RRTY4	007526	STARIC	003456
MSG48	023653	MTC	000602	RDCMD	000626	RRTY5	007556	STAR40	003562
MSG49	023701	MTD	000610	RDER1	001114	RSEQ	006626	STAR44	003630
MSG5	022452	MTINT	017466	RDER2	001116	RSEX	006720	STAR45	003644
MSG50	023715	MTRD	000612	RDER3	001120	RSFRD	006676	STAR46	003664
MSG51	023724	MTS	000600	RDER4	001122	RSFROA	006656	STAS	014240
MSG52	023733	MWC	000604	RDER5	001124	RSTAL	000632	STAUT	003106
MSG53	023753	NOP	= 000240	RDER6	001126	RTCNT	000752	STAUTO	003264
MSG54	024001	OCTP	020716	RDER7	001130	TRN	000724	STCOFL	001010

STCS	014236	TINPB	010752	TTIN2	020512	WRTS80	006566	W1A	005370
STDLY	021226	TINPB1	011000	TTOUT	020530	WRTY	006142	W3	005440
SUSWR	022040	TINPO	011020	TTR	020272	WRTYR	006174	W3A	005462
SWR	000652	TINPOA	011310	TTR0	020300	WRTYTM	006170	W3AO	005460
SWREG	000176	TINPOB	011326	TTR1	020326	WRTYO	006150	W3B	005552
TAPG	017006	TINPOC	011364	TTR1A	020342	WRTY1	006256	W3C	005602
TAPG0	017024	TINPOD	011416	TTR1B	020356	WRTY2	006260	W3D	005616
TAPG1	017070	TINPOE	011436	TTR2	020414	WRTY3	006340	W4	005670
TAPG2	017104	TINPOF	011442	TTR3	020424	WRTY3A	006356	XOR	013650
TAPG2A	017114	TINPOG	011230	TTR4	020434	WRTY4	006444	XORS	013674
TAPG3	017126	TINPI	011446	TTR5	020436	WRTY5	006500	YOZ	007600
TAPG4	017230	TINP2	011526	UDES	000616	WSTAL	000634	YOZA	007626
TAPG5	017240	TINP2A	011606	UNP	000734	WTER1	001074	YOZB	007634
TAPG6	017314	TINP2B	011634	UNX	001032	WTER2	001076	YOZC	007654
TAPG7	017326	TINP3	011654	UN1	001012	WTER3	001100	YOZC1	007736
TAPG8	017334	TINP4	012164	UN2	001014	WTER4	001102	YOZD	007750
TBELL	020654	TKB	000660	UN3	001016	WTER5	001104	YOZE	007764
TCRLF	020572	TKS	000656	UN4	001020	WTER6	001106	YOZE1	007774
TCRLFA	020610	TMEX	000646	UN5	001022	WTER7	001110	YOZF	010010
TEMPST	001002	TMFLG	000756	UN6	001024	WTER8	001112	YCZG	010112
TEMP1	000706	TOB	000702	UN7	001026	WTM	005712	YOZO	007620
TEMP2	000710	TOG	020670	UN8	001030	WTMO	005760	YSTAL	000640
TEMP3	000712	TPB	000664	VECT	000614	WTM1	006012	\$CNTG	026152
TEMP4	000714	TPOS	012342	WDATA	026176	WTM2	006070	\$ENDAD	004650
TEX	020704	TPS	000662	WEOTF	017402	WTM3	006130	\$MNEW	026166
TIB	000704	TSTAL	000636	WEX	005636	WTM4	006134	\$MSWR	026156
TINER	020440	TTIN	020456	WEX1	005650	WO	005312	\$READ	022226
TINF	000700	TTINT	017404	WEX2	005666	WOB	005350	\$SVPC	= 000042
TINP	010722	TTINTO	017436	WRITE	005254	W1	005424	.	= 032212
TINPA	010732	TTINI	020476	WRTS8	006504				

. ABS. 032212 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DZTMHE,DZTMHE+SYSMAC.SML(400,1066),DZTMHE.P11(400,1370)
RUN-TIME: 29 42 2 SECONDS
RUN-TIME RATIO: 108/74=1.4
CORE USED: 32K (63 PAGES)

EOF1DZTMHESEQ

00010000

770224

B09
FDF10 411